Accessible design of user interface elements

Published by BFIT-Bund AG02 Software

Authors: Andreas Englisch and Carola Meixner, Deutsche Telekom MMS GmbH (External Link)
on behalf of the IT-Systemhauses der Bundesagentur für Arbeit (External Link)

Contact persons IT-Systemhaus der Bundesagentur für Arbeit: Markus Brand and Michael Zetzsche

Distributor: Überwachungsstelle des Bundes für Barrierefreiheit von Informationstechnik (External Link)
, AG02 Software applications

You can also download this guide as PDF (Opens PDF document)

Version: 1.0 (April 2023)

Table of contents

- Foreword (3 min)
  - Table of contents
- Goals and target group (3 min)
- Scope of validity (2 min)
  - Scope of validity
- Notes for the reader (7 min)
  - Layout of the document
  - Classification of the requirements
  - Coverage of the requirements
  - Coverage of the elements
  - Technology-specific features
- Application-related requirements (1 min)
  - Table of contents
- Conforming alternate version (4 min)
- Application language and change of language (3 min)
  - Presentation
  - Programming/interfaces
- Error prevention and correction (8 min)
  - Presentation
  - Programming/interfaces
- Help and support (4 min)
  - Presentation
  - Operation
- Resizing (3 min)
  - Presentation
- Accessibility API (17 min)
  - Programming/interfaces
  - Practical tip: Accessibility API with desktop applications
- Authentication (5 min)
  - Presentation and operation
  - Programming/interfaces

- Animations (4 min)
  - Presentation
  - Programming/interfaces
- Navigation sequence (8 min)
  - Operation
  - Practical tip: navigation sequence in Web applications
  - Practical tip: efficient navigation in applications
- Changes of context (6 min)
  - Operation
  - Programming/interfaces
- Time limits (3 min)
  - Presentation and operation
- Element-spanning requirements (1 min)
  - Table of contents
- Element status (4 min)
  - Operation
  - Programming/interfaces
- Operability status (3 min)
  - Presentation
  - Operation
  - Programming/interfaces
- Consistency (3 min)
  - Presentation
  - Programming/interfaces
- Keyboard shortcuts and shortcut keys (6 min)
  - Presentation
  - Operation
  - Programming/interfaces
  - Practical tip: keyboard shortcuts and shortcut keys
- Use of the pointing device (8 min)
  - Presentation
  - Operation
- Use of the keyboard (15 min)
  - Presentation
  - Operation
  - Programming/interfaces
  - Practical tip: use of the keyboard in Web and desktop applications
- User preferences (10 min)
  - Presentation
  - Practical tip: contrast adjustment for desktop applications
- Colors and contrasts (12 min)
  - Presentation
  - Programming/interfaces
  - Practical tip: color coding
  - Practical tip: anti-aliasing
- Font (11 min)
  - Presentation
  - Programming/interfaces
  - Practical tip: special characters

    - Practical tip: font style and text formatting
- Focus indicator (8 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: focus indicator
- Text cursor (3 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Required field label (8 min)
    - Presentation
    - Programming/interfaces
    - Practical tip: required field label
    - Practical tip: programmatic label of required fields in Web applications
- Text elements (1 min)
    - Table of contents
- Headings (6 min)
    - Presentation
    - Operation
    - Use of the keyboard: heading (in an application without a virtual cursor)
    - Programming/interfaces
    - Practical tip: headings in Web applications
- Label (20 min)
    - Presentation
    - Programming/interfaces
    - Practical tip: labeling in Web applications
- Description (15 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: descriptions in Web applications
- Text (11 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: text
- Graphical elements (1 min)
    - Table of content
- Graphic (10 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: graphics in Web applications that communicate a role, status, or value
    - Practical tip: graphics in Web applications
- Layout graphic (4 min)
    - Presentation
    - Operation
    - Programming/interfaces

    - Practical tip: layout graphics in Web applications
- Progress bar (7 min)
    - Presentation
    - Operation
    - Practical tip: progress bar in Web applications
- Structural elements (1 min)
    - Table of contents
- Desktop: Window (4 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Tooltip (7 min)
    - Presentation
    - Operation
- Form (4 min)
    - Presentation
- Toolbar (9 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: toolbar in Web applications
- Group (8 min)
    - Presentation
    - Programming/interfaces
    - Practical tip: form field groups in Web applications
- List (11 min)
    - Presentation
    - Operation
    - Practical tip: lists in Web applications
    - HTML
- Table (15 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: tables in desktop applications
- Hierarchical table (4 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Title bar (4 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Status bar (3 min)
    - Operation
    - Programming/interfaces
- Modal dialog (6 min)
    - Presentation
    - Operation

    - Programming/interfaces
- Control elements (1 min)
    - Table of contents
- Section separator (7 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Link (13 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: links in Web applications
- Button (12 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: buttons in Web applications
- Switch (7 min)
    - Presentation
- Toggle switch (7 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: toggle switch in Web applications
- Split button (6 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: split buttons in Web applications
- Context menu (4 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Menu (11 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Menu button (5 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Tab group (12 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: tab groups in Web applications
- Input field (one-line) (13 min)
    - Presentation
    - Operation

    - Programming/interfaces
    - Practical tip: one-line input field in Web applications
- Input field (multi-line) (9 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: multi-line input field in Web applications
- Password input field (7 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: password input field in Web applications
- Input field with autocomplete function (7 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Spin button (10 min)
    - Presentation
    - Operation
    - Use of the pointing device: spin button
    - Programming/interfaces
    - Practical tip: spin button in Web applications
- Selection list (12 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: selection list in Web applications
- Multiple selection list (9 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: alternatives to the selection list with multiple selection
    - Practical tip: simplified use of multiple selection
    - Practical tip: multiple selection list in Web applications
- Tree structure (13 min)
    - Presentation
- Control point (7 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Combined input field (10 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: combined input field in Web applications
- Drop-down list (13 min)
    - Presentation
    - Operation
    - Programming/interfaces

    - Practical tip: drop-down list in Web applications
- Slider (11 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: sliders in Web applications
- Scrollbars (8 min)
    - Presentation
    - Operation
    - Use of the pointing device: scrollbar
    - Programming/interfaces
- Radio buttons (10 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: radio buttons in Web applications
- Check box (8 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: check box in Web applications
- Composite control elements (1 min)
    - Table of contents
- Pagination (5 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Composite form fields (5 min)
    - Presentation
    - Programming/interfaces
    - Practical tip: composite form fields
- Accordion (10 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: accordion in Web applications
- Date picker (8 min)
    - Presentation
    - Operation
    - Programming/interfaces
- Color picker (8 min)
    - Presentation
    - Operation
    - Programming/interfaces
    - Practical tip: color picker in Web applications
- Carousel (8 min)
    - Presentation
    - Operation
    - Programming/interfaces

- Annex (1 min)
  - Table of contents
- References and sources (2 min)
- Glossary (14 min)
  - Accessibility
  - Accessible Description
  - Accessible Name
  - Area navigation
  - Assistive Technology
  - Authoring Tool
  - Captcha
  - Click area
  - Closed functionality
  - Color scheme
  - Context-sensitive Help
  - Continuous text
  - Contrast ratio
  - Control element
  - Custom element
  - Description
  - ENTER key
  - Explicit identifier
  - Flash
  - Focus indicator
  - Form
  - Form element
  - Hard line break
  - Hover area
  - Hybrid application
  - Image of text
  - Implicit identifier
  - Keyboard
  - Keyboard focus
  - Keyboard navigation
  - Keyboard shortcut
  - Label
  - Navigation
  - Non-text content
  - Open functionality
  - Platform software
  - Pointer
  - Pointing device
  - Quick navigation
  - Real-time event
  - Role
  - Screen magnifier
  - Screen reader
  - Shortcut key
  - Skip link

    - Speech recognition software
    - Standard element
    - Status
    - Status message
    - Support services
    - Text cursor
    - UI
    - UI element
    - User agent
    - User preferences
    - Value
    - Virtual cursor
- Index (6 min)
    - A
    - B
    - C
    - D
    - E
    - F
    - G
    - H
    - I
    - K
    - L
    - M
    - N
    - O
    - P
    - Q
    - R
    - S
    - T
    - U
    - W
    - V
    - Z

Foreword

Online betrachten

According to the objectives of the “Accessible Information Technology Ordinance” (BITV 2.0), modern information and communications technology should be designed so that, as far as possible and as a general rule, it is accessible without any restriction.

There is considerable ambiguity as regards the implementation, as the existing specifications leave a lot of scope for interpretation.

These discrepancies frequently lead to differing interpretations of the existing specifications, and in the event of the acceptance of artifacts or entire ICT systems, to annoyance, and in some cases, legal disputes.

The authors of this document have taken on the task of closing this gap between the legal requirements, guidelines, standards and existing design guides or style guides – also from software manufacturers.

The visual user interface elements described in DIN EN ISO 9241-161 have been considered for this purpose, supplemented by additional elements and extended with regard to the accessibility requirements in accordance with EN 301 549 v3.2.1. Requirements have been defined for each UI element in terms of presentation, operation and programming/interfaces.

This reference guide serves as a supplement to DIN EN ISO 9241-161 and as an aid for the implementation of EN 301 549 v3.2.1, and is not intended to replace the currently valid legal requirements or guidelines. On the contrary, this collection is based on and subject to a corresponding update process.

Table of contents

- Goals and target group
- Scope of validity
- Notes for the reader

Goals and target group

Online betrachten

This document attempts to fill the following gaps in the accessibility publications:

- For Web documents, in W3C, there are several specific examples of how accessible pages must be designed (see, for example How to Meet WCAG (Quickref Reference) (w3.org) (External Link)). Corresponding documents are not available for software.
- Section 11 of EN 301 549 details the requirements concerning software. These requirements are of a general nature. There is no description of how UI elements must be specifically implemented in order to meet the requirements. For desktop software, there is no equivalent to the WAI-ARIA Authoring Practices (WAI-ARIA Authoring Practices 1.2 (w3.org) (External Link)).
- In the form of WCAG2ICT (Guidance on Applying WCAG 2.0 to Non-Web Information and Communications Technologies (WCAG2ICT) (w3.org) (External Link)) in 2013, W3C published a document describing the way in which the WCAG requirements can be applied to software and documents. This document is outdated, however, and does not contain the requirements of the current WCAG 2.1.

The document is primarily aimed at software developers.

Other roles in the field of software development for which the document may be helpful include

- Design (especially with regard to the requirements of fonts, colors an contrasts and the information in the “presentation” sections for each UI element),
- Editing for the text content of the software and Help,
- Accessibility test,
- Disabled people who use software (to familiarize themselves with common keyboard conventions, for example, refer to the “use of the keyboard” sections for each UI element).

Scope of validity

Online betrachten

Scope of validity

This document describes the accessibility requirements for Web and Desktop software as well as hybrid applications (which combine Web and Desktop technologies) which operate on the Microsoft Windows platform software and which have an open functionality. The requirements are primarily derived from EN 301 549 (version 3.2.1, sections 9 and 11).

This document does not initially apply to the following software:

- Software with closed funktionality (Section 5.1 of EN 301 549),
- Software for two-way voice communication (section 6 of EN 301 549),
- Software with video functionality (section 7 of EN 301 549),
- Help authoring tools (section 11.8 of EN 301 549),
- Software with access to transposition or emergency services (section 13 of EN 301 549),
- Software that runs on platform software other than Microsoft Windows (such as Unix, Linux, Chrome OS, macOS, iOS, Android),
- Assistive Technology,
- Platform software,
- Apps for mobile devices (such as tablets or smartphones).

In addition, this document does not apply to:

- Hardware (section 8 of EN 301 549),
- Documents (section 10 of EN 301 549), even if the documents are interactive (e.g. spreadsheet with macros, PDF with form)

Many of the requirements described here can be transferred to software from other platforms.

Notes for the reader

Online betrachten

Layout of the document

The document is divided into the following sections:

- Application-related requirements, which apply to the whole of the software,
- Element-spanning requirements, which apply to all or several UI elements,
- Text elements,
- Grafical elements,
- structural elements (for the structuring of the dialog boxes in individual areas),
- control elements,
- Composite control elements (complex control elements which consist of several simple control elements).

Each section contains several areas, “control elements”, for example, is divided into one section for each specific control element.

The individual sub-sections are divided into:

- Introduction:
  - Synonyms: Other names for the described UI element, which can also be used to locate the element in the index,
  - Reference to similar elements or related topics,
  - Description of the element or topic,
- Presentation (requirements concerning the visual presentation)
- Operation (operating requirements, especially with keyboard and pointing devices)
- Programming/interfaces (requests for information which are communicated to the Accessibility API.

The requirements are presented in table form:

No.	Property	Description	Classification	Reference
Unique requirement number	Topic-related classification of the requirement	Requirement to be met, supplemented with explanatory notes if necessary	Relevance of the requirement (see Classification of the requirements))	Origin of the requirement (see References)

Note: The validity of the requirements is specified as follows:

- Web applications: “Web:”
- Desktop applications: “Desktop:”
- Hybrid applications: “Desktop:”
- Valid for all applications: Not applicable

Classification of the requirements

The requirements are classified as follows:

Classification	Meaning	Reference	Formulation
Must	Legal requirement according to BITV 2.0 Minimum requirements that must be met to achieve compliance with BITV 2.0	EN 301 549, Version 3.2.1 (External Link) Note: All the requirements of EN 301 549, sections 11.1 to 11.4, refer to the WCAG 2.1. The numbering of the corresponding requirements from EN 301 549 corresponds to the numbering in the WCAG 2.1.	• Must • May not

Classification	Meaning	Reference	Formulation
Should	Key requirements that should be met According to Section 3 (4) of BITV 2.0, efforts should be made to comply with the requirements for certain areas of application: “For central navigation and entry offers, as well as for offers that enable user interaction such as forms and the completion of authentication, identification and payment processes, efforts should be made to achieve the highest possible level of accessibility.”	• WCAG (External Link) 2.1, AAA criteria • WCAG 2.1, A- and AA-criteria that are not a component of section 11 of EN 301 549 • WCAG 2.2 criteria • Other W3C specifications • Other ISO norms • Further standards of the DIN EN ISO 9241 series (suitability for use) with special relevance to accessibility	• should • should not
Implementation recommendations, notes			• Note • Can • Recommended

The specific requirements concerning the use of the keyboard, i.e. which keys are to be used for which purpose, are classified as follows:

Classification	Meaning
Required	Minimum requirements If these requirements cannot be met, the alternate use of the keyboard should be documented.
Recommended	Recommended requirements The fulfillment of these requirements serves the purpose of easier and more efficient operations with the keyboard.

Note: The requirements regarding the use of the keyboard cannot be classified as “must” or “should”, as EN 301 549 only requires the possibility for the use of the keyboard, but does not define specific keys, as

these depend on the respective platform, for example. The elementary guide also contains notes, recommendations and practical tips. These are not standardized. In the notes, recommendations and practical tips, however, “must”, “may not”, “should” and “should not” are used insofar as reference is made to a requirement.

Coverage of the requirements

The sections on general topics (“application-related requirements” and “element-spanning requirements”) address the dialog-related requirements of EN 301 549 (in particular, section 9 on Web and 11 on Software). The requirements are explained here in general (i.e. not in relation to specific UI elements) and in as much detail as possible (i.e. with possible special cases, exceptions, etc.).

In the sections on individual UI elements (text, graphic, structure, control elements), only the relevant requirements regarding the respective UI element are listed. This article describes what a general requirement means as regards a specific element. However, the requirements are not necessarily explained in detail; i.e. for special cases and exceptions, reference is made to the respective general section.

For example:

- In the section on the UI element Checkbox the requirement for the visual resizing of the check box is not addressed, because no check box-specific problems or requirements in relation to zooming exist. However, the requirements concerning the resizing can be found in the “resizing” section and also apply to check boxes.
- In the section on the UI element check box, the specific contrast requirements are described so as to further explain the extent to which the general contrast requirements from the “colors and contrasts” section apply to the check box. However, the special case of the disabled check box is not discussed here, because exceptions for disabled elements are described in the “colors and contrasts” section.

Coverage of the elements

The following elements are not described in this document due to their limited degree of relevance to software:

- Rich text editor,
- Video,
- Audio,
- Image map,
- Maps.

However, there is the intention to include these requirements and elements in a future version of the document.

Technology-specific features

Some programming languages or frameworks do not allow all the requirements to be met due to limitations to the respective technology. In this case, it should be checked as to whether another programming language or another framework can be used. Alternatively, the requirements should be met as sufficiently as possible. All deviations should be documented in the help and in the accessibility declaration.

This document does not address technology-specific features, but focuses on the expected behavior of the UI elements.

Application-related requirements

Online betrachten

Table of contents

- Conforming alternate version
- Application language and change of language
- Error prevention and correction
- Help and support
- Resizing
- Accessibility API
- Authentication
- Animations
- Navigation sequence
- Changes of context
- Time limits

Conforming alternate version

Online betrachten

If the application fails to meet all the accessibility requirements, a conforming alternate version can be made available. In this respect, however, the following requirements must be met:

- The conforming alternate version meets all the requirements, i.e. it is completely accessible. If several alternative versions are offered, at least one alternative version is fully conforming. Therefore, offering specific alternative versions for individual user groups which only meet the requirements of the respective group is not permitted unless there is no alternative version which fulfills all the requirements of all user groups.
- The conforming alternate version is equivalent in terms of all its content and functions to the version which is not accessible. This means that the alternative version may not contain outdated information. Insofar as the standard version is offered in different languages, the alternative version must also be offered in the languages.
- The conforming alternate version can be achieved in an accessible way. This means:
  - The function for changing to the conforming alternate version must be accessible.
  - The standard version may not contain any keyboard traps or flashing content. In addition, the standard version may not contain any moving, flashing, automatically updating or acoustic content that cannot be stopped (https://www.w3.org/TR/WCAG21/#cc5).
  - Alternatively, both the conforming and standard version can be achieved using an accessible screen (e.g. the login screen), or the conforming alternate version is the standard version.
- The purpose and achievement of the conforming alternate version is explained in the documentation.
- The Support Service can explain the purpose and achievement of the conforming alternate version (to comply with EN 301 549, section 12.2.2).

It is recommended that only one version of the application is offered at a time and that it is designed so that it is accessible.

Note 1: A typical use case for a conforming alternate version is when the standard version of the Web application does not meet the contrast requirements for content consisting of text or graphics due to the corporate design. In this case, the conforming alternate version can use a CSS marking that provides for sufficient contrast.

Note 2: For selected Web applications that are already largely accessible, an overlay tool may be able to generate a conforming alternate version that fully meets the above requirements. However, this is usually not the case, especially if the Web application has problems that cannot be found and fixed automatically. Therefore, an overlay tool cannot be used as a standard to provide a conforming alternate version (Monitoring Center of the German Federal Government for the Accessibility of Information Technology - Publications - Joint Assessment of the Monitoring Centers of the Federal Government and Federal States for the Accessibility of Information Technology on the use of Overlay Tools (bfit-bund.de) (External Link))

Note 3: For desktop applications, EN 301 549 does not make any statements about alternative versions. However, it may be assumed that the same requirements apply to desktop applications.

Application language and change of language

Online betrachten

Synonyme: Language

See also: labels, description, text, font

Foreign-language content can be difficult for disabled people to understand. This is especially the case if such content is output by the text-to-speech software with the incorrect pronunciation.

Presentation

No.	Property	Description	Classification	Reference
1	Foreign-language content	The content should be displayed in the language of the user. Note 1: This does not apply to technical terms in foreign languages as long as it may be assumed that the users can understand them. Note 2: If the application is used by users who are from different language areas, the application should offer the possibility for changing the language. All content should then be displayed in the selected language.	Soll	EN 301 549: 11.2.4.6

Programming/interfaces

No.	Property	Description	Classification	Reference
2	Application language	The application language must be communicated to the Accessibility API Note 1: In the case of Desktop software, the application language can also be communicated to the Accessibility API through the platform. Note 2: With Web applications, the language must be communicated using the lang attribute on theelement.	Must	EN 301 549: 11.3.1.1.1
3	Web: Foreign-language content	The change of language within the application must be communicated to the Accessibility API. Note 1: This does not apply to • Proper names, • technical terms, • Words included in the vocabulary of the application language, and • Words whose language cannot be determined. Note 2: In HTML, the marking of the change of language takes place with the lang attribute.	Must	WCAG 2.1: 3.1.2 (AA)
4	Desktop: Foreign-language content	If the language of the foreign-language content can be programmatically communicated, this should take place. Note: This is possible in hybrid applications that use Web	Should	WCAG 2.1: 3.1.2 (AA)

Error prevention and correction

Online betrachten

Synonyms: Error message, error messages, input instructions, context-specific help

See also: Required field labeling,, description, help and support,, authentication, form

Error messages inform users about incorrect entries or incorrect operations. Error messages help with the correction of errors. Error messages can be displayed

- after submitting a form,
- when filling out a form, or
- when operating the software.

Input instructions help users to avoid errors. These can be displayed

- on the form,
- on the respective form field, or
- when operating the software.

Presentation

No.	Property	Description	Classification	Reference
5	Error message	If an error occurs, the incorrect form element must be identified and the cause of the error described in text form	Must	EN 301 549: 9.3.3.1, 11.3.3.1.1
6	Error message	The Submit button may not be disabled as long as the form is incomplete or filled out incorrectly. Note: This does not apply if there is an alternative method for displaying the error messages.	Must	EN 301 549: 9.3.3.1, 11.3.3.1.1
7	Error message	The error message must be permanently displayed. This does not apply if the error has been fixed. For further exceptions, see time limits .	Must	EN 301 549: 9.2.2.1, 11.2.2.1

No.	Property	Description	Classification	Reference
8	Error prevention	Form elements must have an expressive label so that their purpose is identifiable.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.3.3.2, 11.3.3.2
9	Error prevention	If the expected form field inputs cannot be derived clearly from the label of the form fields, additional input instructions must be provided (also see Beschreibungen). Note: Examples of input instructions include: • Required field labeling , • Instruction on the required input format, • Instruction on the permissible value range.	Must	EN 301 549: 9.3.3.2, 11.3.3.2

No.	Property	Description	Classification	Reference
10	Error prevention	If, when submitting a form, • a legal or financial commitment is made (e.g. when a contract is concluded), • user-managed data in a database is changed or erased, or • solutions are communicated in the context of an examination, one of the following error prevention options must be offered: • The submission of the form can be undone. • The data is checked by the application for its accuracy, and users can subsequently correct any errors. • Users are asked to check their data for its accuracy and to correct incorrect inputs.	Must	EN 301 549: 9.3.3.4, 11.3.3.4
11	Error prevention	When submitting information, one of the following error prevention options must be offered: • The submission of the form can be undone. • The data is checked by the application for its accuracy and the users can subsequently correct any errors. • The users are asked to check their data for its accuracy and are able to correct incorrect inputs.	Should	WCAG 2.1: 3.3.6 (AAA)

No.	Property	Description	Classification	Reference
12	Error prevention	A context-sensitive Help option should be offered.	Should	WCAG 2.1: 3.3.5 (AAA)
13	Error prevention	If it is necessary to enter information (such as a user name and password) when logging in, a variant must be available so that users do not have to remember this information. Note: The software can save the login data and enable the addition of the information from the clipboard or using a password manager.	Should	WCAG 2.2
14	Error prevention	If it is necessary to input data in a process several times, then such data should be filled in automatically after the first input or be made available for selection. Note 1: This also applies to information that may potentially be different, such as delivery and billing addresses. After inputting the delivery address, users can be given the opportunity to accept the information for the billing address automatically instead of having to enter it again. Note 2: This does not apply if inputting the data again is an essential requirement, is necessary for security reasons, or the data is no longer valid.	Should	WCAG 2.2

No.	Property	Description	Classification	Reference
15	Error correction	If the application is able to find suggestions for corrections to an incorrect input, these suggestions must be displayed. Note: This does not apply if it presents a risk to the security or the purpose of the application, in authentication processes, for example.	Must	EN 301 549: 9.3.3.3, 11.3.3.3
16	Error correction	If an automatic error correction takes place, an error message in text form must be displayed.	Must	EN 301 549: 9.3.3.1, 11.3.3.1.1

Programming/interfaces

No.	Property	Description	Classification	Reference
17	Update	If the error message is displayed as a status message, it must be marked so that the assistive technology is able to output it automatically. Note: If the error message is automatically focused, it is not considered a status message.	Must	EN 301 549: 9.4.1.3, 11.4.1.3.1

No.	Property	Description	Classification	Reference
18	Error message	Error messages on the form field must be linked to the form field in such a way that they are communicated to the Accessibility API. Note: If the Accessibility API you use does not allow error messages to be communicated with a form field, the error messages must be sent as part of the Accessible Name oder der Accessible Description . In this case, the Accessible Description should preferably be used, so that the Accessible Name is not excessively long.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
19	Error prevention	Input instructions which are visually assigned to a form field must be programmatically associated with the form field as input instructions. Note: If the Accessibility API you use does not allow for the communication of input instructions, the error messages must be communicated as part of the Accessible Name or the Accessible Description. In this case, the Accessible Description should preferably be used, so that the Accessible Name is not excessively long.	Must	EN 301 549: 11.1.3.1
20	Error prevention	If the technology used is able to identify the input purpose of form fields, the purpose of the form fields must be marked for the data of the respective users (e.g. surname, date of birth, place of residence) according to https://www.w3.org/TR/WCAG21/#input-purposes .	Must	EN 301 549: 9.1.3.5, 11.1.3.5.1

No.	Property	Description	Classification	Reference
21	Status	If it is only the case that a visual, non-textual error indicator is displayed on the incorrect field, then the error status must be communicated to the Accessibility API . Note: Depending on the technology, the error status may only mean “incorrect” or be communicated in a more differentiated way (with respect to the criticality, for example, as a “note”, a “warning” or an “error”, or with respect to the error type during a spell check, for example, as “spelling”, “grammar” and “expression”).	Must	EN 301 549: 9.4.1.2, 11.4.1.2

Help and support

Online betrachten

Synonyms: Manual, documentation, support, help

See also: description, error messages

Presentation

No.	Property	Description	Classification	Reference
22	Accessibility features	The Help option must identify all the accessibility features of the application and explain how to use them. Note: The Help option should describe which accessibility features are available, what they are intended for, what they are effective for and how they can be enabled	Must	EN 301 549: 12.1.1
23	Accessibility features	The Support must be able to name the accessibility features documented in the Help and explain how to use them.	Must	EN 301 549: 12.2.2

No.	Property	Description	Classification	Reference
24	Help dokuments	The Help documents must be provided in at least one digital, accessible format. Note 1: If the Help is offered as an accessible Web document, all the requirements in section 9 of EN 301 549 must be taken into account. If the Help is offered as an accessible non-Web document, all the requirements in section 10 of EN 301 549 must be taken into account. Note 2: This also applies to Help documents that cannot be accessed from within the application but are made available by Support, for example..	Must	EN 301 549: 12.1.2, 12.2.4
25	Support	The Support must take account of the communication needs of people with disabilities, either directly or through a mediator. Note: In this context, the provision of telephone support alone may be considered insufficient, as it is not accessible to people with hearing impairments (2 senses principle).	Must	EN 301 549: 12.2.3

No.	Property	Description	Classification	Reference
26	Reference to sensory properties	Information in the Help option which relates to the application must not refer exclusively to sensory properties. Note: Therefore, for example, a button in the application should not be described by its appearance or position, but by its label.	Must	EN 301 549: 9.1.3.3, 10.1.3.3, 11.1.3.3
27	Error prevention	A context-sensitive Help option should be offered.	Should	WCAG 2.1: 3.3.5 (AAA)
28	Consistency	If the application has a Help or Support option, this can be found in the same location throughout the applicataion. Note: If the application has multiple Help or Support options, it is considered sufficient when one of these meets the requirement.	Should	WCAG 2.2

Operation

Use the keyboard: Help

Action	Key	Classification
Desktop: Initialize Help	F1	Required
Initialize context-specific Help	SHIFT+F1	Recommended

Resizing

Online betrachten

Synonyme: Scaling, adjusting the font size, zoom

See also: font, text

The following requirements should ensure that the font size can be adjusted to the user preferences without assistive technology.

Presentation

No.	Property	Description	Classification	Reference
29	Resizing	It must be possible to resize the text of the application by up to 200% without the use of assistive technology. The resizing may not lead to a loss of content or functionality. Note 1: To do this in desktop applications, the application can offer its own zoom function or support the font size adjustment of the operating system (Settings > System > Display > Scale and layout: advanced scaling settings). Note 2: For Web applications, the zoom function of the browser should be supported. Note 3: For superior perceptibility, it should also be possible to resize non-text content.	Muss	EN 301 549: 9.1.4.4, 11.1.4.4.1

No.	Property	Description	Classification	Reference
30	Resizing	In the case of a screen size with a width of 320 px and a height of 256 px, no loss of content or function may occur. It should only be necessary to scroll content in either the vertical or horizontal direction, and not in both directions. Two-dimensional scrolling is only permitted with elements that are invariably two-dimensional (such as graphics, maps, videos or tables) Note 1: This requirement should ensure that the content can be resized by up to 400%. Note 2: When using a zoom ratio of 400%, all the content must be scaled accordingly. This does not apply to content that is already sufficiently large, such as headings. It must be possible to resize these by at least 200%. Note 3: If text is placed within two-dimensional content, such as tables, a single block of text (in a table cell, for example) must be readable without the need for two-dimensional scrolling.	Must	EN 301 549: 9.1.4.10, 11.1.4.10

Accessibility API

Online betrachten

Synonyms: Accessibility interface, interoperability with assistive technology, compatibility with assistive technology, platform support for accessibility services for assistive technology, programming interface for assistive technology

See also: Element status, changes of context, contrast adjustment

Assistive technology such as screen readers, screen magnifiers, Windows Contrast Adjustment and voice input software do not generally interact directly with the software or the browser, but using an interface for the accessibility which is made available by the operating system, for example: the Accessibility API (Application Programming Interface). The software and/or browser sends all the relevant information to the Accessibility API in standardized form, and the assistive technology accesses the information which is made available in the Accessibility API. However, the assistive technology only uses the information from the Accessibility API, which is relevant based on the needs of the user. If applications are operated using assistive technology, to a certain extent the operation does not take place directly, but is also conveyed using the Accessibility API.

The best-known Accessibility APIs on Microsoft Windows are:

- IAccessible2,
- MSAA (Microsoft Active Accessibility, Standard 1997-2005),
- UIA (Microsoft UI Automation, Standard seit 2005).

Windows applications should use the current Accessibility API UIA.

Software which does not use the Accessibility API of the operating system can implement its own interfaces for the communication of information to the assistive technology. In this context, Java applications use the Java Accessibility API (JAAPI).

The following information, for example, is communicated by the software and/or the browser to the Accessibility API and read out by the assistive technology if required:

- Role of an object (e.g. heading, check box, table cell),
- Status of an object (e.g. focused, focusable, disabled, open),
- Labeling of an object,
- Description of an object,
- Value of an object (e.g. for form fields),
- Possible values (e.g. maximum and minimum values in the case of specific form fields),
- Position in the object hierarchy (e.g. parent and child objects, amount of sibling objects, position relative to sibling objects),
- Spatial size and location in terms of the current screen area,
- Events (e.g. change of object properties).

Please note:

- Most programming languages which are usable for software development purposes support an Accessibility API. This applies to browsers as well.

- In this respect, the Accessibility API is usually supported automatically as long as the default elements of the programming language or markup language (e.g. HTML) are used. For example, if the language provides input fields as a control element, then the role, value, status, position in the object hierarchy, size, and location of the input field will be communicated correctly to the Accessibility API. In most cases, it is necessary for the visible label to be linked to the input field correctly so that it is also communicated to the API as the label of the input field.
- This applies in the same way to the standard characteristics of the programming language or markup language, which are communicated to the Accessibility API automatically. If an input field is provided with the “disabled” characteristic for example, the “disabled” and “not keyboard focusable” characteristics are communicated to the API automatically. 05.15.2023 Page 20 of 317
- Depending on the programming language or markup language used, it may be the case that certain characteristics of the object that are to be communicated to the Accessibility API must be explicitly specified (i.e. in text form), as otherwise they cannot be communicated.
- Unless a default element of the programming language or markup language is used, the developers are required to ensure that all the relevant object information is communicated to the Accessibility API correctly. If the language does not provide a possibility to explicitly define this information, only default elements should be used.
- If a programming language does not support an Accessibility API or it does not offer an alternative form of access to the required information through the various assistive technology, it should not be used, or the software must then meet the requirements detailed in Section 5.1 (closed functionality) of EN 301 549.

Programming/interfaces

No.	Property	Description	Classification	Reference
31	Desktop: General	Applications must use the existing Accessibility APIs of the operating system, insofar as the requirements in this table can be met with their use. If the Accessibility API is not sufficient to meet the following requirements, other methods must be used.	Must	EN 301 549: 11.5.2.3

No.	Property	Description	Classification	Reference
32	Syntax	Applications which use a markup language, and with which the Accessibility API or the assistive technology are able to access the markup language, must comply with the following rules regarding the marking: • Elements have full start and end tags, • Elements are correctly nested according to their specification, • Elements have no duplicate attributes, • The same IDs are not used more than once. Note: This does not apply if the markup language allows for deviations from these rules.	Must	EN 301 549: 9.4.1.1, 11.4.1.1.1
33	Role	The role of the elements must be communicated to the Accessibility API. Note: The role of the elements may not be changed during operation.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
34	Status	The status of the elements must be communicated correctly to the Accessibility API (see also Element status and Operability status).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
35	Value	The value of the elements must be communicated to the Accessibility API. In the case of elements that have a defined range of values, the minimum and maximum values must also be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7

No.	Property	Description	Classification	Reference
36	Orientation	If the orientation of the element has an influence on the operation, the orientation must be communicated to the Accessibility API. Note: Horizontally oriented elements can be operated with the RIGHT/LEFT ARROW, vertically oriented elements can be operated with the UP/DOWN ARROW, for example.	Must	EN 301 549: 9.4.1.2, 11.4.1.2
37	Name	The name and description of the elements must be communicated to the Accessibility API as the Accessible Name and Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
38	Keyboard shortcut, shortcut key	If the element has a visually perceptible keyboard shortcut or a visually perceptible shortcut key, these must be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1

No.	Property	Description	Classification	Reference
39	Desktop: Element hierarchy	The parent / child relationships of the elements must be communicated to the Accessibility API. Note: Among others, this allows the assistive technology to correctly output the following information: • The amount of sibling elements (e.g. list entries in a selection list), • The position of the element within the parent element (e. g. a list entry in a selection list), • Parent group label (e.g. in the case of a radio button group), • Nesting level (e.g. in the case of tree structures).	Must	EN 301 549: 11.5.2.9

No.	Property	Description	Classification	Reference
40	Web: Element hierarchy	The elements must be marked so that the browser is able to communicate the parent / child relationships of the elements to the Accessibility API correctly. Note: This can be achieved with the following methods: • correct nesting of elements according to the HTML specification, • correct nesting of the ARIA roles according to the ARIA specification, • use of corresponding ARIA attributes (such as aria-owns).	Must	EN 301 549: 9.1.3.1, 11.1.3.1
41	Desktop: Operation	All the control options of the element must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.11
42	Web: Operation	The control options of the element must correspond to the role used. Note: Different or additional operating options should be documented in the application and the Help option.	Must	EN 301 549: 9.4.1.2

No.	Property	Description	Classification	Reference
43	Operation	It must be possible to implement all the control options of the element with the assistive technology. Note 1: This applies, for example, to the enabling of elements, value and status changes, as well as changes of position for the focus and text cursor. Note 2: This does not apply to security-related applications for intelligence or military services, or to encryption software for the purposes of national security.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.14, 11.5.2.16, 11.5.2.17
44	Update	If an element characteristic which has been communicated to the Accessibility API is updated, this update must also be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
45	Update	In applications, Status messages must be marked so that they are output by the assistive technology without receiving the focus.	Must	EN 301 549: 9.4.1.3, 11.4.1.3.1
46	Desktop: Position	The spatial size and position of the elements must be communicated to the Accessibility API (see Fokusindikator).	Must	EN 301 549: 11.5.2.5, 11.5.2.10
47	Position	The focused element, the position of the text cursor and the selected entry within an element must be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.13

Practical tip: Accessibility API with desktop applications

The default elements of the platform software or the framework used typically communicate the correct information to the Accessibility API automatically. They should therefore be used with preference.

Example 1:

- The following figure shows the “Save as” dialog of the “Windows Fax and Scan” application.
- A complete list of the information which is communicated to the Accessibility API for the “file name” input field which is located in this dialog (read out with Accessibility Insights for Windows) is also shown.
- In addition, the text-to-speech output of the screen reader JAWS is displayed during the focusing of the “file name” input field. For the acoustic output, JAWS only uses the Accessibility API information which is relevant in the current context (e.g. label, role, value, keyboard shortcuts), partially translates this information into the application language (e.g. the “Edit(50004)” role into “input field”), and supplements this information with its own operating instruction (“add text”), which is derived from the communicated role.

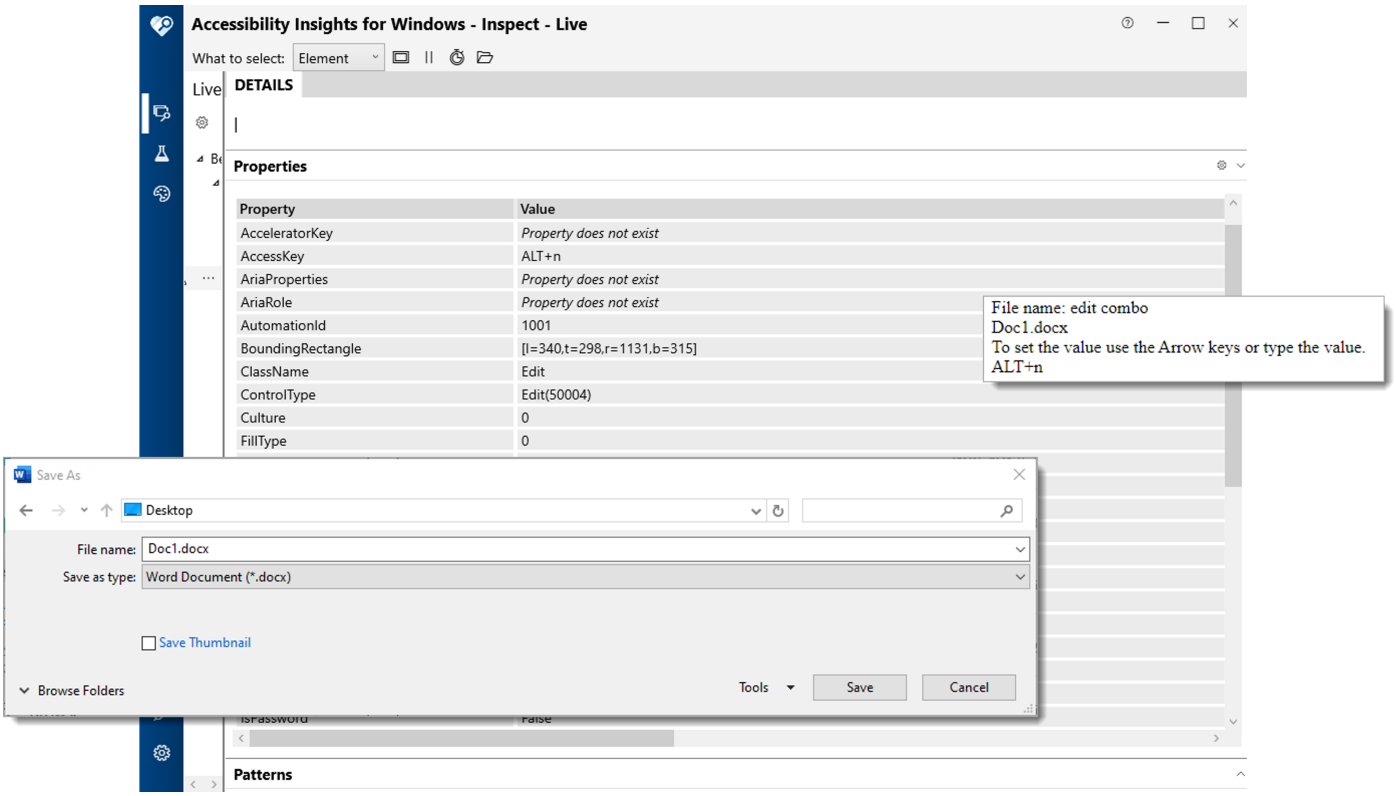

Figure 1: Information of the Accessibility API and its use by the screen reader JAWS

If custom elements are used, attention should be paid to the following in particular:

- the communicated role corresponds to the visual presentation and operation (in particular the use of the keyboard),
- the value and status should be communicated to the Accessibility API,
- the label is communicated to the Accessibility API as the Accessible Name,
- if available, the keyboard shortcuts, description) (as Accessible Description) and labels of the groups are also communicated to the Accessibility API,
- Updates concerning the value, status, label, etc. are communicated to the Accessibility API (note: the role of an element should not be changed).

If a custom element is implemented, it is often recommended that a related default element is used and customized accordingly, as it is then possible to use the basic functionality of the default element.

The corresponding characteristics of the Accessibility API should be used for the communication of the information. If there is no corresponding characteristic for sending the information in the Accessibility API or if the framework used does not support this characteristic, the information must be communicated in text form (i.e., as part of the Accessible Name or the Accessible Description).

For example: Disabled elements

- Disabled elements can usually be marked as disabled with an attribute. In the Accessibility API UIA, this corresponds to the IsEnabled:false characteristic. Assistive technology detects that the element is disabled due to this characteristic.
- If a programmatic marking as disabled is not possible, alternatively, the following options can be applied:
  - The element is removed.
  - The element is not designed as focusable (as long as it does not send any information and is not located within a range which can be read with the virtual cursor).
  - The element is named as “disabled” in the Accessible Name or Accessible Description.

For example: Button with value

- A button can have no value by default. Using Windows, however, it is possible to create a “button with value” custom element, which is based on the default button element. The value is then communicated to the Accessibility API UIA as the value of the Value characteristic.
- In HTML and ARIA, it is not possible to assign a value to a button. If the “button with value” is to be used in a hybrid application which is based on Web technologies, the value must be communicated in text form as part of the Accessible Name or the Accessible Description.

The communication of information about the corresponding characteristics of the Accessibility API is always to be preferred over the communication of such information in text form (as part of Accessible Name or the Accessible Description) for the following reasons:

- the assistive technology is able to output the characteristics which are communicated using the API in a way which is defined by the application or by the users,
- the assistive technology is able to translate the characteristics communicated using the API into the correct language,
- the assistive technology is able to output operating instructions or to offer operating modalities according to the characteristics which are communicated using the API,
- the assistive technology is able to offer a specific form of presentation based on the characteristics communicated using the API (e.g. a specific color when using contrast adjustment),),
- the assistive technology is able to output the characteristics (such as role, status, value) in a specific sequence, which ensures that relevant information is output first and that users are able to identify which information belongs to which type of characteristic,
- if required, users can configure their assistive technology so that they do not wish to receive outputs for certain characteristics.

None of this is possible if the information is only communicated in text form.

Example 2: The following figure shows three buttons with the label “Submit”, “Check” and “Delete”.

- The “Submit” button is visually identifiable as operable (black text color).
- The “Check” button is visually identifiable as disabled (gray font color). The button is programmatically marked as disabled. This implementation is to be preferred for disabled buttons.
- The “Delete” button is also visually identifiable as disabled (gray font color). The button is not programmatically marked as disabled, but only has one tooltip with the word “disabled”. This implementation may only be selected for disabled buttons if no programmatic marking as disabled is possible in the technology which is used.

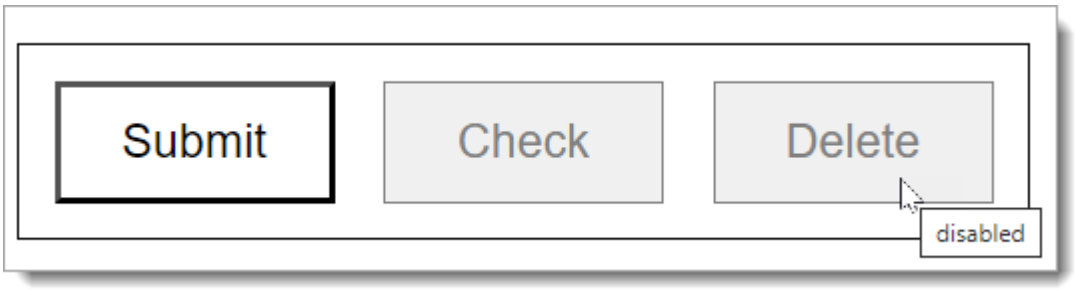

Figure 2: Three buttons, two of which are shown as disabled.

The following figure shows the same buttons from the previous figure, but when using Windows Contrast Adjustment (Contrast no. 1).

- The “Submit” button is correctly visually identifiable as operable (white text color).
- The “Check” button is correctly visually identifiable as disabled (green font color).
- The “Delete” button is displayed as operable (white text color), although it is disabled. The reason for the incorrect presentation is that it was not programmatically marked as disabled, but only as disabled in terms of its color. The color information is lost when using Windows Contrast Adjustment, however, so that a different visual presentation must be used here for the identification of the “disabled” status (e.g. a strike-through).

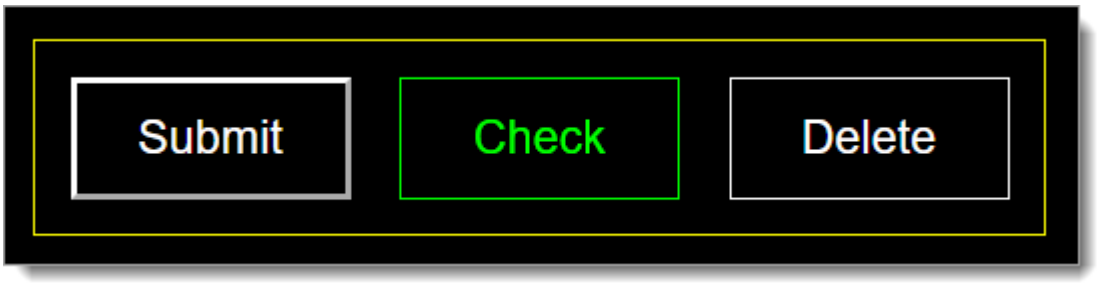

Figure 3: The three buttons from the previous figure when using Windows Contrast Adjustment.

The following figure shows the acoustic screen reader output of the three buttons from the previous figure (with the example of the JAWS element overview).

- The “Submit” button is correctly output as operable (“button” role).
- The “Check” button is correctly output as disabled (“not available” status).
- The “Delete” button is incorrectly output as operable (“button” role), as the “disabled” status is only communicated by tooltip, and the tooltip content is only output by the screen reader with certain navigation methods.

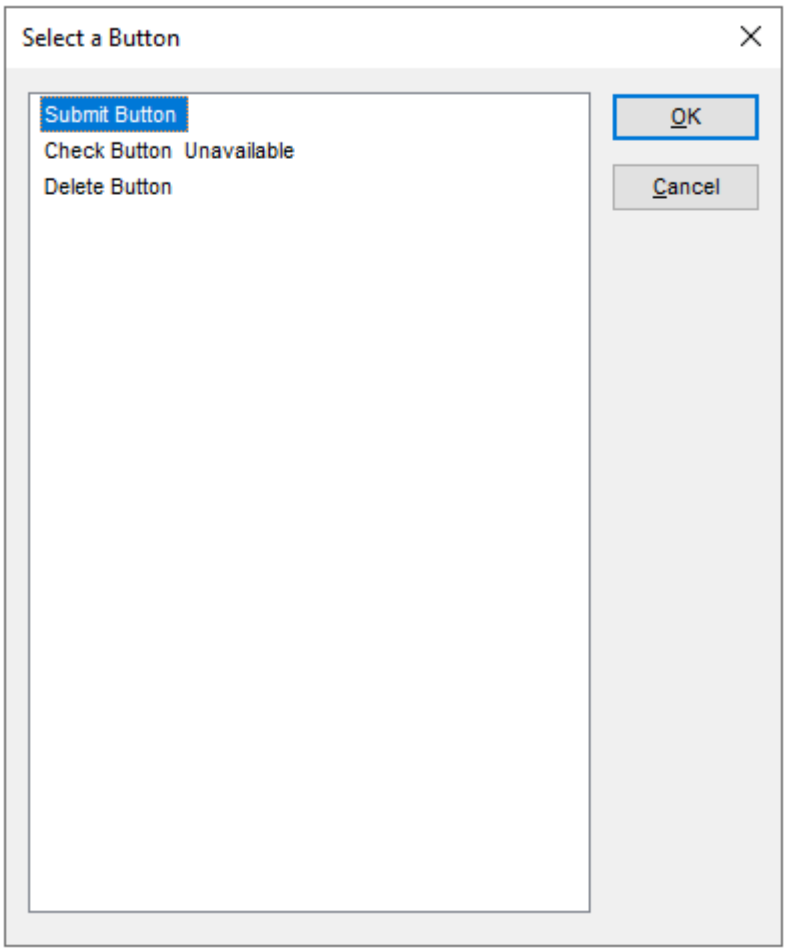

Figure 4: Screen reader output of the three buttons from the previous figure.

In the three following figures, the output of the three buttons from the preceding figures is shown on the Braille line (with the example of JAWS).

- The “Submit” button is correctly output as operable (“button” role) which is shown with the abbreviation “sltr”).
- The “Check” button is correctly output as disabled (“not available” status, which is shown with the abbreviation “xx”).
- The “Delete” button is shown as disabled (“disabled” description), but the output is not in the abbreviated form, not at the expected position, and is not translated into the language of the user.

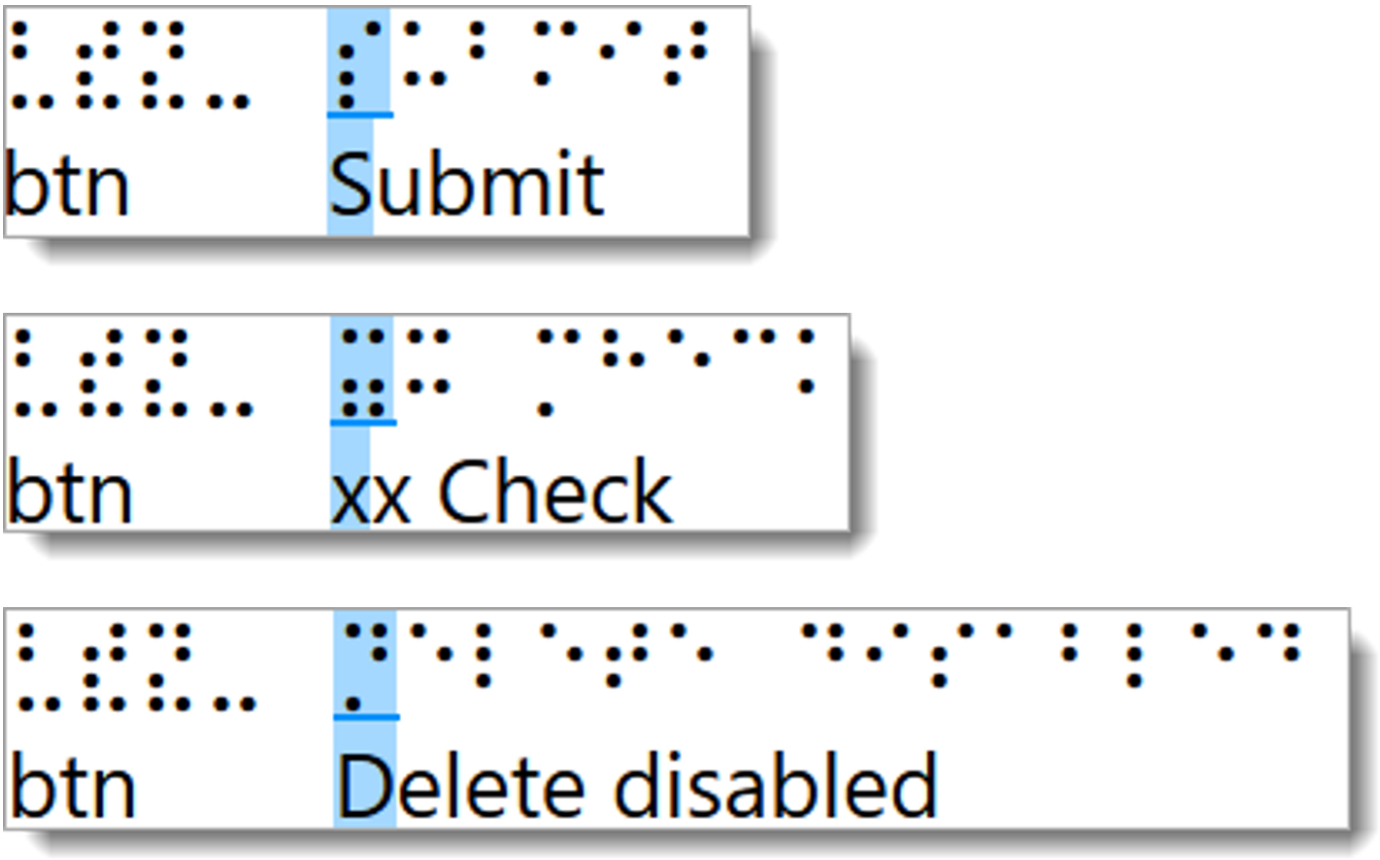

Figure 5: Braille output of the three buttons from the previous figures.

Authentication

Online betrachten

Synonyme: Sign in, sign out, login, logout

See also: Changes of context, time limits, password input field

The authentication encompasses the processes of signing in and out of an application or within an application. Logging in may be required to be able to use an application or certain parts of the application.

Note: Requirements regarding the authentication control elements (such as input fields, password input fields and buttons) are described for the respective element.

Presentation and operation

No.	Property	Description	Classification	Reference
48	Captcha	If a Captcha is used during the authentication, appropriate Captchas with at least two different sensory systems must be offered for different disabilities. Note 1: For hearing-impaired people, a visual Captcha can be offered, and for blind people, an audio Captcha can be offered. Note 2: The use of Captchas which require users to solve a task should be avoided as far as possible. Note 3: If a Captcha cannot be dispensed with, a non-sensory Captcha (such as one with a general knowledge question or a math task) should also be offered.	Must	EN 301 549: 9.1.1.1, 11.1.1.1

No.	Property	Description	Classification	Reference
49	Logout	If an automatic logout takes place in the application after a certain time, it must be possible for this time limit • to be disabled in advance, or • to be adjusted in advance (extendible to at least 10-times the time); or • to be extendible to at least 10 times the time with a simple action at least 20 seconds before expiry. Note: This does not apply to an automatic logout which takes place after at least 20 hours.	Must	EN 301 549: 9.2.2.1, 11.2.2.1
50	Logout	No automatic logout should take place in the application.	Should	WCAG 2.1: 2.2.3 (AAA)
51	Logout	If an automatic logout takes place, it should be possible to continue working without a loss of data after logging in again.	Should	WCAG 2.1: 2.2.5 (AAA)
52	Logout	Users should be informed in advance of the time at which an automatic logout takes place if the logout can lead to a loss of data. Note: This does not apply to a logout after more than 20 hours.	Should	WCAG 2.1: 2.2.6 (AAA)
53	Login	If a certain form of biometric data is required for the login (e.g. fingerprint, facial recognition), an alternative login method must be made available. Note: The alternative login method may also be based on biometric data provided that a different form of biometric data is used for this.	Must	EN 301 549: 5.3

No.	Property	Description	Classification	Reference
54	Login	If the login takes place with the movement of the device or the user, an alternative login method must be provided. Note: The movement of the device or the user may be necessary to enter biometric data (e.g. fingerprint, facial recognition), for example.	Must	EN 301 549: 9.2.5.4, 11.2.5.4
55	Login	If it is necessary to enter information (such as a user name and password) when logging in, a variant must be available for which users do not have to remember this information. Note: The application can save the login data and/or enable the addition of the information from the clipboard or using a password manager.	Should	WCAG 2.2: 3.3.7 (A)

Programming/interfaces

No.	Property	Description	Classification	Reference
56	Status	If the technology used is able identify the input purpose of form fields, the purpose of the form fields must be marked for the data of the respective users (such as the name, email address, password) according to Input Purposes for User Interface Components - Web Content Accessibility Guidelines (WCAG) 2.1 (w3.org) (External Link) Note: This does not mean the role (e.g. “input field”) or the specific label (e.g. “user name”), but a defined input purpose (e.g. “first name”, “user name”, “new password” or “current password”).	Must	EN 301 549: 9.1.3.5, 11.1.3.5.1

Animations

Online betrachten

Synonyms: Sparkling, flashing, updates, flash

See also Time limits, carousel, Video, progress bar

Animations are:

- automatic visual changes or
- unexpected visual changes when using the application.

Examples of automatic visual changes:

- moving text,
- automatically started video,
- automatically scrolling content,
- sparkling or flashing content,
- content that is automatically updated after a certain time.

Examples of unexpected visual changes when using the application are:

- content is additionally animated when scrolling in the screen,
- content is scrolled through at different speeds when scrolling in the screen,
- manually started video contains flashing content,
- flashing error message which is shown after sending a form,
- when viewing content, the content is not displayed immediately, but shown in animated form (e.g. scaled, shifted, rotated or changed in terms of its transparency).

Presentation

No.	Property	Description	Classification	Reference
57	Status	Content which flashes for more than 3 times per second and exceeds a certain flash threshold (siehe General flash and red flash thresholds - Web Content Accessibility Guidelines (WCAG) 2.1 (w3.org) (External Link)), must be avoided.	Must	EN 301 549: 9.2.3.1, 11.2.3.1
58	Flashing	Content which flashes for more than 3 times per second should be avoided.	Should	WCAG 2.1.: 2.3.2 (AAA)

No.	Property	Description	Classification	Reference
59	Animation	If the application contains content that automatically moves, scrolls or flashes, and if this animation lasts longer than 5 seconds and is displayed along with other content, it must be possible for the animation to be paused, stopped, or hidden. Note: Avoiding the use of automatically moving, scrolling or flashing content is recommended.	Must	EN 301 549: 9.2.2.2, 11.2.2.2
60	Animation	If moving animations are displayed during the operation of the application, a mechanism should be available to disable them. Note: The mechanism for disabling moving animations can be implemented in the application. Alternatively, the user preference in the operating system should be taken into consideration (Control panel > Ease of Access Center > Make the computer easier to use > Turn off all unnecessary animations).	Must	EN 301 549: 9.2.2.2, 11.2.2.2
61	Update	If the application contains content which is updated automatically and displayed along with other content, the update must be paused and/or stopped, or it must be possible to determine the frequency of the update and/or to hide the area with the automatically updated content. Note: Avoiding the use of automatically updating content is recommended.	Must	EN 301 549: 9.2.2.2, 11.2.2.2

Programming/interfaces

No.	Property	Description	Classification	Reference
62	Alternative text	If information is communicated using the animation, this information must also be communicated in text form. Note: Additional requirements apply to videos.	Must	EN 301 549: 9.1.1.1, 11.1.1.1
63	Reference to sensory properties	Information which serves the understanding or the operation of the application may not make exclusive reference to the animation of the described elements.	Must	EN 301 549: 9.1.1.3, 11.1.3.3

Navigation sequence

Online betrachten

Synonyms: Tab sequence, focus order

See also: Use of the keyboar, Changes of context, Time limits

The navigation sequence decides the sequence in which the elements and areas to be focused with the keyboard receive the focus. Typically, this applies to the following navigation methods:

- Navigating between the elements with the tab key,
- Navigating within the elements with the arrow keys,
- Navigating between areas (e.g. with the F6 key),
- Quick navigation (e.g. with PAGE UP, END).

Operation

No.	Property	Description	Classification	Reference
64	Navigation sequence	The navigation sequence must be such that the contents can be viewed in a meaningful sequence and the control elements can be accessed according to their task-appropriate processing sequence. Note: This is achieved, for example: • when opening a modal dialog this receives the focus, • when closing a modal dialog, the focus is reset to the triggering element, • the buttons for submitting a form are located at the end of the form	Must	EN 301 549: 9.2.4.3, 11.2.4.3

No.	Property	Description	Classification	Reference
65	Navigation sequence	In the case of elements which are operated with the arrow keys, the navigation must be limited to the element when using the arrow keys. Note: This relates to, for example, radio button groups, selection lists, tabs, menus, toolbars..	Must	EN 301 549: 9.2.4.3, 11.2.4.3
66	Navigation sequence	After page updates which make a focus change necessary, the focus must be positioned so that the work can be continued in a consistent way. For example: After deleting an element, the focus should be positioned on the previous or next element.	Must	EN 301 549: 9.2.4.3, 11.2.4.3
67	Navigation sequence	With modal dialogs, the navigation must be limited to the dialog. Note: The rest of the application can only be focused and operated when the modal dialog is closed.	Must	EN 301 549: 9.2.4.3, 11.2.4.3

No.	Property	Description	Classification	Reference
68	Navigation sequence	The navigation sequence should be appropriate for the work task. Note 1: For German language applications, this usually means that the navigation should correspond to the reading sequence, and should take place from the top left to the bottom right. Note 2: The navigation sequence should match in both navigation directions (forward and backward). Note 3: It may be necessary for the visual sequence to be adjusted to meet this requirement.	Should	DIN EN ISO 9241-171: 9.3.18
69	Web: Amount of navigation steps	It must be possible to skip areas of content that occur on several pages (see Practical tip: efficient keyboard navigation).	Must	EN 301 549: 9.2.4.1
70	Change of context	No loss of focus may occur during the keyboard navigation.	Must	EN 301 549: 9.3.2.1, 11.3.2.1
71	Change of context	Form elements may not be subject to an unexpected loss of focus when changing their value (see Change of context).	Must	EN 301 549: 9.3.2.2, 11.3.2.2
72	Change of context	When using the keyboard, the focus must be positioned correctly if an expected change of context occurs that has to be operated. Alternatively, the change of context must receive the keyboard focus after the current focus position (see change of context).	Must	EN 301 549: 9.2.4.3, 11.2.4.3

No.	Property	Description	Classification	Reference
73	Web: Consistency	Navigation elements must be displayed in the same relative sequence on each page within the application and receive the keyboard focus (see Consistency).	Must	EN 301 549: 9.3.2.3
74	Desktop: Consistency	Navigation elements should be displayed in the same relative sequence on each screen within the application and receive the keyboard focus (see Konsistenz).	Should	WCAG 2.1: 3.2.3 (AA)

Practical tip: navigation sequence in Web applications

HTML

The correct navigation sequence should be controlled by the order of the elements in the source code. The tabindex=0 attribute should not be used. The autofocus attribute should only be used with caution, as the contents before the automatically focused element are only perceptible with difficulty for visually impaired or blind people. There are applications in which autofocus can be put to good use, however, e.g. on a login page for the focusing of the first input field. Further information: 6.6.3 The tabindex attribute - HTML Standard (whatwg.org), 6.6.7 The autofocus attribute - HTML Standard (whatwg.org)

ARIA

The correct navigation sequence should be controlled by the order of the elements in the source code. The tabindex=0 attribute must be used for control elements that would not otherwise receive the keyboard focus. Elements that should be focusable by JavaScript but are not automatically in the TAB circuit are marked with tabindex=-1. This applies, for example, to buttons in toolsbars, Einträge in selection lists oder radio buttons within a group of radio buttons. In these cases, just one button, one list entry and/or one radio button is marked with tabindex=0 and all others are marked with tabindex=-1. Further information: Developing a Keyboard Interface | APG | WAI | W3C

Practical tip: efficient navigation in applications

If the current application window contains several focusable elements, users who depend on the use of a keyboard cannot navigate through the window efficiently because they have to use the TAB key to navigate through these elements. To allow for an efficient form of keyboard navigation, it is recommended that one or more of the following methods are implemented and documented in the Hilfe option:

- Area navigation (e.g. with F6),
- Skip links at the start of the window and/or before areas with several navigation steps,
- keyboard shortcuts for frequently required functions and/or control elements,
- Designing areas with several navigation steps so that they can be shown and hidden (e.g. Menu with sub-menus, tabs, accordion),
- Outsourcing of contents with several navigation steps to secondary screens (e.g. in dialog windows which can be opened separately),

- Use of grouping elements that are navigated within using the arrow keys instead of the TAB key (e.g. toolbar, menu),
- with applications that do not support the virtuellen Cursor: Mode in which control elements receive the focus only.

Changes of context

Online betrachten

Synonyms: Updates, change of context

See also: Animations, time limits, navigation sequence, modal dialog

Changes of context are:

- Changing to another program,
- Changing of the view port (e.g. switching to another application window),
- Changes to the keyboard focus,
- Changing the content that changes the meaning of the page.

Changes of context expected and required during operation include:

- Enabling a link: Opening a new screen,
- Enabling a Help link: Open Help, possibly in another application (e.g. in a browser or PDF reader),
- Enabling an internal page link: Scrolling to the linked location and changing the keyboard focus,
- Enabling a Submit button: Opening a new screen,
- Enabling a Delete button: Removing the element to be deleted, possibly changing the keyboard focus if necessary (as the delete button has been disabled or removed),
- Enabling the Logout button: Logout from the application,
- Enabling a Menu button: Opening the menu and focusing of the first menu item,
- Enabling a button to open a modal dialog: Opening the dialog and focusing of the first element in the dialog,
- Page scrolling: change of the visible area,
- Navigation with the keyboard (e.g. with the tab key): Change of the keyboard focus, possible change of the visible area (if the focused element is not situated in the visible area),
- Navigating through a radio button group in which there are associated form fields behind each radio button which can only be operated depending on the radio button: The form elements are enabled when the radio button is selected, the other form elements are disabled.
- Navigation through a group of tab panels: The corresponding tab is shown (alternatively, only after the enabling of the tab panel).

Examples of unexpected changes of context which must be avoided or announced:

- Enabling of a check box within a form: Showing additional form fields that are visually located in the navigation sequence before the check box,
- Text input in an input field: Deleting of entries already made in other fields because the entries in this field and the other fields are mutually exclusive,
- Text input in an input field: After reaching the maximum number of characters, the focus is placed in the following input field,

Examples of unexpected changes of context which have to be avoided or announced:

- Enabling a link: Opening a new screen in another application,
- Enabling of a tab panel: The displayed tab or an element in it receives the focus.

Examples of unexpected changes of context which must be avoided:

- Navigation through a group of radio buttons: Opening a new screen,
- Navigation through a table: Opening a modal dialog,
- Fully filled out form: Submitting the form,
- Focusing of a Submit button: Submitting the form,
- Focusing of the Delete button: Removing the element to be deleted.

Operation

No.	Property	Description	Classification	Reference
75	Use of the keyboard, use of pointing device	During the focusing of an element, no change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1
76	Use of the keyboard, use of pointing device	When a form element changes its value, no unexpected change of context may occur. Note 1: Unexpected changes of context are changes of context that do not conform to the default behavior of the element and have not been previously announced. Note 2: If unexpected changes of context prevent the use of the keyboard (e. g. due to a loss of focus during use), these are not permitted, even if they are announced in advance.	Must	EN 301 549: 9.3.2.2, 11.3.2.2
77	Use of the keyboard, use of pointing device	Changes of context should only occur if the users have initiated them. Alternatively, the user should be able to disable the changes of context.	Should	WCAG 2.1: 3.2.5 (AAA)
78	Updates	It must be possible to disable or adjust automatic changes of context that do not take place after 20 hours (seee time limits and animations).	Must	EN 301 549: 9.2.2.1, 11.2.2.1, 9.2.2.2, 11.2.2.2

No.	Property	Description	Classification	Reference
79	Updates	It should be possible to avoid or disable automatic changes of context. Note: This does not apply to emergency messages.	Should	WCAG 2.1: 2.2.3 (AAA), 2.2.4 (AAA)

Programming/interfaces

No.	Property	Description	Classification	Reference
80	Update	Status messages must be marked so that they are output by the assistive technology without receiving the focus.	Must	EN 301 549: 9.4.1.3, 11.4.1.3.1

Time limits

Online betrachten

Synonyme: Timeout, Time limit

See also: Animations, changes of context, authentication, carousel

Time limits are specified time constraints in which to view content, operate elements or complete tasks. Time limits may occur, for example, if

- an automatic logout occurs after a certain period of inactivity,
- a PIN required for authentication is only valid for a certain time,
- messages are hidden after a certain time,
- content is updated automatically (e.g. in a carousel),
- scrolling is displayed.

Presentation and operation

No.	Property	Description	Classification	Reference
81	Use of the keyboard	The use of the keyboard must be possible without specified time constraints. Note: The following, for example, is not permitted: • having to press a key for a certain time to trigger the associated function, • having to press two keys in succession with a certain gap.	Must	EN 301 549: 9.2.1.1, 11.2.1.1.1

No.	Property	Description	Classification	Reference
82	Adaptability	It must be possible for time limits • to be disabled in advance, or • to be adjusted in advance (extendible to at least 10-times the time), or • to be extendible to at least 10 times the time with a simple action at least 20 seconds before expiry. Note: This does not apply to time limits • which last longer than 20 hours, or • which are necessary (e.g. during a test or during real-time events such as an auction).	Must	EN 301 549: 9.2.2.1, 11.2.2.1
83	Avoid	Time limits should be avoided. Note: This does not apply to, e.g. real-time events or videos	Should	WCAG 2.1: 2.2.3 (AAA)
84	Inform	Users should be notified in advance of time limits which may lead to a loss of data. Note: This does not apply to time limits which last longer than 20 hours.	Should	WCAG 2.1: 2.2.6 (AAA)

Element-spanning requirements

Online betrachten

Table of contents

- Element status
- Operability status
- Consistency
- Keyboard shortcuts and shortcut keys
- Use of the pointing device
- Use of the keyboard
- User preferences
- Colors and contrasts
- Font
- Focus indicator
- Text cursor
- Required field label

Element status

Online betrachten

See also: Operability status, focus indicator

Several control elements can have a status, e.g.

- highlighted or not highlighted,
- selected or unselected,
- open or closed,
- pressed or not pressed,
- correct or incorrect (also see Error prevention and correction),
- focused or not focused (also see Focus indicator),
- operable or disabled (also see Operability status).

Note: Which specific status is possible for which control element is specified by the programming language used.

Note: Specific recommendations on specific status changes for individual elements are described at the respective element.

No.	Property	Description	Classification	Reference
85	Contrast	If the status difference is only conveyed visually with the use of color, the contrast ratio between these colors must be at least 3:1. Note 1: Alternatively, the status difference can be conveyed as follows: • different frame shape, • different font style, • icons. Note 2: If different frames or icons are used to convey the status difference, they must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.1, 11.1.4.1, 9.1.4.11, 11.1.4.11
86	Contrast	Regardless of the status, a contrast ratio of at least 4.5:1 for text and 3:1 for the content of graphics must be maintained.	Must	EN 301 549: 9.1.4.3, 11.1.4.3, 9.1.4.11, 11.1.4.11

Operation

No.	Property	Description	Classification	Reference
87	Use of the keyboard	Status changes that can be made with a pointing device must also be possible with the use of the keyboard (see Use of the keyboard).	Must	EN 301 549: 9.2.1.1, 11.2.1.1

Programming/interfaces

No.	Property	Description	Classification	Reference
88	Status	The status of the control element must be communicated to the Accessibility API .	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.13
89	Status change	The status change must be possible using the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.14, 11.5.2.16
90	Update	Status updates must be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
91	Contrast adjustment	To ensure that the status of the elements is visible when using contrast adjustment, the status should not only be communicated in color.	Should	EN 301 549: 11.7

Operability status

Online betrachten

Synonyms: disabled, inactive, display mode, write-protected, read-only, display-only

See also: Element status

Control elements can be operated (standard) or disabled. Form fields can also be read-only or in display mode.

Note: Which specific status is possible for which control element is specified by the programming language used.

Presentation

No.	Property	Description	Classification	Reference
92	Contrast	If the status difference is only conveyed visually with the use of color, a contrast ratio of at least 3:1 must be maintained between the different colors.	Must	EN 301 549: 9.1.4.1, 11.1.4.1
93	Contrast	Regardless of the status, a contrast ratio of at least 4.5:1 for text and 3:1 for the content of graphics must be maintained. Note: Contrast requirements do not apply to disabled elements unless they provide information.	Must	EN 301 549: 9.1.4.3, 11.1.4.3, 9.1.4.11, 11.1.4.11

Operation

No.	Property	Description	Classification	Reference
94	Use of the keyboard	In applications that do not support the virtual Cursor , all operable and read-only elements as well as display mode elements must receive the keyboard focus. Disabled elements must receive the focus if they convey information. Note: If the application contains several disabled elements or elements in the display mode, to avoid unnecessary navigation steps for seeing keyboard users, there should be an operating mode in which only the operable elements receive the focus.	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.1.3.1, 11.1.3.1

Programming/interfaces

No.	Property	Description	Classification	Reference
95	Status	The status of the control element must be communicated to the Accessibility API .	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
96	Update	Status updates must be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
97	Contrast adjustment	To ensure that the status of the elements is also visible when using contrast adjustment disabled elements should be marked as such. To ensure that their status is identifiable when using contrast adjustment, read-only elements should not only differ from operable elements in terms of their color. Elements in the display mode should be marked as text.	Should	EN 301 549: 11.7

Consistency

Online betrachten

Synonyms: Conformity with expectations

Consistent design and operation helps users to understand and operate the application efficiently. Consistency should be aimed for:

- within the application and
- with other applications..

Presentation

Nr.	Eigenschaft	Beschreibung	Klassifizierung	Referenz
98	Web: Consistent presentation	Control elements and content with the same function must be labeled and designed consistently within the application.	Must	EN 301 549: 9.3.2.4
99	Web: Consistent presentation	Navigation elements must be displayed in the same relative sequence on each page within the application and receive the keyboard focus. Note 1: The same relative order means that, for example, elements A and B always occur on the pages in the order “A B” and not as “B A”, although depending on the page, further elements may be between A and B (e.g. “A X B” and “A Y B”). Note 2: This does not apply if the sequence of the navigation elements has been changed by the users.	Must	EN 301 549: 9.3.2.3
100	Desktop: Consistent presentation	Control elements and content with the same function should be labeled and designed consistently within the application.	Should	WCAG 2.1: 3.2.4 (AA)
101	Desktop: Consistent presentation	Navigation elements should be displayed in the same relative sequence on each screen within the application and receive the keyboard focus.	Should	WCAG 2.1: 3.2.3 (AA)

Programming/interfaces

No.	Property	Description	Classification	Reference
102	Name	The visual label must match or be included in the Accessible Name.	Must	EN 301 549: 9.2.5.3, 11.2.5.3.1
103	Web: Name	Control elements with the same function must be labeled consistently within the application using the Accessible Name.	Must	EN 301 549: 9.3.2.4
104	Desktop: Name	Control elements with the same function should be labeled consistently within the application using the Accessible Name.	Should	WCAG 2.1: 3.2.4 (AA)

Keyboard shortcuts and shortcut keys

Online betrachten

Synonyms for keyboard shortcuts: Key combination, keyboard combination, keyboard command, keyboard equivalents, key sequence, access key, hot key, shortcut

Synonyms for shortcut keys: Mnemonic, acceleration key, shortcut key, menu accelerator

Keyboard shortcuts are keys or combinations of keys with which functions can be initialized or elements can be focused from within the application. The keyboard shortcuts can be initialized regardless of the current focus position. Keyboard shortcuts often consist of a combination of a printable character (letter, number, special character) with one or more modifier keys (e.g. CTRL or ALT). Keyboard shortcuts must be unique.

Shortcut keys are keys that can only be used within an element for the purpose of efficient navigation. They are usually used in menus. The shortcut key is usually the first letter of the corresponding element (e. g. menu item). Shortcut keys do not have to be unique: If, for example, different entries within a menu have the same shortcut key, the entries can be focused one after the other by pressing the key. Shortcut keys are used without modifier keys.

Presentation

No.	Property	Description	Classification	Reference
105	Documentation	Keyboard shortcuts that are necessary for exiting a keyboard trap must be documented in the application in such a way that they are perceptible before the keyboard trap or when it is reached.	Must	EN 301 549: 9.2.1.2, 11.2.1.2
106	Documentation	Keyboard shortcuts that are necessary for operating the application must be documented in the Help option.	Must	EN 301 549: 12.1.1
107	Documentation	All keyboard shortcuts and shortcut keys should be documented in the application.	Should	WCAG 2.1: 3.3.5 (AAA)

Operation

No.	Property	Description	Classification	Reference
108	Keyboard shortcut	If printable characters are used as keyboard shortcuts without a modifier key, it must be possible to: • disable the keyboard shortcut, or • set the keyboard shortcut so that it can be used with modifier keys.	Must	EN 301 549: 9.2.1.4, 11.2.1.4.1
109	Keyboard shortcut	Keyboard shortcuts should not be the only way of using the keyboard, but an additional method. Note: all control elements should therefore receive the focus with TAB or arrow keys.	Should	DIN EN ISO 9241-171: 9.3.10

Programming/interfaces

No.	Property	Description	Classification	Reference
110	Documentation	Keyboard shortcuts that are necessary for exiting a keyboard trap must be communicated to the Accessibility API in the application in such a way that they are perceptible before the keyboard trap or when it is reached.	Must	EN 301 549: 9.2.1.2, 11.2.1.2
111	Documentation	Keyboard shortcuts and shortcut keys that are visually perceptible in the application must also be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1

Practical tip: keyboard shortcuts and shortcut keys

Platform-specific keyboard shortcuts

The keyboard shortcuts of the platform (of the Windows operating system, for example) should not be overridden or disabled in your application. i.e. the keyboard shortcuts of the platform should also work in the application (e.g. CTRL+X and CTRL+V to cut and paste text).

Keyboard shortcuts of assistive technology

The assistive technology uses several keyboard shortcuts. These keyboard shortcuts should not be used in the application, as otherwise the assistive technology or the application won't work with these keyboard shortcuts.

The screen readers use the INSERT key as a modifier key for several of their keyboard shortcuts, for example. Therefore, the INSERT key should not be used as a modifier key for the keyboard shortcuts of the application.

As assistive technology also uses additional modifier keys and keyboard shortcuts without modifier keys, it should be possible to redefine or disable application-specific keyboard shortcuts.

Configuring keyboard shortcuts

Regardless of the assistive technology which is used, keyboard shortcuts are important for people who depend on the use of the keyboard. The application should therefore make it possible to define special keyboard shortcuts for all functions, even if they are not provided with any keyboard shortcuts by default. Where this is possible, the Help option should make reference to it.

Documentation of the keyboard shortcuts and shortcut keys

The keyboard shortcuts should be explicitly documented in the application, i.e. in text form at the respective element (permanently visible or in a tooltip), e.g. "Print (CTRL+D)".

The shortcut keys should be implicitly documented in the application, by underlining the letter for example, such as "Print".

The shortcut keys within selection lists and drop-down lists are not generally documented – there, the first letter of the respective list entry acts as the shortcut key.

In the Help option, the keyboard shortcuts should be documented as follows:

- Documentation of the respective keyboard shortcuts within the topic-specific sections (description of a screen, function, elements etc.) as well as
- Documentation of all keyboard shortcuts on a separate page with instructions for the use of the keyboard.

Use of the pointing device

Online betrachten

Synonyms: Use of the mouse, use of the touch pad, use of the pen, pointing device operation, pointer operation

See also: Use of the keyboard

The use of the pointing device encompasses all the operational modalities with a pointing device, e.g.

- Mouse,
- Graphics tablet,
- Touch pad,
- Touch display,
- Track point,
- Track ball,
- Joystick,
- Pen.

Presentation

No.	Property	Description	Classification	Reference
112	Web: Consistency	Control elements that have the same functionality must be designed consistently within the application.	Must	EN 301 549: 9.3.2.4
113	Desktop: Consistency	Control elements that have the same functionality should be designed consistently within the application.	Should	WCAG 2.1: 3.2.4 (AA)

Operation

From the perspective of accessibility, there is no compulsory requirement for an application to be operable with a pointing device. All that is required is the operation with the keyboard. However, if an application can be operated with a pointing device, specific requirements must be met.

No.	Property	Description	Classification	Reference
114	Use of the keyboard	It must be possible to operate the entire application using the keyboard, i.e. all the functions that can be initialized with a pointing device can also be operated using the keyboard.	Must	EN 301 549: 9.2.1.1, 11.2.1.1

No.	Property	Description	Classification	Reference
		Note: This does not apply to path-bound inputs, such as a freehand screen in an image editing program.		
115	Biometry	If biometric data is required for operational purposes (e.g. fingerprint, facial recognition), an alternative method of operation must be made available. Note: The alternative method may also be based on biometric data provided that a different form of biometric data is used for this.	Must	EN 301 549: 5.3

No.	Property	Description	Classification	Reference
116	Complexity	The complex use of the pointing device must be avoided, unless • a non-complex, alternative form of operation is available, • the complex form of operation is essential, • the complex form of operation is for the control of the assistive technology. Please note: Complex use of the pointing device means • multipoint operation (e.g. swiping with several fingers), • path-based operation (where the start and end points of the use of the pointing device aren't just relevant, but at least one intermediate point is).	Must	EN 301 549: 9.2.5.1, 11.2.5.1

No.	Property	Description	Classification	Reference
117	Complexity	The dragging use of the pointing device should be avoided, unless: • a non-complex use of the pointing device without dragging is available, • the complex form of operation is essential, • the complex form of operation is for the control of the assistive technology. Note 1: This applies to sliders and drag and drop functions, for example. Note 2: An alternative use of the keyboard for the dragging use of the pointing device (using a keyboard shortcut, for example) is not sufficient. An alternative use of the pointing device should be available which can also include text input, however.	Should	WCAG 2.2

No.	Property	Description	Classification	Reference
118	Shown content	If additional content is shown when hovering with a pointing device, it must be possible to hide such content again without moving the pointing device away, unless: • the content consists of an error message, or • the content only hides white space or decorative non-text content. . Note 1: This does not apply to unchanged content which is displayed by the platform software by default, such as standard tooltips for the respective programming language. Note 2: The automatically-shown content can be hidden, e.g., with ESC or clicking on the triggering element, as long as no other actions are triggered.	Must	EN 301 549: 9.1.4.13, 11.1.4.13

No.	Property	Description	Classification	Reference
119	Shown content	If additional content is shown when hovering with a pointing device, this content must be displayed until the pointing device is moved away from the triggering element and/or content shown, unless: • the content was deliberately closed (e.g. with ESC) or • the content is no longer valid (e.g. an error message in the input field after inputting a correct value). Note: This does not apply to unchanged content which is displayed by the platform software by default, such as standard tooltips for the respective programming language.	Must	EN 301 549: 9.1.4.13, 11.1.4.13
120	Shown content	If additional content is shown when hovering with a pointing device, it must then be possible to move over this content with the pointing device, i.e. the content may not be hidden as soon as the pointing device is no longer positioned over the triggering element. Note 1: This does not apply to unchanged content which is displayed by the platform software by default, such as standard tooltips for the respective programming language. Note 2: To be able to move over the shown content, the content must be displayed at the triggering element.	Must	EN 301 549: 9.1.4.13, 11.1.4.13

No.	Property	Description	Classification	Reference
121	Cancelling the use of the pointer	During the use of the pointing device, the function of the control element may not be performed during the pressing (down event), but during the actual release (up event) unless: • the triggering of the function during the pressing of the key is essential (an electronic piano or a keyboard simulation, for example), • during the pressing, the function is automatically reversed upon the release, or • the function is part of a more complex function that can be canceled (e.g., picking up a drag-and-drop object by pressing the key on the pointing device and completing the action by releasing it again).	Must	EN 301 549: 9.2.5.2, 11.2.5.2

No.	Property	Description	Classification	Reference
122	Click area	The click area of the control element should be at least 24 x 24 px. Note 1: This does not apply to: • elements with which the offset of the click areas is more than 24 px, • elements that are situated within continuous text (e.g. links), • elements whose size is essential. Note 2: The offset of the click areas is defined as the spacing between the farthest point of one element and the nearest point of the other element. The offset is determined in both directions and must be at least 24 px in each case.	Should	WCAG 2.2

No.	Property	Description	Classification	Reference
123	Click area	The click area of the control element should be at least 44 x 44 px. Note: This does not apply to: • elements whose function can be initialized using an alternative element with sufficient size (44 x 44 px), • elements that are situated within continuous text (e.g. links), • elements whose size is essential.	Should	WCAG 2.1: 2.5.5 (AAA)
124	Different methods of operation	Users should be able to switch between different methods of operation (e.g. operation using the keyboard or operation using the mouse) at any time.	Should	WCAG 2.1: 2.5.6 (AAA)

Use of the keyboard

Online betrachten

Synonyms: keyboard operation, keyboard interface

See also: Use of the pointing device,, fokus indicator, navigation sequence

It must be possible for all functions which can be initialized using a pointing device, motion control or voice input to also be initialized using the keyboard, as disabled users may not be able to use a pointing device and/or may not see the pointer, may not be able to perform the movement or may not be able to speak. It is also possible that disabled users are unable to use a keyboard, but their assistive technology simulates the keyboard and interacts with the keyboard interface of the operating system and/or the Accessibility API.

In this context, the simulation of a pointing device with the keyboard (e.g., keyboard mouse using the numeric keypad) is not considered a permissible alternative form of operation to the use of the pointing device.

Presentation

No.	Property	Description	Classification	Reference
125	Contrast	The contrast requirements must also be met when using the keyboard, e.g. when receiving the focus (see Colors and contrasts).	Must	EN 301 549: 9.1.4.3, 11.1.4.3, 9.1.4.11, 11.1.4.11
126	Focus visibility	If a control element receives the keyboard focus, the focus indicator must be visible (see Fokus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7
127	Focus visibility	The focus indicator must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
128	Web: Consistency	Control elements that have the same functionality must be designed consistently within the application. (see Consistency)	Must	EN 301 549: 9.3.2.4
129	Web: Consistency	Navigation elements must be displayed in the same relative sequence on each page within the application and receive the keyboard focus.	Must	EN 301 549: 9.3.2.3
130	Desktop: Consistency	Control elements that have the same functionality should be designed consistently within the application.	Should	WCAG 2.1: 3.2.4 (AA)
131	Desktop: Consistency	Navigation elements that repeat on several screens should always be displayed in the same sequence and receive the focus.	Should	WCAG 2.1: 3.2.3 (AA)

Operation

No.	Property	Description	Classification	Reference
132	Use of the keyboard	It must be possible to operate the entire application using the keyboard. This does not apply to necessary path-bound inputs, such as a signature or a freehand screen in an image editing program.	Must	EN 301 549: 9.2.1.1, 11.2.1.1

No.	Property	Description	Classification	Reference
		Note 1: An application is operable with the keyboard if all the interactive elements can be accessed and operated with the keyboard. Note 2: If control elements do not receive the keyboard focus, an alternative use of the keyboard for the corresponding functions must be offered.		
133	Use of the keyboard	Path-bound inputs should also be operable with the keyboard.	Should	WCAG 2.1: 2.1.3 (AAA)
134	Consistency	The use of the keyboard should be possible according to the known conventions of the platform software. If the use of the keyboard deviates from these conventions, the users should be informed accordingly. Note: The use of the keyboard for individual elements is described in the “Use of the keyboard” section in this document.	Should	ISO 9241-171: 9.3.15

No.	Property	Description	Classification	Reference
135	Time limits	The use of the keyboard must be possible without specified time constraints. Note: For example, it is not permitted • that a key has to be pressed for a certain time to trigger an action. • that within a certain period of time, two keys must be pressed in succession to trigger an action.	Must	EN 301 549: 9.2.1.1, 11.2.1.1
136	Keyboard trap	The application may not contain any keyboard traps. Note: A keyboard trap is an element of the page that can be accessed using the keyboard but cannot be left again using the keyboard. It must be possible to exit the element using either the standard navigation buttons (such as the tab key, arrow keys, ESC), or users must be informed of the Keyboard shortcuts which allow them to exit.	Must	EN 301 549: 9.2.1.2, 11.2.1.2

No.	Property	Description	Classification	Reference
137	Keyboard shortcuts	Keyboard shortcuts for printable characters without a modifier key may not be used, unless: • the keyboard shortcuts can be disabled, • the keyboard shortcuts used can be redefined so that it is not necessary to use any printable characters, • the keyboard shortcuts only apply if the keyboard focus is on a specific element. Note: Modifier keys include the Alt and Ctrl keys. Printable characters include upper and lower case letters, numbers, punctuation marks, and special characters. Among others, the following keys can be used without a modifier key: ESC, delete, function keys, tab key, enter key, space bar, arrow keys.	Must	EN 301 549: 9.2.1.4, 11.2.1.4
138	Navigation sequence	Bei der Navigation mit der Tastatur muss die Navigationsreihenfolge aufgabenangemessen sein (siehe Navigation sequence).	Must	EN 301 549: 9.2.4.3, 11.2.4.3
139	Navigation sequence	When navigating with the keyboard, the focus order of the work task should be appropriate.	Should	EN 301 549: 9.2.4.3, 11.2.4.3

No.	Property	Description	Classification	Reference
140	Motion control	If it is possible for the application to be controlled by motion, it must be possible to disable the motion control and provide a keyboard alternative to the motion control. Note 1: Motion control encompasses both the movement of the hardware and the movements of the users, which are registered by the software by camera, for example. Note 2: This does not include necessary motion controls, such those with a pedometer or a GPS device.	Must	EN 301 549: 9.2.5.4, 11.2.5.4
141	Biometry	If biometric data is required for operational purposes (e.g. fingerprint, facial recognition), an alternative method of operation must be made available. Note: The alternative method may also be based on biometric data provided that a different form of biometric data is used for this.	Must	EN 301 549: 5.3
142	Change of context	When navigating with the keyboard, no Change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1
143	Change of context	When changing the value of form elements with the keyboard, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.2, 11.3.2.2

No.	Property	Description	Classification	Reference
144	Shown content	If additional content is shown when receiving keyboard focus, it must be possible to hide such content again using the keyboard without moving the keyboard focus away, unless • the content consists of an error message, or • the content only hides white space or decorative non-text content . Note 1: This does not apply to unchanged content which is displayed by the platform software by default, such as standard tooltips for the respective programming language. HNote 2: The automatically-shown content can be hidden with ESC, for example.	Must	EN 301 549: 9.1.4.13, 11.1.4.13
145	Shown content	If additional content is shown when receiving keyboard focus, it must be displayed until the keyboard focus is moved away, unless • the content was deliberately closed (e.g. with ESC) or • the content is no longer valid (e.g. an error message in the input field after inputting a correct value). Note: This does not apply to unchanged content which is displayed by the platform software by default, such as standard tooltips for the respective programming language.	Must	EN 301 549: 9.1.4.13, 11.1.4.13

No.	Property	Description	Classification	Reference
146	Different methods of operation	Users should be able to switch between different methods of operation (e.g. operation using the keyboard or operation using the mouse) at any time.	Should	WCAG 2.1: 2.5.6 (AAA)
147	Web: Efficiency	It must be possible to skip areas of content that occur on several pages using the keyboard (see Practical tip: efficient keyboard navigation).	Must	EN 301 549: 9.2.4.1
148	Efficiency	It must be possible to initialize frequently used functions efficiently with the keyboard. Note: To do this, keyboard shortcuts and context menus can be implemented. The keyboard shortcuts should be documented in the application and the Help option.	Should	DIN EN ISO 9241-171: 9.3.10

Use of the keyboard (general requirements)

Note: The operation of individual elements is described at the respective element.

Note: The use of the keyboard does not generally need to be implemented separately, as it is already made available by the platform software or the framework which is used. Care should be taken to avoid overwriting the keyboard shortcuts for other individual functions, however.

Action	Key	Classification
Navigating to an interactive element, exiting an interactive element	TAB	Required
Reversal of the direction of navigation	SHIFT + [Navigation key] z. B. SHIFT+TAB oder SHIFT+F6	Required
Highlight, select	SHIFT + [Navigation key] z. B. SHIFT+DOWN ARROW oder SHIFT+POS1	Required

Action	Key	Classification
Navigate within interactive elements (e. g. table, tree structure, selection list, radio button group, etc.)	Arrow keys	Required
Enabling of interactive elements	• ENTER • PSACE	Required
Opening the context menu	• CONTEXT MENU • SHIFT+F10	Required
Desktop: Desktop: System menu of the application window	ALT+SAPCE	Required
Quick navigation to start and end	• POS1 • END	Recommended
Quick navigation (skip multiple elements)	• PAGE UP • PAGE DOWN	Recommended
Desktop: Focusing and exiting the main menu	• ALT • F10	Recommended
Desktop: Navigating between application areas	• CTRL+TAB • F6	Recommended
Closing of shown content (such as tooltips, pop-ups, sub-menus)	ESC	Recommended
Select all	CTRL+A	Recommended
Copy selection to clipboard	CTRL+C	Recommended
Cut selection to clipboard	STRG+X	Recommended
Paste from clipboard	CTRL+V	Recommended

Action	Key	Classification
Undo the last action	CTRL+Z	Recommended
Repeat the last action and/or restore the undo	CTRL+Y	Recommended
Delete elements	DEL	Recommended
Desktop: Initialize Help	F1	Recommended
Desktop: Initialize context-sensitive Help	SHIFT+F1	Recommended
Desktop: Closing the application	ALT+F4	Recommended

Programming/interfaces

No.	Property	Description	Classification	Reference
149	Desktop: Operation	All the control options of the element must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.11
150	Operation	It must be possible to run all the control options of the element with the assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.14, 11.5.2.16, 11.5.2.17
151	Keyboard shortcut, shortcut key	Keyboard shortcuts and shortcut keys that are visually perceptible in the application must also be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
152	Position	The focused element and the selected entry within an element must be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.13
153	Desktop: Position	The position of the text cursor must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.13
154	Desktop: Position	The spatial size and position of the elements must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5, 11.5.2.10

Practical tip: use of the keyboard in Web and desktop applications

Custom elements

The standard elements of the marking and/or programming language or of the framework used are generally fully keyboard-operable. They should therefore be used with preference.

If custom elements are used, in terms of the keyboard operability, it is necessary to pay attention to the following in particular:

- Accessibility with the keyboard (e.g. using the tabindex),
- Operability with the keyboard (event handler for the keyboard and/or device-independent event handler),
- Operability conforms to the expectations or has been documented (conforms to expectations regarding the visual presentation and the role, that is communicated to the Accessibility API.
- Focus indicator,
- Focus handling (e.g. no loss of focus during operation; Navigation sequence),
- possibly defined keyboard shortcuts are consistent with those of the platform software and/or do not override those of the platform software.

If a custom element is implemented, it is often recommended that a related default element is used and customized accordingly, as it is then possible to use the basic functionality of the default element.

Drag & Drop

Dragging and dropping elements (drag and drop) can only be done with a pointing device. Designing the alternative form of operation with the keyboard so that it can be used efficiently and in accordance with expectations is recommended. Possible variants include

- Kontextmenü (e.g. for moving elements to other areas),
- several buttons, grouped in one toolbar (e.g. for changing the sequence of elements within an area),
- a button (e.g. for uploading files),
- individual keyboard shortcuts (e.g. adjustment of the size of elements),
- keyboard shortcuts of the platform (e.g. cutting and pasting elements using CTRL+X and CTRL+V),
- use of the arrow keys (e.g. slider),
- Start and finish with the ENTER key, move with the arrow keys (e.g. drawing a freehand screen).

A combination of the variants is often a good idea (e.g. buttons and keyboard shortcuts).

As the alternative form of operation with the keyboard is not usually visible, it should be described in the application and the Help option.

Control elements that are not permanently visible

Control elements that are shown and hidden when using the application, e.g.

- because they are in tooltips, or
- because they only appear when the keyboard focus is in a particular position,

are not usually operable with the keyboard.

Displaying control elements all the time and avoiding control elements in tooltips, for example, is recommended.

Alternatively, an alternative form of operation must be implemented using the keyboard and described in the Help option and the application. In addition to this, the application must be designed so that it is perceptible with the assistive technology when elements that are not always visible are shown. Blind users, for example, must be able to perceive that a tooltip contains control elements with the screen reader, so that they can use the documented alternative form of operation with the keyboard.

User preferences

Online betrachten

Synonyms: User settings, customization, individual adaptation, adaptation to preferences, user preferences

See also: Colors and contrasts,, font, Focus indicator

Presentation

No.	Property	Description	Classification	Reference
155	User preferences	The application must comply with the platform settings for measurement units, color, contrast, font style, font size, and focus pointer (mouse and text cursors, and focus indicator), unless overridden by users. Note 1: This does not apply to applications that are isolated from the platform and do not have access to the platform settings. Note 2: The application is also able to offer an alternative mode that does not apply the platform settings. Note 3: In the case of Web applications, the browser is the platform whose settings are to be applied. The browser is also able to accept settings from the operating system.	Must	EN 301 549: 11.7

No.	Property	Description	Classification	Reference
156	User preferences	If users change the platform settings for color, contrast, font style and font size, all content must be displayed correctly, and all features must be functional. Note 1: This means, for example, that after adjusting the font style or font size, the text content is displayed in full and without overlay. Note 2: If users have adjusted the colors or contrasts and the selected colors have been applied correctly by the application, the application is not responsible for complying with the contrast ratios (see Practical tip: contrast adjustment).	Must	EN 301 549: 11.7, 9.1.4.3, 11.1.4.3, 9.1.4.4, 11.1.4.4.1, 9.1.4.5, 11.1.4.5.1, 9.1.4.10, 11.1.4.10, 9.1.4.11, 11.1.4.11, 9.1.4.12, 11.1.4.12
157	User preferences	It should be possible to make the following settings for text blocks: • The foreground and background colors can be adapted to the user requirements without adjusting the other colors of the application. • The length of a line of text can be limited to 80 characters. • The justify text setting can be disabled. • The line spacing can be increased to 150% of the standard line spacing. • The paragraph spacing can be increased to 1.5 times the line spacing.	Should	WCAG 2.1: 1.4.8 (AAA)

Practical tip: contrast adjustment for desktop applications

In Windows, users can customize the presentation of the colors to suit their needs (Settings > Ease of Access > High contrast). Unlike other operating systems which only allow certain color changes (such as coloring, darkening or inverting the colors), Windows allows you to freely choose the foreground and background colors. In addition to this, it is possible to define individual colors for different element types and/or their status.

With Windows Contrast Adjustment, it is possible to adjust the colors for the following elements and states:

- Text (SystemColorWindowTextColor)
- Background (SystemColorWindowColor)
- Links (SystemColorHotlightColor)
- Highlighted text, selected element (SystemColorHighlightTextColor)
- Background of highlighted text or selected elements (SystemColorHighlightColor)
- Text of control elements (außer Links) (SystemColorButtonTextColor)
- Background of control elements (außer Links) (SystemColorButtonFaceColor)
- Disabled elements (SystemColorGrayTextColor)

For Windows Contrast Adjustment to work correctly in an application, the following must be considered:

- The foreground and background colors must be defined in such a way that they can be overwritten with Windows Contrast Adjustment. This does not apply to graphics or videos.
- Content whose foreground color is adjusted may not have a background whose color is not adjusted. This applies, for example, to text whose background is a graphic that is not adjusted. This does not apply if the font has a contour in the background color, as the visibility is then ensured through the contour.
- Graphics with a transparent background must be avoided. This does not apply if either the foreground color of the graphic can be adjusted (by using SystemColorWindowTextColor, for example), or if the content of graphic has a contour.
- For the element types Text, Links and Form Elements as well as the states Disabled, Highlighted and Selected, the corresponding information about the role and status must be communicated to the Accessibility API. Alternatively, the appropriate colors for the elements and states must be used (for example, SystemColorGrayTextColor for disabled elements).
- Images of text must be avoided. Alternatively, the foreground and background colors of the images of text must be adjustable according to the Windows settings.
- Only color-coded contents must be avoided, regardless of the contrast between the colors.
- Page areas, tooltips, pop-ups and control elements that are shown in different background colors should be given a frame so that they also stand out from the background with Windows Contrast Adjustment.
- If the application does not support Windows Contrast Adjustment, the application must offer its own possibility for adjusting the colors with the equivalent functionality. This option must be described in the application and the Help option.
- If your application only supports Windows Contrast Adjustment for a specific configuration (for example, selecting a specific color scheme), it must be described in the application and the Help option.

Example 1: The two following figures show two lines with two pieces of text, two buttons and two links – one in the default presentation (figure on the left), and one with the use of Windows Contrast Adjustment (Contrast no. 1).

- In the figure on the left, the text, button and link are visible in both lines based on the colors used:
  - Text: black font on white background,
  - Button: black font on gray background,
  - Link: blue font on white background.
- In the figure on the right, the text, button and link can only be identified correctly in the top line, as they have been programmatically marked as text, button and link:
  - Text: yellow font on black background,
  - Button: white font on black background,
  - Link: blue font on black background.
- In the figure on the right, the text, button, and link in the bottom row are indistinguishable, as they are all displayed in the text color (yellow font on a black background). The reason for the incorrect presentation is that the button and link were not programmatically displayed as such.

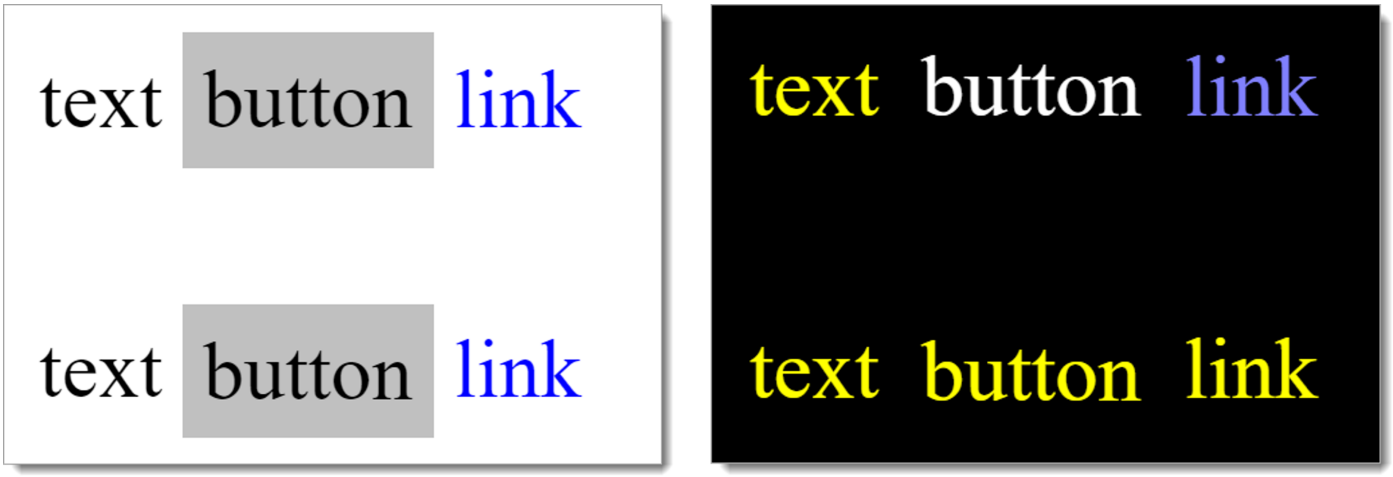

Figure 6: Text, button and link in the default presentation (left) and with the use of Windows Contrast Adjustment (right).

Example 2: In the two following figures, one operable button (“Send”) and two disabled buttons (“Check” and “Delete”) are shown – one in the default presentation (the figure on the left), and one with the use of Windows Contrast Adjustment (Contrast no. 1).

- In the standard presentation, the two disabled buttons can be seen to be disabled due to the grayed-out colors.
- When using Windows Contrast Adjustment, only the “Check” button is correctly visually identifiable as disabled (green font on black background).
- The “Delete” button is not identifiable as disabled when using Windows Contrast Adjustment, but is shown as operable (white font on a black background). The reason for the incorrect presentation is that it was not programmatically marked as disabled.

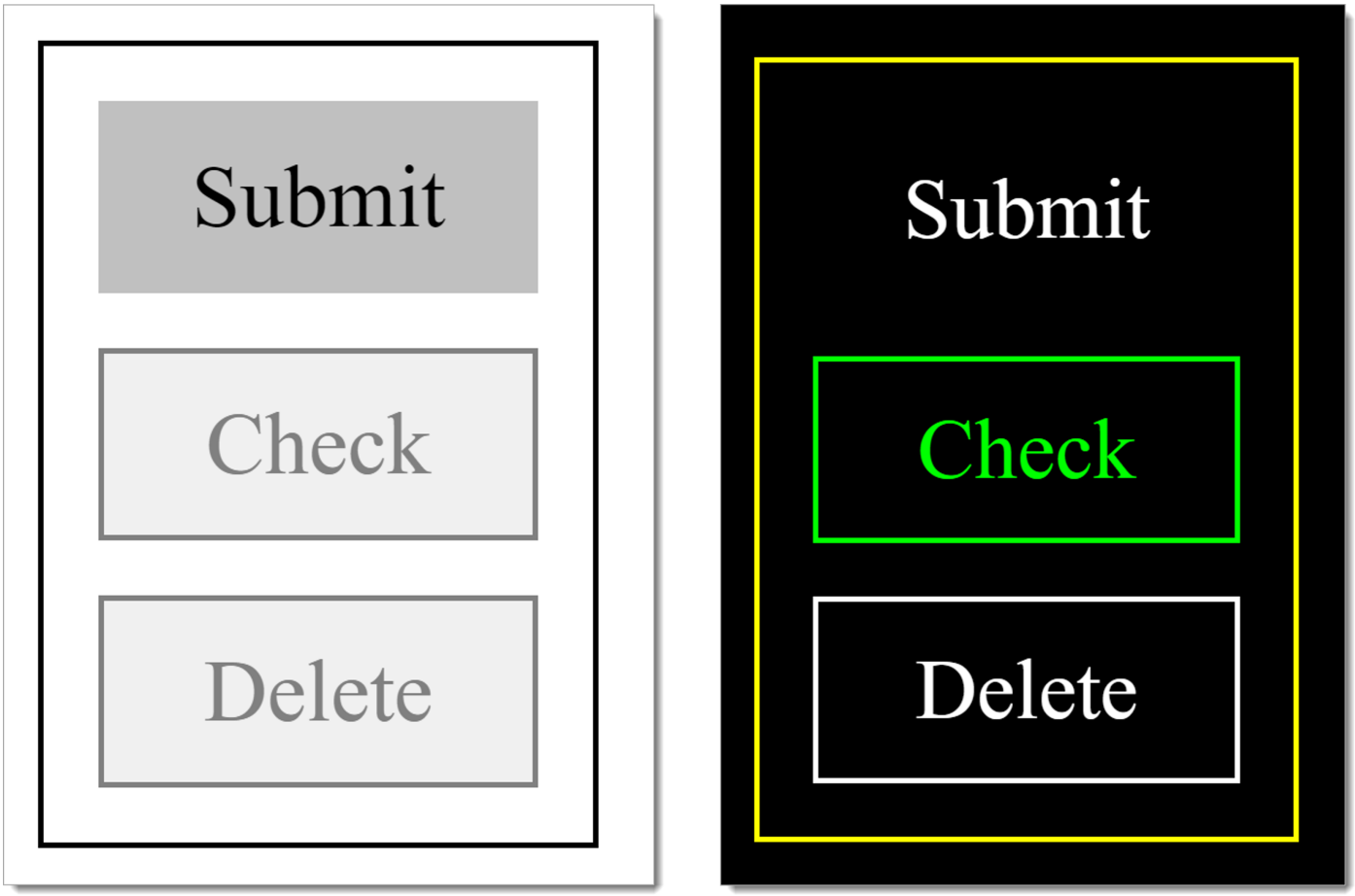

Figure 7: One operable and two disabled buttons in the default presentation (left) and with the use of Windows Contrast Adjustment (right).

Example 3: In the two following figures, four buttons are shown that are labeled with a black icon – one in the default presentation (figure on the left), and one with the use of Windows Contrast Adjustment (Contrast no. 1).

- In the default presentation all four buttons look the same (black icon on white background).
- When using Windows Contrast Adjustment, the four buttons look different because of the different technology used for the presentation of the icon:
  - The first icon is displayed correctly because a font icon was used in which the foreground and background colors are adjusted (white icon on black background). This variant is to be preferred.
  - The second icon is not adjusted (black icon on white background) because it is a graphic without a transparent background. This variant is acceptable.
  - The third icon is not so easy to identify because it is a graphic with a transparent background (black icon on a black background). The icon is perceptible due to its white outline, however. This variant is acceptable.
  - The fourth icon cannot be seen because it is a graphic with a transparent background (black icon on a black background). This variant should not be used.

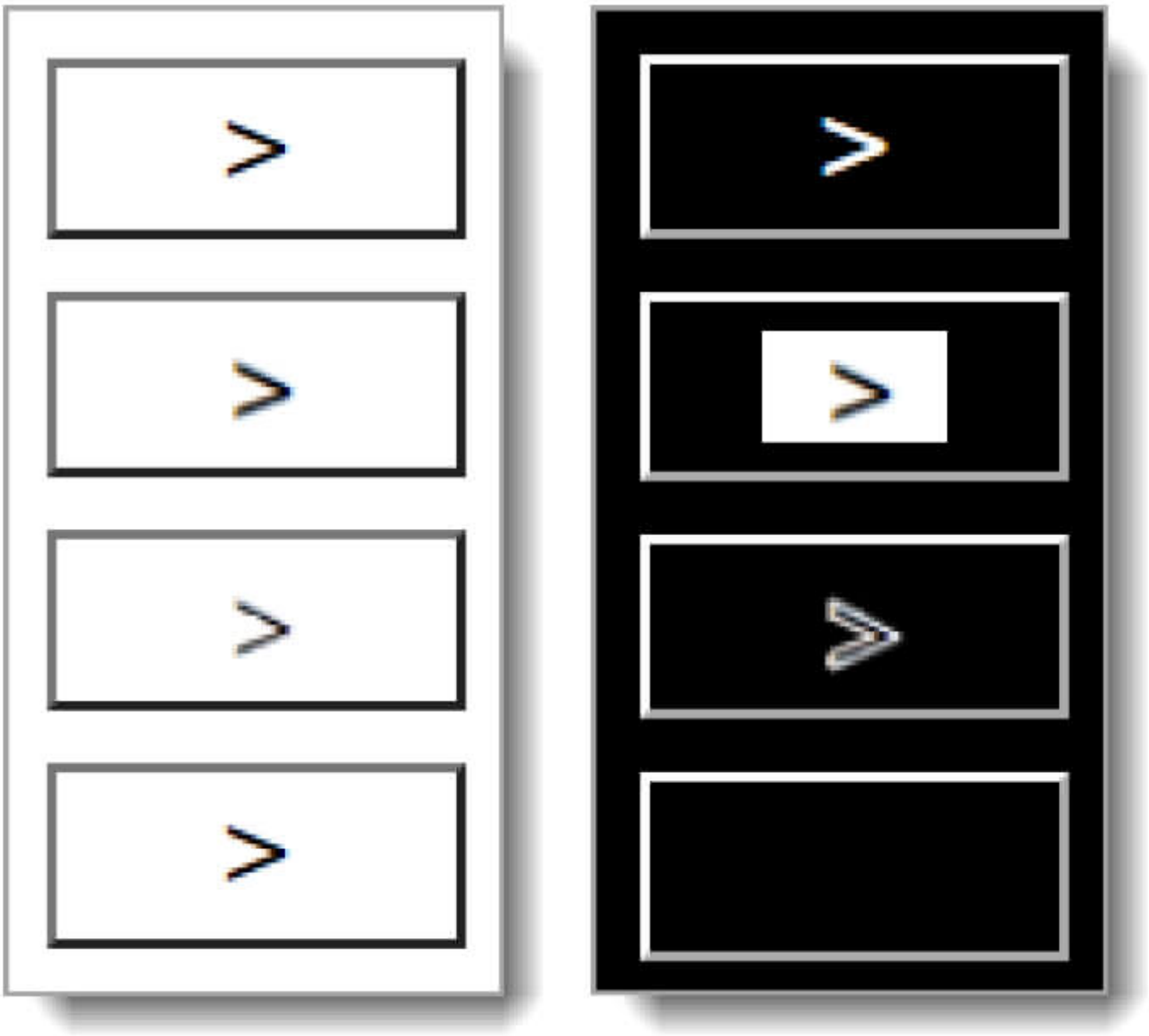

Figure 8: Four icon buttons in the default presentation (left) and with the use of Windows Contrast Adjustment (right).

Colors and contrasts

Online betrachten

Synonyms: Color coding, color, contrast

See also: text, font, graphics, contrast adjustment

Colors are an important visual design tool. People with impaired vision may not perceive colors or differences in color (contrasts), however.

Presentation

No.	Property	Description	Classification	Reference
158	Text contrast	All text content must have a contrast ratio of at least 4.5:1 with respect to the background. Note 1: With large font (from 24 px and/or from 18.7 px in the case of bold font), a contrast ratio of at least 3:1 is sufficient. Note 2: The contrast requirements do not apply to • disabled elements if they do not communicate any information, • purely decorative text content, • invisible text content, • incidental text content in figures that are not relevant to the understanding of the figure,, • logos and brand names insofar as they are not used as control elements. Note 3: We recommend that you always use a plain-colored background for text and no graphics or color gradients.	Must	EN 301 549: 11.1.4.3

No.	Property	Description	Classification	Reference
159	Text contrast	All text content must have a contrast ratio of at least 7:1 with respect to the background. Note: With large font (from 24 px and/or from 18.7 px in the case of bold font), a contrast ratio of at least 4.5:1 is sufficient.	Should	WCAG 2.1: 1.4.6 (AAA)

No.	Property	Description	Classification	Reference
160	Graphics contrast	All content of graphics must have a contrast ratio of at least 3:1. This applies to the contrast of the graphic with the background as well as to the contrasts within the graphic (between neighboring areas) insofar as they are relevant to the understanding of the graphic. Note 1: The contrast requirements do not apply to • purely decorative graphics, • graphics whose information is also available in text form, • certain photographs (e.g. of people and landscapes), • screenshots, • figures in which the colors used are specified, with heat maps and medical charts, for example, • logos and flags, insofar as they are not used as the only label for control elements. Note 2: If the contrast between neighboring colors is not sufficient within a graphic, for the purpose of visual differentiation, • a contour can be inserted, or • the area can be provided with hatching, insofar as the contour and/or hatching have a contrast ratio of at least 3:1.	Must	EN 301 549: 11.1.4.1, 11.1.4.11

No.	Property	Description	Classification	Reference
161	Contrast of information to status and type	All information necessary for identifying the type or status of a control element must have a contrast ratio of at least 3:1 to the background or neighboring colors. Note 1: This relates to the border of form fields, the focus indicator and to a graphical element that identifies a selected option (within a selection list, menu, table, etc). Note 2: The contrast requirements do not apply to • disabled elements if they do not communicate any information, • standard elements whose presentation has not been changed.	Must	EN 301 549: 11.1.4.11
162	Contrast	The contrast ratios must be maintained in each status of the element, e.g. upon receiving the focus with the keyboard, when hovering with a pointing device, and during use and/or enabling with the keyboard or pointing device. Note: When hovering or during operation with a pointing device, the elements can, but do not have to change their appearance (e.g. text or background color). A clearly visible focus indicator is only necessaryduring the keyboard navigation.	Must	EN 301 549: 11.1.4.3, 11.1.4.11

No.	Property	Description	Classification	Reference
163	Anti-aliasing	Since the anti-aliasing influences the presentation of the defined colors, care should be taken to ensure sufficient line thicknesses and/or contrast ratios (see Practical tip: anti-aliasing)	Should	
164	Color coding	If information is communicated by using different colors, then all colors (with respect to each other) must have a contrast ratio of at least 3:1. Note: This applies if the colors themselves have no meaning, but only the color difference does.	Must	EN 301 549: 11.1.4.1
165	Color coding	If information is communicated using a particular color, this information must also be communicated in another way. Note: This applies when the color itself has a meaning, such as “green” for correct and “red” for incorrect, or “black” for a positive number and “red” for a negative number.	Must	EN 301 549: 11.1.4.1
166	Color coding	Color coding should be avoided. Note: Even if a contrast ratio of at least 3:1 is maintained, the color-coded information may no longer be visible when using Windows contrast adjustment .	Should	EN 301 549: 11.1.3.1
167	User preferences	For block text, it must be possible for the foreground and background colors to be adapted to the user requirements without adjusting the other colors of the application.	Should	WCAG 2.1: 1.4.8 (AAA)

Programming/interfaces

No.	Property	Description	Classification	Reference
168	Color coding	Information conveyed using color or color differences must be communicated to the Accessibility API programmatically or in text form. Note: Color information can present the status of a control element, for example (disabled, incorrect, selected, see Element status and Operability status) and be programmatically communicated. The color information, in contrast, can be communicated in a diagram in text form (diagram data as a table).	Must	EN 301 549: 11.1.3.1, 11.1.4.1

Practical tip: color coding

To comply with EN 301 549: 9.1.4.1 and 11.1.4.1, respectively, a distinction must be made between two different forms of color coding:

- the color difference conveys a piece of information,
- the color itself conveys a piece of information.

The color difference conveys a piece of information

Examples:

- The active menu item has a gray background color, all the other menu items have a white background color.
- The text color is black and the link color is blue.

If the colors themselves have no meaning, but only the difference between the colors does, a contrast ratio of at least 3:1 between the colors is sufficient.

The use of another visual method, however, (e.g. icon, font style, size, position, shape, border, underline, text) to reproduce the information conveyed by the color difference is recommended, e.g.

- The active menu item has a bold font style, all other menu items have a normal font style.
- The text is not underlined, but all links are underlined.

Note: EN 301 549: 9.1.4.1 and 11.1.4.1 formulate requirements for visually impaired people who use the application without assistive technology. For users of assistive technology, according to EN 301 549: 9.1.3.1 and 11.1.3.1.1, additional requirements apply with regard to the color coding, e.g.

- To enable blind people to perceive the information conveyed by color difference, this information must be communicated to the Accessibility API programmatically or in text form.
- To enable users of the Windows Contrast Adjustment to perceive the information conveyed by color difference, this information must be conveyed programmatically (see Practical tip: contrast adjustment), by another visual means (see above), or in text form.

The color conveys a piece of information

Examples:

- Incorrectly filled out form fields have a red border, correctly filled out form fields have a green border.
- Different background colors are used for messages: red for warning messages, green for success messages and yellow for tips.
- In a financial statement, losses are shown with a red-colored text and profits with black-colored text.

In these cases, the color itself provides a piece of information, and not just the difference between the colors. Therefore, a contrast ratio of at least 3:1 between colors is not sufficient in this case. Another visual method must certainly be used to convey the information, e.g.

- The incorrectly filled out form fields have an error icon (e.g. an exclamation mark in a red circle) and the correctly filled out form fields have a success icon (e.g. a check mark in a green circle). An expressive error message is also displayed in text form in order to comply with EN 301 549, 9.3.3.1 or 11.3.3.1.
- The messages each have a heading that conveys the status (e.g. “Warning”, “Success” and “Tip”).
- In the financial statement, losses are shown with a minus sign in front of the number.

Note: EN 301 549: 11.1.4.1 formulates requirements for visually impaired people who use the application without assistive technology. For users of assistive technology, according to EN 301 549: 11.1.3.1.1, additional requirements apply with regard to the color coding (see above).

Practical tip: anti-aliasing

Regarding the contrast requirements, the WCAG assumes that anti-aliasing has been disabled. However, this is unrealistic, as most users work with anti-aliasing enabled. Due to the anti-aliasing, in case of thin line thicknesses, the actual contrasts can be significantly below the contrasts that are calculated from the color values. Therefore, either the line thickness or the colors should be adjusted in such a way that the contrast ratio (of at least 4.5:1 for text and 3:1 for content of graphics) can also be maintained when anti-aliasing is enabled. Because the anti-aliasing can be configured differently, it is not possible to specify exact values. However, in most cases the following guideline values will help to ensure that the contrast ratios can be maintained:

- minimum line thickness of 2 px or
- doubled contrast ratio at lower line thicknesses (e.g. at least 9:1 instead of 4.5:1).

In this context, it should be noted that a line thickness of 2 px is only achieved for letters above a certain font size; from 25 px for Arial and from 75 px for Times New Roman.

For example: In the following two figures, the letter “x” and a forward slash (“/”) are shown three times in the font style “Times New Roman” and with the font size 16 px. The contrast between the gray and/or black text color defined in the CSS and the white background is:

- Left: 4,5:1,
- Center: 7:1,
- Right: 21:1.

This means that the required minimum contrasts will be maintained throughout the text. In the considerably magnified presentation, however, it can be seen that due to the anti-aliasing, the contrasts calculated for the “x” are only partially achieved with the thicker downward line (diagonal line from the top left to the bottom right) and the horizontal serifs. For the thinner upward line (diagonal line from bottom left to top right), the contrast is clearly below this, e.g. at the most in the lower range at:

- Left: 2,9:1,
- Center: 2,9:1,
- Right: 5,2:1.

For the forward slash, the calculated contrasts are not achieved at any point.

- Left: maximum 2,4:1, average approx.. 2,0:1,
- Center: maximum 2,5:1, average approx. 2,1:1,
- Right: maximum 3,5:1, average approx. 2,3:1.

x/ x/ x/

Figure 9: Effect of anti-aliasing on contrasts using the example of “x/” in normal size.

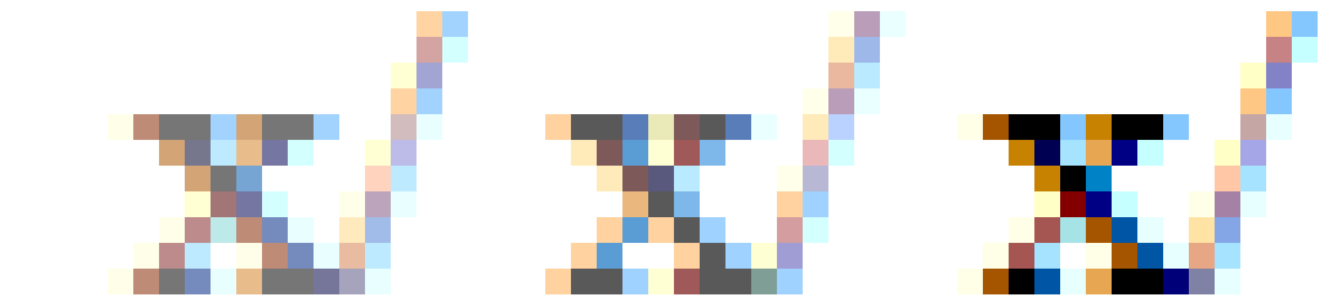

Figure 10: Effect of anti-aliasing on contrasts using the example of “x/” with resizing.

Font

Online betrachten

Synonyme: Letters, characters, numbers, font

See also: Text, contrast, label, heading

The font is used for the presentation of text information.

Presentation

No.	Property	Description	Classification	Reference
169	User preferences	The application must accept the font style, size, and color settings from the platform software or provide a mode in which the settings are applied. Note 1: If the settings of the platform software are not applied automatically, the corresponding mode must be explained in the information on accessibility. Note 2: The application can also offer a mode in which users can choose their preferences for the font style, size, color and any other possible font attributes directly in the application. Note 3: The requirements concerning the contrasts only apply as long as the users have not adjusted the colors to their requirements.	Must	EN 301 549: 11.7 und 12.1.1

No.	Property	Description	Classification	Reference
170	User preferences	It should be possible to make the following settings for text blocks: • The foreground and background colors can be adapted to the user requirements without adjusting the other colors of the application. • The length of a line of text can be limited to 80 characters. • The justify text setting can be disabled. • The line spacing can be increased to 150% of the standard line spacing. • The paragraph spacing can be increased to 1.5 times the line spacing.	Should	WCAG 2.1: 1.4.8 (AAA)
171	Contrast	All text content must have a contrast ratio of at least 4.5:1 with respect to the background. Note: With large font (from 24 px and/or from 18.7 px in the case of bold font), a contrast ratio of at least 3:1 is sufficient. See Colors and contrasts .	Must	EN 301 549: 9.1.4.3, 11.1.4.3
172	Color	If information is communicated using different font colors, the contrast ratio between the colors must be at least 3:1 (see Practical tip: color coding). Note: This applies if the colors themselves have no meaning, but only the color difference does.	Must	EN 301 549: 9.1.4.1, 11.1.4.1

No.	Property	Description	Classification	Reference
173	Color	If information is conveyed using a particular font color, this information must also be conveyed in another way (see Practical tip: color coding). Note: This applies when the color itself has a meaning, such as “green” for correct and “red” for incorrect, or “black” for a positive number and “red” for a negative number.	Must	EN 301 549: 9.1.4.1, 11.1.4.1
174	Spacing	If users are able to adjust the spacing between the lines, paragraphs, letters and words, then in this context, no content or functions may be lost. Note: This applies to the following spacing: • line spacing up to 1.5 times the font size, • paragraph spacing up to 2 times the font size, • character spacing up to 0.12 times the font size, • word spacing up to 0.16 times the font size.	Must	EN 301 549: 9.1.4.12, 11.1.4.12
175	Spacing	The line spacing of continuous text should be 1.5 times the font size.	Should	WCAG 2.1: 1.4.8 (AAA)
176	Abstand	The paragraph spacing for continuous text should be 1.5 times the size of the line spacing, i. e. 2.25 times the font size.	Should	WCAG 2.1: 1.4.8 (AAA)

No.	Property	Description	Classification	Reference
177	Reference to sensory properties	Information which serves the understanding or the operation of the application may not make exclusive reference to the written formatting of the described elements.	Must	EN 301 549: 9.1.3.3, 11.1.3.3
178	Line length	A line of text in continuous text should not exceed 80 characters.	Should	WCAG 2.1: 1.4.8 (AAA)
179	Orientation	Justify should be avoided in continuous text. Note: Justify is the orientation of the text to the left and right margins.	Should	WCAG 2.1: 1.4.8 (AAA)

Programming/interfaces

No.	Property	Description	Classification	Reference
180	Character set	To encode the font characters, a character set must be used whose characters can be output correctly by the assistive technology for the text-to-speech output. Note: Currently, only letters that occur in the application language should be used, as other letters are not usually supported.	Must	EN 301 549: 9.1.3.1, 11.1.3.1

No.	Property	Description	Classification	Reference
181	Special characters	Special characters may only be used if they are output correctly by the assistive technology. Note: This applies to font icons and ligations, for example. Alternatively, these special characters must be marked so that they are ignored by the assistive technology. The information conveyed by the characters must then be communicated in text form or programmatically (see Practical tip: special characters).	Must	EN 301 549: 9.1.1.1, 11.1.1.1, 9.1.3.1, 11.1.3.1
182	Hyphenation	The hyphenation requires the use of a character which is not output by the assistive technology. Otherwise, the hyphenation must be dispensed with. Note: This does not apply if the hyphenation has to be perceptible, e.g. in a dictionary in which the possible hyphenations are specified.	Must	EN 301 549: 9.1.3.2, 11.1.3.2
183	Spaces, punctuation marks	The word boundary must be perceptible, with the use of a space, hyphen or punctuation mark, for example.	Must	EN 301 549: 9.1.3.2, 11.1.3.2
184	Spaces	A word may not contain any spaces or line breaks.	Must	EN 301 549: 9.1.3.2, 11.1.3.2

No.	Property	Description	Classification	Reference
185	Formatting	If font formatting is used to communicate information, this information must also be communicated to the Accessibility API in text form or programmatically. Note: For example, an important piece of text which is highlighted in bold can also be preceded with “Note: ” or given a separate heading.	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 9.1.4.1, 11.1.4.1, 11.5.2.10

Practical tip: special characters

When using special characters, different use cases must be distinguished between in terms of the communication of the characters to the Accessibility API:

Decorative special characters

Purely decorative special characters are marked so that they are not communicated to the Accessibility API. The same rules apply to them as to layout graphics.

Examples:

- Button with the label “Next »”: The two forward arrows are purely decorative. The button should have the Accessible Name “Next”.
- Form field label “- - Location - -”: The dashes are purely decorative. The form field should have the Accessible Name “Location”.

Non-purpose-related special characters that contain content

Special characters which are not used according to their meaning are to be accompanied by an expressive alternative text. The same rules apply to them as to graphics.

Examples:

- Button with the label »: The two forward arrows visually convey the information “Next”. The button should have the Accessible Name “Next”.
- Button with the label x: The multiplication symbol visually conveys the information “Close”. The button should have the Accessible Name “Close” (possibly with an indication of what is being closed, e.g. “Close window”).

Note: If used as a required field label, the asterisk (“*”) is not considered to be non-purpose-related.

Purpose-related special characters that contain content

Special characters which are used according to their meaning can be used, provided the character is output correctly using the assistive technology. Otherwise, it should be accompanied by expressive alternative text.

Examples:

- Mathematical formula “3+5 > 5-3”: The two arithmetic operators as well as the larger character are output correctly by the assistive technology and can therefore be used.
- Text with the ordinal characters “ª” and “º”: The assistive technology does not output the two characters at all and/or does not do so correctly, and therefore the two characters require an alternative text.
- The word “guest” with a ligation of s and t. Most screen readers do not know the ligation, and instead of “guest”, only state “gu” and a space.



Figure 11: The word “guest” with a ligation

Special-purpose characters should be used sparingly. It is recommended to use a character which is output in concise form by the assistive technology.

Examples:

- A quote can be given different quotation marks. The “ character is output by the assistive technology, for example, as “typographical quotation mark, top” or “typographical quotation mark, right”. The " character, by contrast, is output as “inverted commas” or “quotation mark”, and is therefore to be preferred.
- The sentence “I prefer the character »#« to “#”” (see the following figure) is output by the screen reader, for example, as “I prefer the double bracket black hashtag with white X double bracket character to lower typographical quotation mark 3D narrow right-pointing arrow head inverted commas”.



Figure 12: Sentence with special characters is output by the screen reader in a difficult to understand way

Practical tip: font style and text formatting

To make texts easy to read, a legible font style and text formatting should be chosen. In this respect, the following should be taken into account:

- The most readable fonts are those from the humanist sans serif category.
- Ligations should be avoided.
- The minimum font size for text blocks should be 22px.
- The minimum font size for additional text (e.g. footnotes) should be 17px.
- Narrow and wide fonts should be avoided.
- Small and large character spacing should be avoided.
- Thin and thick font styles should be avoided.
- The line spacing should be at least 120% of the font size.
- Upper and lower case lettering should be used according to the orthography rules. The exclusive use of upper or lower case lettering should be avoided.
- Upper and lower case lettering should be used according to the orthography rules. The exclusive use of upper or lower case lettering should be avoided.
- Small caps should be avoided.
- Highlighting should be used sparingly. A bold font style should be used for highlighting.
- Italic font styles or underlining should be avoided. Note: Links should be underlined.
- The text should be left-aligned. The use of justify should be avoided.

You can find more information at: https://www.leserlich.info (External Link)

.

Focus indicator

Online betrachten

Synonyms: Focus, focus frame, focus appearance

See also: Use of the keyboard, text cursor

The focus indicator displays which element currently has the keyboard focus (see DIN EN ISO 9241-161: 8.37).

The focus indicator is usually displayed with a border around the focused element. Other focus indicators are also permitted, provided that they meet the requirements, e.g.

- Inverting of foreground and background color,
- Changed background color,
- Changing the size of the element,
- Superimposing of a graphical element, such as a side bar.

In certain cases, several focus indicators can be displayed:

- Example 1: When a selection list receives the focus, a focus indicator can be displayed around the entire list. A focus indicator must also be displayed with the current list entry.
- Example 2: With a combined input field, the focus can be on the input field as well as on the selection list.
- Example 3: If an interactive element receives the focus within a page area, then the page area can also be shown as focused.
- Example 4: The title bars of all application windows that do not have the focus are shown grayed out.

In some cases, the focus indicator can be identical to the selection marker (i.e. the marking of the selected option, see Element status), if the focused element is identical to the selected element:

- Example 1: A selection list without multiple selection has a focus indicator on the focused list entry. This indicator can serve as a selection marker at the same time, as the focused list entry is identical to the selected list entry. If this selection list does not also have a focus indicator for the entire list, however, to be able to detect whether the selection list is focused, it is necessary to ensure that the selection marker in the focused state of the selection list differs clearly from the selection marker in the non-focused state of the selection list (e.g. minimum contrast ratio 3:1).
- Example 2: A selection list without multiple selection has a focus indicator on the focused list entry, e.g. a border. This focus indicator is not used simultaneously as a selection marker. A different background color, for example, is used as the selection marker. When navigating through the list entries, both the focus indicator and the selection marker are moved. In this case, an additional focus indicator for the entire list or a differentiation of the selection marker in the focused and non-focused state is not necessary, as the focusing and selection can be perceived independently of each other.
- Beispiel 3: A multiple selection list has a focus indicator that surrounds the entire list. Since the focused list entry does not match the selected list entries in the case of the multiple selection list, the focus indicator and selection marker must be marked separately for the list entries (e.g., as shown in example 2 with a border and a different background color).

Presentation

No.	Property	Description	Classification	Reference
186	General	The focus indicator must be visible at each navigation step.	Must	EN 301 549: 9.2.4.7, 11.2.4.7
187	Contrast	The focus indicator must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 11.1.4.11
188	Contrast	The elements must also have a contrast ratio of at least 4.5:1 for text and at least 3:1 for the content of graphics in the focused status.	Must	EN 301 549: 9.1.4.3, 11.1.4.3, 9.1.4.11, 11.1.4.11
189	Consistency	The focus indicator should be clearly assignable to the focused element.	Must	EN 301 549: 9.2.4.7, 11.2.4.7
190	Visibility	The element must be scrolled into the visible area on receiving the focus so that both the element and its focus indicator are visible. Note: This also applies when navigating the arrow keys through the list entries of a selection list, for example.	Must	EN 301 549: 11.2.4.7
191	Size	The area of the focus indicator shall be at least as large as: • 1px times the circumference of the element or • 4px times the length of the shorter side.	Should	WCAG 2.2

Operation

Use of the keyboard: focus indicator

The change of the focus indicator between the elements is described in the use of the keyboard and navigation sequence sections. By default, the navigation takes place with the TAB key.

The change of the focus indicator within an element is described at the respective elements. The navigation within elements often takes place with the arrow keys.

Use of the pointing device: focus indicator

Action	Key	Classification
Placing the focus	Left click	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
192	Position	The focused element must be communicated to the Accessibility API .	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.13, 11.5.2.15
193	Desktop: Position	The size and position of the focused element must be communicated to the Accessibility API. Note: This is important, for example, to ensure that screen magnifiers can display the focused element in the visible area and a focus highlighting.	Must	EN 301 549: 11.5.2.5, 11.5.2.13
194	Operation	It must be possible to position the focus with assistive technology (see Use of the keyboard).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.14, 11.5.2.16

Practical tip: focus indicator

The focus indicator must have sufficient contrast, and should be designed consistently at the same time. With applications that have different background colors, this can be achieved by using a two-color border (e.g. black and 05.15.2023 Page 68 of 317 white) that has sufficient contrast against each background. A two-color focus indicator is also useful with control elements on gradients or graphics.

For example: Two buttons on a color gradient from white to black. The top button is currently focused. The focus indicator consists of a black border (inside) and a white border (outside), which means that it always easy to see, regardless of the background color. It is the standard focus frame offered by Google Chrome.

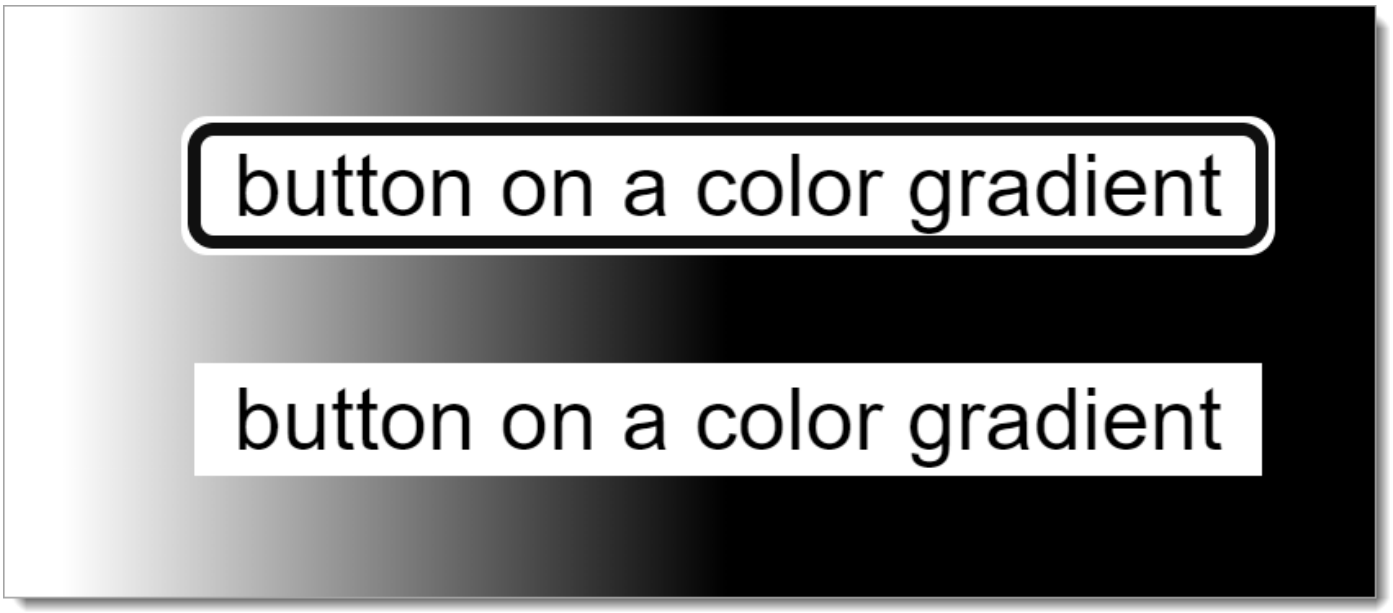

Figure 13: Two-color focus indicator

Text cursor

Online betrachten

Synonyms: Cursor

See also: Use of the keyboard, Focus indicator

The text cursor displays the position in an input field (see DIN EN ISO 9241-161: 8.8). The text cursor is typically displayed by a vertical line at the point at which the text is entered, edited or deleted.

Presentation

No.	Property	Description	Classification	Reference
195	User preferences	The application must adopt the text cursor setting from the platform software and/or provide a mode in which this setting is adopted. Note: If the settings of the platform software are not applied automatically, the corresponding mode must be explained in the information on accessibility.	Must	EN 301 549: 11.7, 12.1.1

Operation

Use of the keyboard: text cursor

The change of the focus indicator between the elements is described at the input field and multi-line input field elements. By default, the navigation within the input fields takes place with the arrow keys.

Use of the pointing device: text cursor

Action	key	Classification
Placing the focus	left click	required

Programming/interfaces

No.	Property	Description	Classification	Reference
196	Desktop: Position	The position of the text cursor must be communicated to the Accessibility API .	Must	EN 301 549: 11.5.2.13
197	Desktop: Operation	It must be possible to position the text cursor with assistive technology (see using the keyboard).	Must	EN 301 549: 11.5.2.14

Required field label

Online betrachten

Synonyms: Required form fields, required input, required

See also: error message, label, element status

A required field label is a visual indicator for form fields that have to be filled out. In this respect, for example, an asterisk (*) can be used to indicate that a form field has to be filled out.

Presentation

No.	Property	Description	Classification	Reference
198	Presentation	Required fields must be visually perceptible. Note 1: The optional fields can also be labeled instead of the required fields. Note 2: If the required fields are also identifiable from the context without a label (e.g. on a login page with two input fields for the user name and password), then the required field label can be omitted. Note 3: Required fields must be labeled consistently in the application. Note 4: In the case of Gruppen of radio buttons and check boxes, the required field label should be situated at the label of the group insofar as a random element of the group has to be selected.	Must	EN 301 549: 9.3.3.2, 11.3.3.2
199	Contrast	The required field label in graphic form must have a contrast ratio of at least 3:1 with respect to the background. The required field label in text form must have a contrast ratio of at least 4.5:1 with respect to the background.	Must	EN 301 549: 9.1.4.3, 11.1.4.3, 9.1.4.11, 11.1.4.11
200	Color	Required fields should not only be labeled with the use of color (e.g. a different border color). Note: Color can be used as an additional means of the required field label.	Must	EN 301 549: 9.1.4.1, 11.1.4.1

Programming/interfaces

No.	Property	Description	Classification	Reference
201	Status	The required field status must be communicated to the Accessibility API (see Accessibility API). Note 1: The programmatic required field labeling should take place using an attribute of the programming language provided for this purpose or using the textual supplement in the Accessible Name of the form field (e.g. asterisk). The redundant programmatic marking of the required fields with attribute and supplement in the Accessible Name label should be avoided. Note 2: If filling out a form field group is labeled as required without the necessity for every field of the group to be filled out, the programmatic required field labeling should only take place at the group and not at every field. In this case, care must be taken to ensure that the required field label of the group is communicated correctly to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 9.3.3.2, 11.3.3.2, 9.4.1.2, 11.4.1.2

Practical tip: required field label

Required fields can be labeled with a corresponding textual supplement, such as “required field” or “required”. If the majority of the fields are required, the fields that do not need to be filled out can alternatively be labeled with “optional”, for example.

The required fields can also be marked with a symbol. The asterisk (“*”) has established itself for this purpose, in which it can be assumed, at least in specialist applications, that it is known to all users. If another character is used, its meaning should be explained at the beginning of the form.

Practical tip: programmatic label of required fields in Web applications

Screen reader output

- JAWS: required invalid entry | required [after the role and before the value]
- NVDA: required invalid entry | required [after the role and before the value]
- Windows Narrator: required invalid | required [after the value]

Note: A required field which is not filled out but which has been marked with required is in the incorrect status due to the HTML specification, and is therefore output by the screen readers as “invalid entry” and /or “invalid”. The problem does not occur when using aria-required.

HTML

Required fields can be marked with the required attribute..

- Form fields that have been marked as required are automatically validated by the browser and will have an incorrect status if they have not been filled out.
- Form fields that have been marked as required are not automatically visually identifiable as required fields. They have a tooltip, (e.g. “Please complete this field”), which is not perceptible to users of the keyboard and with the screen reader, however, as it is only displayed in the case of a mouse over.
- By default, a form which has required fields that have not been filled out cannot be submitted. Instead, the focus is placed in the first incorrect field and the tooltip is displayed as an error message and output by the screen reader as a warning message (analogous to role=alertalert). These error messages are not accessible for the following reasons:
    - When exiting the field, the error message is hidden automatically and cannot be displayed again.
    - Several browsers automatically hide error messages after a few seconds, even if the focus is in the field (with Chrome and Edge, for example).
    - The error message is only displayed at the first incorrect field. Other required fields that are not filled out are not to be perceived as incorrect.

Therefore, when using required, the following should be considered:

- Required fields must also be visually labeled as such.
- The visual required field label should be marked so that it is not output by the screen reader to avoid the redundant output.
- The application should continuously display its own error messages for all required fields that have not been filled out.

Note: In the practical tip on radio buttons and checkboxes , the particularities regarding their marking up as required fields are explained.

Further information 4.10.5.3.4 The required attribute - HTML Standard (whatwg.org)

ARIA

Required fields can be marked with the aria-required=true attribute.

- Form fields that have been marked with aria-required=trueare not automatically validated by the browser and will not have an incorrect status if they have not been filled out. The incorrect status can be communicated with aria-invalid=true .
- Form fields that have been marked with aria-required=true are not automatically visually identifiable as required fields.

Further information: aria-required property - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)

Text elements

Online betrachten

Table of contents

- Headings
- Label
- Description
- Text

Headings

Online betrachten

Synonyme: Section heading, main heading, heading

See also: font, text, label, titel, group

Headings are used to outline sections of text. They describe the content of each section of text.

Presentation

No.	Property	Description	Classification	Reference
202	Contrast	Headings must have a contrast ratio of at least 4.5:1 and/or 3:1. Note: From a font size of 24 px (and/or 18.5 px in the case of bold font), a contrast ratio of at least 3:1 is sufficient.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
204	Label	The heading must be expressive. Note: To achieve this, the heading should describe the following section concisely and unambiguously.	Must	EN 301 549: 9.2.4.6, 11.2.4.6
205	Label	An image of text may not be used for the heading, unless its text content can be adapted to the user preferences (font style, font size, font color, background color).	Must	EN 301 549: 9.1.4.5, 11.1.4.5.1
206	Structure	Sections of text should be subdivided with headings.	Should	WCAG 2.1: 2.4.10 (AAA)
207	Hierarchy	The hierarchy of headings must correspond to the logical structure of the page. Note: As a rule, the page should have a main heading with the highest hierarchy. Section headings should be organized in the correct hierarchical way without skipping a hierarchy level.	Must	EN 301 549: 9.1.3.1, 11.1.3.1.1
208	Focus visibility	If the heading receives the focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
209	Use of the keyboard	In applications that do not support the virtual cursor , it must be possible to access and exit the heading with the keyboard (see Use of the keyboard table). Note: If the application contains several headings that receive the keyboard focus, there should be an operating mode in which only the interactive elements receive the focus to avoid unnecessary navigation steps for sighted keyboard users.	Must	EN 301 549: 9.1.1.1, 11.1.1.1

Use of the keyboard: heading (in an application without a virtual cursor)

Aktion	Taste	Klassifizierung
Focusing the heading	TAB	required
Exiting the heading	TAB	required

Programming/interfaces

No.	Property	Description	Classification	Reference
210	Role	The “heading” role and its hierarchy level must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 11.5.2.5
211	Hierarchy	The hierarchy of headings must correspond to the logical structure of the page. Note: In this context, the maximum number of hierarchy levels should be considered (desktop applications: typically 9, Web applications: 6).	Must	EN 301 549: 9.1.3.1, 11.1.3.1
212	Desktop: Position	The size and position of the heading must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: headings in Web applications

Screenreader output

- JAWS: Heading level [number] [label]
- NVDA: Heading level [number] [label]
- Windows Narrator: Heading level [number] [label]

HTML

Headings are marked with the HTML elements <h1>, <h2>, <h3>, <h4>, <h5> and <h6>. In this respect, the following should be taken into account:

- Headings should be concise, clear and expressive, as for users of screen readers, they are the primary method of navigating and perceiving the page structure.
- The headings should be nested correctly from a hierarchical perspective, i.e. an <h1> should be followed by <h2>-headings whose sections can in turn contain <h3> headings. According to the HTML specification, no heading levels may be skipped, for example, <h2> should not be followed by <h4>, <h5> oder <h6>.
- The main heading (<h1>) should describe the purpose of the respective page (and not just include the application name, for example).

Headings can be grouped within the <hgroup> elements along with paragraphs (<p>-Element) to specify a caption or an alternative heading, for example. This grouping is not perceptible with any of the screen readers for Windows, i.e. the content of the <p> element is output as normal text, as according to the “HTML Accessibility API Mappings”, the <hgroup> element does not have any semantics.

Further information: 4.3.6 The h1, h2, h3, h4, h5, and h6 elements - HTML Standard (whatwg.org) , 4.3.11 Headings and outlines - HTML Standard (whatwg.org) (External Link)
, Headings | Web Accessibility Initiative (WAI) | W3C

ARIA

If the heading is not implemented with the HTML elements, it is also necessary to take account of the following:

- The role is communicated with role=heading.
- The heading level is communicated with aria-level Only the numbers from 1 to 6 should be used. With numbers greater than 6, JAWS outputs the heading with the level 2.
    - With numbers greater than 6, JAWS outputs the heading with the level 2.
    - With numbers greater than 9, NVDA and Windows Narrator output the heading with the level 2.
- The labeling should take place using text content.

Further information: heading role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)

Label

Online betrachten

Synonyms: Designation, form field label, name, accessible name

See also: font, text, graphik, form, group, description, error message, required field label

Labels are used for the identification of control elements (see DIN EN ISO 9241-161: 8.21).

A label consists of a short, descriptive text or an expressive graphic and/or a combination of text and graphic. The label can be situated inside the element or adjacent to the element which is labeled.



Figure 14: Label before an input field

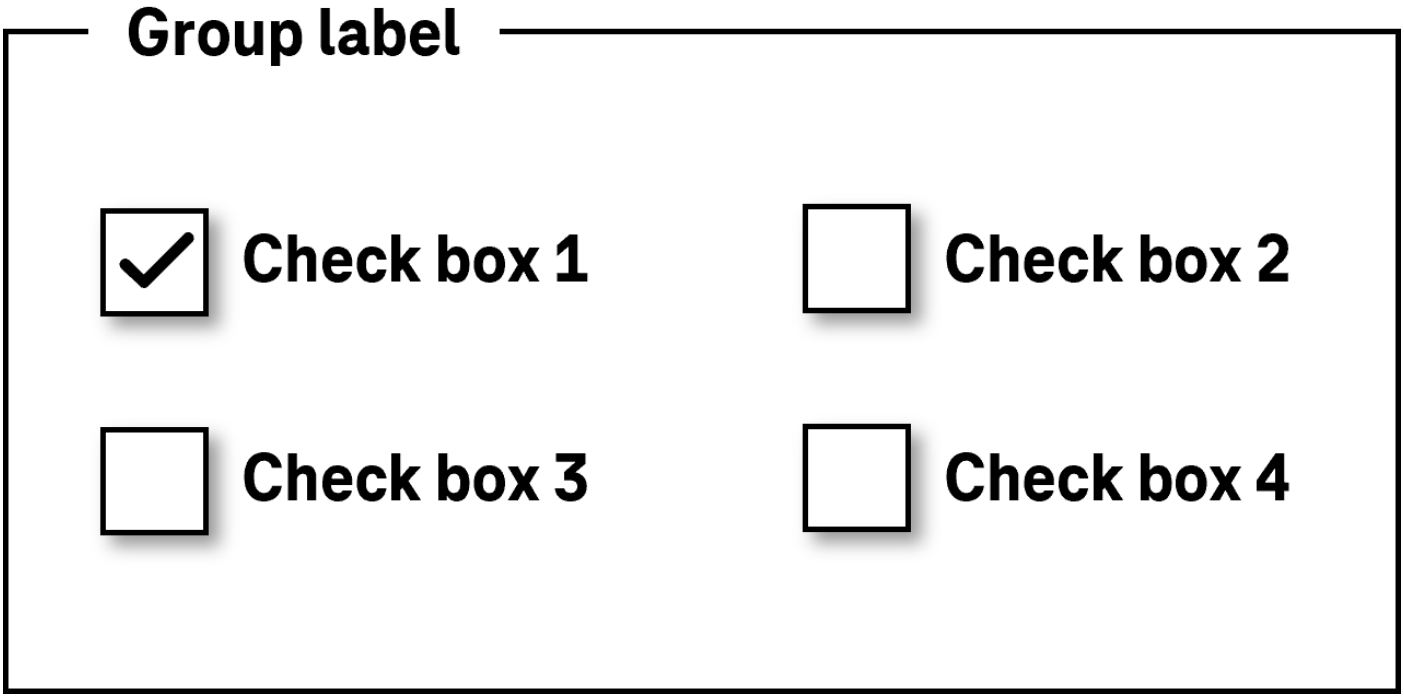

Figure 15: Labels and a group label placed to the right of check boxes

Presentation

No.	Property	Description	Classification	Reference
213	Contrast	Text labels must have a contrast ratio of at least 4.5:1 with respect to the background. Note 1: With large font (from 24 px and/or from 18.7 px in the case of bold font), a contrast ratio of at least 3:1 is sufficient.	Must	EN 301 549: 9.1.4.3, 11.1.4.3

No.	Property	Description	Classification	Reference
		Note 2: The contrast requirements also apply when receiving keyboard focus (keyboard focus indicator) or hovering with a pointing device.		
214	Contrast	Text labels must have a contrast ratio of at least 7:1 with respect to the background. Note: With large font (from 24 px and/or from 18.7 px in the case of bold font), a contrast ratio of at least 4.5:1 is sufficient.	Should	WCAG 2.1: 1.4.6 (AAA)
215	Contrast	Graphic labels must have a contrast ratio of at least 3:1. This applies to the contrast with the background as well as to the areas of content within the graphic. This also applies when the form field has focus and when hovering with a pointing device. Note: This does not apply to layout graphics , i.e. if an equivalent text label exists in addition to the graphic	Must	EN 301 549: 9.1.4.11, 11.1.4.11
216	Color coding	If information is conveyed using the color of the label (e.g. the form field is a required field or filled out incorrectly), this information must also be conveyed in another way, e. g. in text form.	Must	EN 301 549: 9.1.4.1, 11.1.4.1, 9.3.3.1, 11.3.3.1
217	Resizing	It must be possible to scale the label by up to 200%. During the scaling, the label must remain completely visible and may not obscure other areas of the page or be obscured by them.	Must	EN 301 549: 9.1.4.4, 11.1.4.4.1

No.	Property	Description	Classification	Reference
218	Resizing	The label must be displayed in full without horizontal scrolling at a screen width of 320 px.	Must	EN 301 549: 9.1.4.10, 11.1.4.10
219	Graphic	The label may not contain any (images of text) , unless they are adaptable to the user requirements (font style, font size, font color, background color).	Must	EN 301 549: 9.1.4.5, 11.1.4.5.1
220	Graphic	The label should not contain any images of text.	Should	WCAG 2.1: 1.4.9 (AAA)
221	Comprehensibility	The label must be expressive (see Practical tip: special characters).). Note 1: To achieve this, the label should be concise and unambiguous. Note 2: In addition to the concise label, more detailed descriptions can be used. Note 3: If an element only has graphic label, a tooltip should be provided with the text alternative.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.3.3.2, 11.3.3.2
222	Comprehensibility	The use of abbreviations in labels should be avoided. Alternatively, a mechanism should be available for displaying the unabbreviated form and/or the meaning of the abbreviation.	Should	WCAG 2.1: 3.1.4 (AAA)
223	Comprehensibility	The label should be in the application language.	Should	WCAG 2.1: 3.1.3 (AAA), 3.1.5 (AAA)

No.	Property	Description	Classification	Reference
224	Context	A visual label must be available on every form field. Alternatively, the purpose of the form field must be clear from the context (e.g. unlabeled search field with “Search” button; unlabeled form fields in a table with expressive column and line headings).	Must	EN 301 549: 9.3.3.2, 11.3.3.2
225	Position	The form field label should be located to the left or above the form field, except for the case of radio buttons and check boxes. Note: In exceptional cases, the label can also enclose the form field (e.g. “Reminder in [input field] days”). However, we recommend that the label is formulated so that it can only be located in front of the form field.	Should	DIN EN ISO 9241-143: 5.3.4, 5.3.8 DIN EN ISO 9241-125: 5.1.14
226	Position	The form field labeling of radio buttons and checkboxes should take place to the right of the form field.	Should	DIN EN ISO 9241-143: 5.3.8 DIN EN ISO 9241-125: 5.1.14
227	Web: Consistency	Labels must be used consistently within the application.	Must	EN 301 549: 9.3.2.4
228	Desktop: Consistency	Labels should be used consistently within the application.	Should	WCAG 2.1: 3.2.4 (AA)
229	Animation	The label may not sparkle, flash or be visually changed in any other way (see Animation).	Must	EN 301 549: 9.2.3.1, 11.2.3.1, 9.2.2.2, 11.2.2.2

No.	Property	Description	Classification	Reference
230	Animation	The label should be displayed all the time and should not be animated during operation. Note: This means that the label should not be displayed within a form field to become invisible or be positioned adjacent to the field during inputting.	Should	WCAG 2.1: 2.3.3 (AAA)
231	Keyboard shortcut, shortcut keys	If an control element has a keyboard shortcut or a shortcut key, it should be visible on the label. Note: For the identification of a shortcut key, the corresponding letter can be underlined. Keyboard shortcuts can be displayed behind the label or in a tooltip.	Should	DIN EN ISO 9241-171: 9.3.11
232	Focus visibility	If the label receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Use of the pointing device: labeling

Action	Key	Classification
Enabling of the element if the label is situated in the element	Left click on the label	Required
Enabling of the element if the label is situated adjacent to the element	Left click on the label Note: This is especially true for form fields with a small click area, such as radio buttons and check boxes.	Recommended

Programming/interfaces

No.	Property	Description	Classification	Reference
233	Label	Each control element must have an Accessible Name which is communicated to the Accessibility API. Note 1: This can be achieved if the control element • is given a text label, • is linked to the visual text label adjacent to the element, or • if a graphical label in the element receives an alternative text. Note 2: This also applies if the element has no visual label because its purpose is inherent to the context. The Accessible Name should not change during operation. Exception: If the value or status of a control element is communicated as part of the Accessible Name because the value or status cannot be communicated programmatically, the Accessible Name may change. A button can therefore have the label “Text color red” or “Text color green” and/or “Show information” or “Hide information” (see Practical tip: Accessibility API).	Must	EN 301 549: 9.1.4.2, 11.4.1.2, 11.5.2.5
234	Desktop: Label	If the visual label is not situated in the control element, the label must be programmatically linked to the control element.	Must	EN 301 549: 11.5.2.8

No.	Property	Description	Classification	Reference
235	Desktop: Label	If the visual label is not situated in the control element, the label should be programmatically linked to the control element.	Should	DIN EN ISO 9241-143: 5.3.2
236	Label	If the purpose of a form element is not clearly identifiable from its Accessible Name, but is clear to sighted users from the visual context, the visual information must be communicated to the assistive technology • as part of the Accessible Name, • as (part of) the Accessible Name of a group or • as (part of) the Accessible Description.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
237	Label	If information is conveyed on the basis of the visual design of the label such as the color, font style or font size (e.g. the form field is a required field or incorrectly filled out), this information must be communicated to the Accessibility API programmatically or in text form.	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 9.3.3.1, 11.3.3.1
238	Graphic	For a graphical label only, the Accessible Name must contain an equivalent text alternative which describes the function. Note: Graphical labels include characters and letters with a graphic meaning such as “x” (for Close), “?” (for Help) or “>” (for Continue).	Must	EN 301 549: 9.1.1.1, 11.1.1.1
239	Graphic	If a visual label contains both text and a graphic and the graphic does not communicate any additional information, the graphic must be marked as a layout graphic so	Must	EN 301 549: 9.1.1.1, 11.1.1.1

No.	Property	Description	Classification	Reference
		that it is ignored by the assistive technology.		
240	Comprehensibility	The Accessible Name must be expressive. Note 1: To achieve this, the Accessible Name should be concise and unambiguous.. Note 2: In addition to the concise Accessible Name, more detailed Accessible Descriptions can be used.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.3.3.2, 11.3.3.2
241	Comprehensibility	Abbreviations in the Accessible Name should be avoided. Alternatively, a mechanism should be available for displaying the unabbreviated form and/or the meaning of the abbreviation with assistive technology.	Should	WCAG 2.1: 3.1.4 (AAA)
242	Web: Change of language	If the Accessible Name contains foreign-language terms, the change of language must be marked. Note: The Accessible Name should contain only words in the application language. Even if the change of language is marked, in the case of Accessible Names of interactive elements, it is often the case that it is not output correctly by the assistive technology.	Should	WCAG 2.1: 3.1.4 (AAA)
243	Desktop: Comprehensibility	The Accessible Name should contain only words in the application language.	Should	WCAG 2.1: 3.1.2 (AA)

No.	Property	Description	Classification	Reference
		Note: In the applications in which marking up the change of language is possible, the language of foreign-language Accessible Names should be marked accordingly.		
244	Consistency	The visual label must match or be included in the Accessible Name. Note 1: This only applies if the visible label consists of a text label or an image of text . Note 2: If an element with a purely graphical label has a tooltip which contains a label in text form, then the tooltip text should match or be contained in the Accessible Name.	Must	EN 301 549: 11.2.5.3
245	Keyboard shortcut, shortcut keys	If a control element has a visible keyboard shortcut or a shortcut key, this must be communicated to the Accessibility API. Note: This should take place using the corresponding characteristic of the API. If this is not possible, the keyboard shortcut or the shortcut key can be communicated as part of the Accessible Name or the Accessible Description.	Must	EN 301 549: 11.1.3.1

Practical tip: labeling in Web applications

Screenreader output

- JAWS: **[Label]** [role] [required field note] [validation note] [value] [description] [error message] [screen reader operating note] [keyboard shortcut note]
- NVDA: **[Label]** [role] [required field note] [validation note] [description] [keyboard shortcut note] [value]

- Windows Sprachausgabe: **[Label]** [role] [value] [required field note] [validation note] [keyboard shortcut note]

Please note:

- The label is output by the screen readers first, immediately before the role.
- Only one group label is output before the label, if available.
- During the reading with the virtual cursor, the label is output at the element itself or where it is located in the source code, depending on the control element. To ensure that labels can also be assigned to the control elements correctly in the case of reading with the virtual cursor, the label should be located in the source code in the element (e.g. with links and buttons), directly in front of the element (with form fields other than radio buttons and check boxes) or directly after the element (with radio buttons and check boxes).

HTML

In HTML, the labeling method depends on the element type:

- Links, buttons (that are marked with the <button element>are labeled using their text content or the alternative text of the graphic,
- Buttons (that are marked with the <input> element) are labeled using the value attribute.
- Form elements are labeled with the <label> element.
- Form field groups (<fieldset>) are labeled with the <legend> element.
- Graphics (<img>) are labeled with the alt attribute.
- Figures (<figure>) are labeled with the <figcaption> element.
- Tables are labeled with the <caption> element.
- iFrames and regions (e.g. <nav> and <section>) are labeled with the title attribute. Certain elements may not be labeled (e.g. or).

Further information: 4.10.4 The label element - HTML Standard (whatwg.org), Providing Accessible Names and Descriptions | APG | WAI | W3C, Labeling Controls | Web Accessibility Initiative (WAI) | W3C, 4. Accessible Name and Description Computation - HTML Accessibility API Mappings 1.0 (w3.org)

ARIA

In ARIA, it is possible to communicate labels with the attributes aria-labelledby and aria-label.

- With aria-labelledby reference can be made to the IDs of visible or invisible labels.
- With aria-label the label can be specified directly in text form.
- Certain elements that are marked with ARIA roles can also be labeled with their text content. This also applies to the button, link, checkbox, radio, option and tab. roles. In the ARIA specification, these elements are marked with “Name From: content”.
- Certain elements that are marked with ARIA roles must be explicitly labeled (generally with aria-label or aria-labelledby) Labeling using the text content is not possible. This also applies to the following roles: listbox, combobox, dialog, form, application, grid.In the ARIA specification, elements that have to be labeled are marked with “Accessible Name Required: True”. Whether an element can be explicitly labeled can be seen in the ARIA specification at “Name From: author”.
- A label can also be marked with role=caption caption, and then serves the purpose of the labeling of tables (role=grid, role=table, role=treegrid) and figures (role=figure), as long as the label is the first (or with role=figure the final) child element of the element to be labeled. Regardless of the marking with role=caption , reference to the label should be made with aria-labelledby.

- Certain ARIA roles cannot be labeled, such as generic, paragraph, presentation, code, insertion, deletion, emphasis, strong, subscript and superscript. In the ARIA specification, these elements are marked with “Name From: prohibited”. Elements with these roles may contain text, however.
- Reference can be made to hidden elements using aria-labelledby , e.g. those that have been marked with display:none or hidden . The contents of the hidden elements are used to label the element with aria-labelledby. In such cases, however, the use of aria-label is recommended..

Further information: aria-label property - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, aria-labelledby property - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, caption role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)

General information

- Each control element must have a label which is communicated to the Accessibility API as the Accessible Name. This also applies if the element has no visible label.
- The label should be concise and expressive.
- The Accessible Name should not contain any change of language or structured content, as this is not perceptible with the assistive technology.
- For each element, it is only possible to communicate a label to the Accessibility API using one method. If several methods are used, only the label whose method has the highest priority is used as the Accessible Name. The priority of the methods is defined as follows:
    1. aria-labelledby,
    2. aria-label,
    3. HTML labeling methods (e.g. <label>, <caption>, value, alt, if applicable to the element),
    4. text content (if permissible for the element),
    5. title,
    6. placeholder (only for input fields).
- Elements should not be labeled using the title- or placeholderattribute, as these two attributes, which are actually intended for the communication of a description and/or an input instruction, are used only as a label if no correct label is available. This is a repair mechanism which is designed to ensure that an element is not output without a label.
- If the labeling of a control element occurs explicitly using an attribute (for example, aria-label, aria-labelledby, id, which are referenced via <label for>the attribute must be situated on the element that receives keyboard focus and has the role of the element to be labeled. The control element cannot be labeled by attribute if the attribute is situated on a parent or child element.
- Text content which is shown before or after as pseudo-elements using the CSS selectors are taken into account when determining the label. To avoid this, they should be marked with aria-hidden=true .
- If a reference is made to a form field using aria-labelledby, the value of the form field (not its label, however) is taken into account when determining the labeling of the element with aria-labelledby.
- ARIA elements must generally be labeled with an ARIA method (e.g. aria-label or aria-labelledby) or using their text content (if permissible for the respective role). In this way, for example, an ARIA check box cannot be labeled with the <label>element and an ARIA graphic cannot be labeled with the alt attribute. An exception is presented by ARIA elements whose underlying HTML elements allow for the appropriate HTML labeling method. Accordingly, a combined input field (<input role=combobox>) can be labeled with the <label> element because the HTML element <input> can be labeled with the <label> element.
- The visible label should not be marked with aria-hidden=true true, even if an Accessible Name is available, to ensure that the text-to-speech works using the mouse.

- The visible label should be used as an Accessible Name to avoid a repeated output of the label with the text-to-speech feature.
- For form fields, the Accessible Name in the source code should be immediately in front of the element, except for radio buttons and check boxes, in which case the Accessible Name in the source code should be immediately behind it. This is important, to ensure that the label can be assigned to the form fields correctly when reading with the virtual cursor of the screen reader. # Depending on the type of element, the text content of elements is also imperceptible with the virtual cursor of the screen reader if the elements are explicitly labeled (e.g. with aria-label or aria-labelledby).
    - In the case of control elements (such as links and buttons), the text content should be used as a label. It must otherwise be ensured that all the information contained in the text content is also available in the Accessible Name.
    - In the case of grouping elements (such as form field groups, regions, lists, or tables), their contents are perceptible with the virtual cursor, even if they are explicitly labeled. These elements should therefore be labeled explicitly so that their purpose is recognizable.

Description

Online betrachten

Synonyms: Note, Help, Operation note, Input instruction, Instruction, Accessible Description

See also: font, text, label, error message, required field label, tooltip, help and support

A description contains additional information on the use and operation of the application (see DIN EN ISO 9241-161: 8.19).

Descriptions can refer to a control element, an area, a screen, or to the entire application. A description consists of an explanatory text, a graphic or a combination of text and graphic.

Descriptions can be

- permanently visible,
- are dynamically shown and hidden during operation (e.g. when hovering with a pointing device, when receiving focus with the keyboard or after enabling a control element).

Presentation

No.	Property	Description	Classification	Reference
246	Contrast	Text descriptions must have a contrast ratio of at least 4.5:1 with respect to the background. Note: With large font (from 24 px and/or from 18.7 px in the case of bold font), a contrast ratio of at least 3:1 is sufficient.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
247	Contrast	Text descriptions should have a contrast ratio of at least 7:1. Hinweis: Note: With large font (from 24 px and/or from 18.7 px in the case of bold font), a contrast ratio of at least 4.5:1 is sufficient.	Should	WCAG 2.1: 1.4.6 (AAA)
248	Contrast	Graphic descriptions must have a contrast ratio of at least 3:1. This applies to the contrast with the background as well as to the areas of content within the graphic.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

No.	Property	Description	Classification	Reference
249	Resizing	It must be possible to scale the description by up to 200%. During the scaling, the description must remain completely visible and may not obscure other areas of the page or be obscured by them.	Must	EN 301 549: 9.1.4.4, 11.1.4.4
250	Resizing	The description must be displayed in full without horizontal scrolling at a screen width of 320 px.	Must	EN 301 549: 9.1.4.10, 11.1.4.10
251	Comprehensibility	If additional instructions are required to be able to understand the operation, descriptions with operating instructions must be displayed. Note: Descriptions are not necessary if the labels of the control elements are sufficient. Descriptions may be necessary, for example, for an input field at which a particular input format is required.	Must	EN 301 549: 9.3.3.2, 11.3.3.2
252	Comprehensibility	The description should be formulated in the application language.	Should	EN 301 549: 9.3.1.2
253	Reference to sensory properties	Information in the description which relates to the elements of the application must not refer exclusively to their sensory properties. Note: Therefore, for example, a button should not be described by its appearance or position, but by its label.	Must	EN 301 549: 9.1.3.3, 11.1.3.3

No.	Property	Description	Classification	Reference
254	Position	The descriptions should be positioned so that they can be unambiguously associated with the elements or areas to which they refer. Note: Descriptions for a form field, for example, can be displayed to the right or below the field.	Should	DIN EN ISO 9241-125: 5.1.1, 5.1.14
255	Animation	The description may not sparkle, flash or be visually changed in any other way (see Animation).	Must	EN 301 549: EN 301 549: 9.2.3.1, 11.2.3.1, 9.2.2.2, 11.2.2.2
256	Focus visibility	If the description receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
257	Use of the keyboard	In applications that do not support the virtual cursor , the descriptions must receive the keyboard focus as long as the description is not linked with a keyboard-focusable element.	Must	EN 301 549: 9.1.3.1, 11.1.3.1

No.	Property	Description	Classification	Reference
		Note: If the application contains several descriptions that receive the keyboard focus, there should be an operating mode in which only the interactive elements receive the focus to avoid unnecessary navigation steps for sighted keyboard users..		
258	Use of the keyboard	If the description is not permanently visible, it must also be possible to display it with the use of the keyboard. Note: This applies regardless of how the description is displayed. Frequent variants include: • button to show an area with the description, • button to show a tooltip with the description, • area or tooltip which is shown when hovering using a pointing device or when focusing, • display using a keyboard shortcut.	Must	EN 301 549: 9.2.1.1, 11.2.1.1

No.	Property	Description	Classification	Reference
259	Use of the keyboard	If the description contains control elements, they must be operable with the keyboard. Note: This also applies if the description is displayed in a tooltip .	Must	EN 301 549: 9.2.1.1, 11.2.1.1
260	Use of the keyboard	If a description is displayed in an automatically-displayed tooltip, the following must be met: • It must be possible to close the tooltip with the keyboard without moving the keyboard focus away (e.g. with ESC). • The tooltip must remain visible until it is explicitly hidden (by moving the keyboard focus away, for example). Hinweis: Note: This does not apply to tooltips that are shown by the platform software by default.	Must	EN 301 549: 9.1.4.13, 11.1.4.13

No.	Property	Description	Classification	Reference
261	Use of the pointing device	If a description is displayed in an automatically-displayed tooltip, the following must be met: • It must be possible to close the tooltip with the pointing device without moving the pointing device away. • It must be possible to move over the tooltip with the pointing device without the tooltip being hidden. • The tooltip must remain visible until it is explicitly hidden (by moving the pointing device away, for example). Note: This does not apply to tooltips that are shown by the platform software by default.	Must	EN 301 549: 9.1.4.13, 11.1.4.13

Programming/interfaces

No.	Property	Description	Classification	Reference
262	Role	If the description receives the keyboard focus, an appropriate role (e.g. “text”) must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
263	Desktop: Description	Any available visual description that refers to a keyboard-focusable control element must be communicated to the Accessibility API as an Accessible Description of this element. Note: In this respect, it does not matter whether this description is permanently visible or is only displayed when hovering with a pointing device or when receiving the keyboard focus.	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 11.5.2.5
263	Desktop: Description	If the description is long or contains structured content, the description should be given keyboard focus and its content can be read using the virtual cursor	Should	EN 301 549: 9.1.3.1, 11.1.3.1
264	Graphic	If the description contains a graphic that contains content, its equivalent text alternative must be communicated to the Accessibility API.	Must	EN 301 549: 9.1.1.1, 11.1.1.1

Practical tip: descriptions in Web applications

Screenreader output

- JAWS: [label] [role] [required field note] [validation note] [value] **[description]** [error message] [screen reader operating note] [keyboard shortcut note]
- NVDA: [label] [role] [required field note] [validation note] **[description]** [keyboard shortcut note] [value]
- Windows Narrator: [label] [role] [value] [required field note] [validation note] [keyboard shortcut note]

Please note:

- The description tends to be given by JAWS and NVDA at the end, and certainly after the label and the role.
- The description is not output by Windows Narrator.
- JAWS and NVDA can be configured so that the description is not output. This feature is used by experienced users of screen readers in order to perceive the content more efficiently. Therefore, only additional information should be included in the description.
- When reading with the virtual cursor descriptions are not generally output by the screen reader. The output of the descriptions only occurs with the TAB navigation. Therefore, only additional information should be included in the description, and the description should only be used with control elements that can receive the keyboard focus.

HTML

In HTML, most of the control elements can only be given a description using the title attribute. However, the title attribute is not accessible for the following reasons:

- When using the browser zoom, the tooltip which is displayed using the title attribute is not scaled.
- During the keyboard navigation, the tooltip is not shown (except in the Edge browser) and is therefore imperceptible to sighted keyboard users.
- It is not possible to move over tooltips with the mouse.
- In certain browsers (such as Firefox), the tooltips cannot be hidden without moving the focus away.
- The tooltips cannot be displayed on mobile devices or can only be displayed with difficulty. For these reasons, the title attribute should be avoided.

In HTML, in addition to the title attribute, a few elements can be given a description using another method:

- Buttons marked with the <input type=button|reset|submit> element through the value attribute, unless this is already used for the Accessible Name,
- Buttons marked with the <summary> element through their text content, unless this is already used for the Accessible Name,
- Tables with the <caption> element, unless this is already used for the Accessible Name.

This is designed to ensure, however, that the visible label is at least perceptible by the assistive technology as a description if it is not output as a label (Accessible Name). As the visible label is required to correspond to or at least be contained in the Accessible Name (EN 301 549: 9.2.5.3), none of the three methods should be used for marking up a description.

Further informationen: 3.2.6.1 The title attribute - HTML Standard (whatwg.org) (External Link) , Providing Accessible Names and Descriptions | APG | WAI | W3C, 4. Accessible Name and Description Computation - HTML Accessibility API Mappings 1.0 (w3.org)

ARIA

In ARIA, it is possible to communicate descriptions with the attributes aria-describedby and aria-description.

- With aria-describedby, reference can be made to the IDs of visible or invisible descriptions.
- With aria-description, the description can be specified directly in text form. The attribute is only defined with ARIA 1.3, which means it is not yet supported by older assistive technology.
- Reference can be made to hidden elements using aria-describedby , e.g. those that have been marked with display:none or hidden . The contents of the hidden elements are used for the

description of the element with aria-describedby. In such cases, however, the use of aria-description is recommended.

Additional ARIA attributes can be used for the communication of further information to the Accessibility API that has a descriptive character, but is not a description (Accessible Description):

- Error messages with aria-errormessage,
- Placeholders with aria-placeholder,
- Information about keyboard shortcuts with aria-keyshortcuts,
- Reference to a detailed description with aria-details.

Please note: Although the aria-roledescription attribute sounds as though it could be used to describe the role of the element, this is not the case. With aria-roledescription , the role of the element is overwritten, which means that the attribute should be used with caution.

Further informations: aria-describedby property - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, aria-description property - Accessible Rich Internet Applications (WAI-ARIA) 1.3 (w3c.github.io) (External Link)

General information

- Although a description that is programmatically communicated as an Accessible Description can be more detailed than the label, it should also be concise and meaningful. The description should not be redundant to the label.
- In addition, the Accessible Description should not contain any change of language or structured content, as this is imperceptible with the assistive technology.
- Longer descriptions or descriptions with changes of language or structured content should be implemented in such a way that they can be read with the virtual cursor of the screen reader, i.e. they are available in text form. From the corresponding control element, reference can be made to the detailed description using aria-details details or in a short Accessible Description.
- For each element, it is only possible to communicate a description to the Accessibility API using one method. If several methods are used, only the description whose method has the highest priority is used as the Accessible Description. The priority of the methods is defined as follows:
    1. aria-describedby,
    2. aria-description,
    3. visible label in the case of buttons and tables, unless used for the Accessible Name (see above),
    4. title.
- If the description of a control element occurs explicitly using an attribute (for example, aria-description, aria-describedby, title), the attribute must be situated on the element that receives keyboard focus and has the role of the element to be described. The control element cannot be described by attribute if the attribute is situated on a parent or child element..
- Text content which is shown before or after as pseudo-elements using the CSS selectors are taken into account when determining the description. To avoid this, they should be marked with aria-hidden=true.
- If a reference is made to a form field using aria-describedby, the value of the form field (not its label, however) is taken into account when determining the description of the element using aria-describedby.

- The visible description should not be marked with aria-hidden=true, even if an Accessible Description is available, to ensure that the text-to-speech works using the mouse.
- The visible label should be used as an Accessible Description to avoid a repeated output of the description with the text-to-speech feature.
- To ensure a meaningful reading sequence with the virtual cursor of the screen reader, descriptions should be in the source code, at or within the element which is described, e.g.
    - Input instructions in a form as a child element at the beginning of the <form> element,
    - Input instructions of a form field directly before or after the form field, but after the label of the field under all circumstances.
- Die Attribute title, aria-describedby and aria-description attributes are global attributes, i.e., they can be used with any HTML element and/or ARIA role. It should be noted, however, that most screen readers do not output the Accessible Description when reading with the virtual cursor, which means that these three attributes should only be used for control elements that receive the keyboard focus. There is no problem in using aria-describedby with other elements, provided that the description text to which the attribute refers is also readable with the virtual cursor, i.e. it has not been hidden (e.g. with hidden or aria-hidden=true).

Text

Online betrachten

Synonyms: Continuous text

See also: font, label, description, error message, graphic

The text is used to convey information in writing. Among others, text can contain headings, links, lists und Graphics enthalten.

Presentation

No.	Property	Description	Classification	Reference
265	Contrast	Text must have a contrast ratio of at least 4.5:1 with respect to the background. Note: With large font (from 24 px and/or from 18.7 px in the case of bold font), a contrast ratio of at least 3:1 is sufficient.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
266	Contrast	Text should have a contrast ratio of at least 7:1 with respect to the background. Note: With large font (from 24 px and/or from 18.7 px in the case of bold font), a contrast ratio of at least 4.5:1 is sufficient.	Should	WCAG 2.1: 1.4.6 (AAA)
267	Color coding	If information in the text is conveyed using color, this information must also be conveyed in another way, e.g. in text form.	Must	EN 301 549: 9.1.4.1, 11.1.4.1

No.	Property	Description	Classification	Reference
268	Resizing	It must be possible to scale the text by up to 200%. During the scaling, the text must remain completely visible and may not obscure other areas of the page or be obscured by them. Note 1: It is permissible for areas of text to have to be scrolled through vertically after the scaling so that they are displayed in full. Note 2: The application can offer its own zoom function. Alternatively, the settings of the platform software can be applied.	Must	EN 301 549: 9.1.4.4, 11.1.4.4
269	Resizing	The text must be displayed in full without horizontal scrolling at a screen width of 320 px.	Must	EN 301 549: 9.1.4.10, 11.1.4.10
270	graphic	The text may not contain any Schriftgrafiken unless they are adaptable to the user requirements (font style, font size, font color, background color).	Must	EN 313 549: 9.1.4.5, 11.1.4.5.1
271	Graphic	The text should not contain any images of text.	Should	WCAG 2.1: 1.4.9 (AAA)

No.	Property	Description	Classification	Reference
272	User preferences	If the text spacing can be adjusted within the application, the operation and perceptibility of the application is maintained. Note: This applies to the following spacing: • line spacing up to 1.5 times the font size, • paragraph spacing up to 2 times the font size, • character spacing up to 0.12 times the font size, • word spacing up to 0.16 times the font size.	Must	EN 301 549: 9.1.4.12. 11.1.4.12
273	User preferences	The application must accept the font style, size, and color settings from the platform software or provide a mode in which the settings are applied. Note 1: If the settings of the platform software are not applied automatically, the corresponding mode must be explained in the information on accessibility. Note 2: The application can also offer a mode in which users can choose their preferences for the font style, size, color and any other possible font attributes directly in the application.	Must	EN 301 549: 11.7, 12.1.1
274	Comprehensibility	Unusual or unintelligible words should be avoided or explained.	Should	WCAG 2.1: 3.1.3 (AAA)

No.	Property	Description	Classification	Reference
275	Comprehensibility	Words whose meaning depends on their pronunciation should be avoided if the meaning is not apparent from the context. Alternatively, their meaning or pronunciation should be explained.	Should	WCAG 2.1: 3.1.6 (AAA)
276	Comprehensibility	Text content should be formulated simply enough to be understood by users who have a lower secondary education level. If this is not possible, an additional comprehensible alternative should be offered (language version, illustration, abstract, text in simple language).	Should	WCAG 2.1: 3.1.5 (AAA)
277	Comprehensibility	Abbreviations in the text should be avoided. Alternatively, a mechanism should be available for displaying the unabbreviated form and/or the meaning of the abbreviation. Note: This does not apply to commonly known abbreviations such as “USA” or “e.g.”.	Should	WCAG 2.1: 3.1.4 (AAA)
278	Comprehensibility	The text should be formulated in the application language.	Should	WCAG 2.1: 3.1.3 (AAA), 3.1.5 (AAA)
279	Readability	Text blocks should not be larger than 80 characters or it is possible to adjust the width.	Should	WCAG 2.1: 1.4.8 (AAA)
279	Readability	Text blocks should not be justified or it is possible to adjust the orientation.	Should	WCAG 2.1: 1.4.8 (AAA)
280	Readability	The spacing between lines of text should be at least 1.5 times larger than the font size, and the spacing between paragraphs of text should be at least 1.5 times larger than the spacing between lines, or it is possible to adjust the text spacing.	Should	WCAG 2.1: 1.4.8 (AAA)

No.	Property	Description	Classification	Reference
281	Animation	The text may not sparkle, flash or be visually changed in any other way (see Animation).	Must	EN 301 549: 9.2.3.1, 11.2.3.1, 9.2.2.2, 11.2.2.2
282	Animation	The text should be displayed all the time and should not be animated during operation.	Should	WCAG 2.1: 2.3.3 (AAA)
283	Fokussichtbarkeit	If the text receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
284	Use of the keyboard	In applications that do not support the virtual cursor , the text must receive the keyboard focus (see Practical tip: text). Note 1: This does not apply if the text is associated with a keyboard-focusable element and therefore serves as a label or desription of the element. Note 2: If the technology to be used is not to allow text to be focused, other elements must be used for the presentation of text that can receive the focus, such as read-only input fields.	Must	EN 301 549: 11.1.3.1
285	Use of the keyboard	If the text contains control elements, these must be operable with the keyboard.	Must	EN 301 549: 9.2.1.1, 11.2.1.1

Use of the keyboard: text (in an application that does not support the virtual cursor)

Action	Key	Classification
Focusing text	TAB	Required
Exiting text	TAB	Required
Navigating within text	ARROW KEYS	Reommended

Programming/interfaces

No.	Property	Description	Classification	Reference
286	Role	A role for “text” must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 11.5.2.6
287	Text	The text content must be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 11.5.2.10
288	Text	If the text contains structured contents (such as headings , lists or tables), these must be communicated to the Accessibility API in structured form. Note: The specific requirements concerning headings, lists and tables can be found in the respective section.	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 11.5.2.6
289	Text	If information is conveyed on the basis of the visual design of text, such as the color, font style or font size, this information must be communicated to the Accessibility API programmatically or in text form.	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 11.5.2.10
290	Web: Change of language	If the text contains foreign-language terms, the change of language must be marked.	Must	EN 301 549: 9.3.1.2

No.	Property	Description	Classification	Reference
291	Desktop: Comprehensibility	The text should contain only words in the application language. Note: In the applications in which marking up the change of language is possible, the language of foreign-language sections of text should be marked accordingly.	Must	EN 301 549: 11.1.1.1
292	Graphic	Content-bearing graphics in the text must have an equivalent alternative text.	Must	EN 301 549: 9.1.1.1, 11.1.1.1
293	Graphic	If a text contains a decorative graphic, the graphic must be marked as a layout graphic so that it is ignored by the assistive technology.	Must	EN 301 549: 9.1.1.1, 11.1.1.1
294	Desktop: size and position	The size and position of the text must be communicated to the Accessibility API (see Fokusindikator).	Must	EN 301 549: 11.5.2.5, 11.5.2.10, 11.5.2.13

Practical tip: text

Application that does not support the virtual cursor

Longer texts (guideline: 80 to 400 characters) without a text structure (i.e., continuous text only, without lists, tables, headings, etc.) in applications that do not support the virtual cursor, should

- receive the focus in multiple TAB steps (approximately 80 characters per TAB step), or
- be marked so that they are readable with the virtual cursor, or
- be offered in a linked document which can be read with the virtual cursor (e.g. HTML, PDF, RTF).

Long texts (from approx. 400 characters) or texts with structure (e.g. lists, tables, headings) should

- be marked so that they are readable with the virtual cursor, or
- be offered in a linked document which can be read with the virtual cursor (e.g. HTML, PDF, RTF).

Applications with several TAB steps on text and other non-operable elements should have a mode to disable such unnecessary navigation steps for sighted keyboard users. This mode and its enabling should be described in the accessibility instructions.

Application that supports the virtual cursor

If copying individual text content in the application context may be relevant, this should also be possible with the keyboard. The following requirements must then be met for the text content:

- the text content is focusable with the keyboard, # the Textcursor is visible within the text,
- within the text, it is possible to navigate with the arrow keys to determine the starting point of the highlighting,
- the text can be highlighted with SHIFT + arrow keys,
- the highlighted text can be copied using CTRL+C (also via the context menu if necessary), and is saved to the Windows clipboard after copying.

To accomplish this, read-only input fields can be used.

If the entire content of a section of text must be copyable, a button for the copying of the text content can be implemented. Pressing the button should result in the text being copied to the clipboard.

Applications with several TAB steps on copyable pieces of text should have a mode for disabling such navigation steps. This mode and its enabling should be described in the accessibility instructions.

Graphical elements

Online betrachten

Table of content

- Graphic
- Layout graphic
- Progress bar

Graphic

Online betrachten

Synonyms: Graphical element, icon, image, illustration, content-bearing graphic, pixel graphic, vector graphic, image

See also: Layout graphic

Graphics are used to convey information in a visual, non-textual format.

Note: Additional requirements concerning graphics which communicate the status, value, role or labeling of an element are described at the respective element and/or in the corresponding sections (e.g. element status, label).

Presentation

No.	Property	Description	Classification	Reference
295	Contrast	All content of graphics must have a contrast ratio of at least 3:1. This applies to the contrast of the graphic with the background as well as to the contrasts within the graphic (between neighboring areas) insofar as they are relevant to the conveying of the information. Exceptions: • disabled elements, • graphics that cannot be changed without distorting them, such as logos, flags, screenshots, heat maps or medical charts, • graphics that have a visible and equivalent text alternative. Note: If graphics do not have sufficient contrast and are classified as one of the exceptions, they must not be used as labels for control elements or convey relevant information.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

No.	Property	Description	Classification	Reference
296	Color coding	If information is conveyed using different colors within a graphic or between different graphics, then all colors (each with respect to the others) must have a contrast ratio of at least 3:1 (see Practical tip: color coding). Note: This applies if the colors themselves have no meaning, but only the color difference does.	Should	EN 301 549: 9.1.4.1, 11.1.4.1
297	Color coding	If information is conveyed using a particular color within a graphic, this information must also be conveyed in another way (see practical tip: color coding). Note: This applies when the color itself has a meaning, such as “green” for correct and “red” for incorrect.	Should	EN 301 549: 9.1.4.1, 11.1.4.1
298	Contrast	Graphics should be clearly visible when using Windows Contrast Adjustment (see Practical tip: contrast adjustment).	Should	EN 301 549: 11.7
299	Contrast	Graphics should not be used as background graphics for text, as this affects their readability. Note: This can lead to insufficient contrast, especially when users adjust the text color or font size according to their requirements.	Should	EN 301 549: 11.7

No.	Property	Description	Classification	Reference
300	Text	Images of text may not be used unless their text content can be adapted to the user requirements (font style, font size, font color, background color). Exception: Logos	Must	EN 301 549: 9.1.4.5, 11.1.4.5.1.1
301	Alternative text	Complex graphics must have a detailed description in text form. Note 1: The detailed description should either be displayed at the graphic, or it should be possible to show it or access it at the graphic using a control element. Note 2: The complex graphic itself must have a concise alternative text. It is recommended that this makes reference to the detailed description.	Must	EN 301 549: 9.1.1.1, 11.1.1.1
302	Animation	The graphic may not sparkle, flash or be visually changed in any other way (see Animation).	Must	EN 301 549: 11.2.3.1, 9.2.2.2, 11.2.2.2
303	Focus visibility	If the graphic receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
304	Use of the keyboard	In applications that do not support the virtual cursor , it must be possible to access and exit the graphic with the keyboard (see Use of the keyboard table, below). Exception: If the graphic serves as a label for a control element or conveys its role, status or value, the control element should receive the keyboard focus and not the graphic. Note: If the application contains several graphics, there should be an operating mode in which only the interactive elements receive the focus to avoid unnecessary navigation steps for sighted keyboard users.	Must	EN 301 549: 11.1.1.1

Use of the keyboard: graphic

Note: The following requirements only apply if the graphic has to be accessible with the keyboard (see above).

Action	Key	Classification
Focusing graphic	TAB	Required
Exiting the graphic	TAB	Required

Use of the pointing device: graphic

Action	Key	Classification
Showing the tooltip	Hover	Recommended

Programming/interfaces

No.	Property	Description	Classification	Reference
305	Role	The “graphic” role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.5
306	Name	The graphic must have a concise and meaningful alternative text which is communicated to the Accessibility API as an Accessible Name.	Must	EN 301 549: 9.1.1.1, 11.1.1.1
307	Description	Complex graphics must be accompanied by a detailed text alternative that describes all the relevant content of the graphic in full.	Must	EN 301 549: 9.1.1.1, 11.1.1.1
308	Operation	In applications that do not support the virtual cursor , it must be possible to access and exit the graphic with the assistive technology.	Must	EN 301 549: 9.1.1.1, 11.1.1.1
309	Desktop: Position	The size and position of the graphic must be communicated to the Accessibility API (see Focus visibility).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: graphics in Web applications that communicate a role, status, or value

HTML

The HTML standard elements do not have to be considered with regard to the graphics that communicate information about role, status or value, because the corresponding information is automatically communicated correctly by the browser to the Accessibility API.

ARIA

In the case of custom elements that are implemented with ARIA roles and ARIA attributes, the graphics that convey information about the role, status, or value should be marked as layout graphics. Instead, the information should be programmatically communicated, e.g. with the following attributes:

- role for the role,
- aria-valuenow and aria-valuetext for the value,
- aria-required for required fields,
- aria-invalid for incorrect form fields,
- aria-checked and/or aria-selected for the “selected” status,

- aria-disabled for disabled control elements,
- aria-pressed for the “pressed” and/or “not pressed” status,
- aria-expanded for the “reduced”/“hidden” and/or “expanded”/“shown” status
- aria-haspopup for elements that show a menu, selection list, tree structure, table, or dialog when they are enabled,
- aria-sort for the sorting direction,
- aria-current for marking the current element.

Insofar as no ARIA attribute exists for the relevant information, the information should be communicated in text form as part of the label or description of the element.

Further information: Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)

Practical tip: graphics in Web applications

Screen reader output

- JAWS: [alternative text] **graphic**
- NVDA: **Graphic** [alternative text]
- Windows Narrator: [alternative text] **image**

HTML and ARIA

The way in which graphics are marked and labeled depends on the method which is used for the presentation of the graphic:

- Graphics marked with the <img> element are labeled with the alt attribute.
- Buttons with graphic labeling which are marked with the <input type=image> element are also labeled with the alt attribute.
- All other graphics should be marked with role=img and explicitly labeled with aria-label or aria-labelledby. This also applies, for example, to the elements <svg> and <canvas>, CSS background graphics, font icons and letters that are used in a graphical context (e.g. “x” for “deleted”).
- Graphics for list characters (list-style-image) cannot be provided with an alternative text or marked as layout graphics and should not therefore be used.

In this case, the following exceptions apply:

- Graphics that are used for the labeling of control elements can be marked as layout graphics. Instead, the control element is expressively labeled (e.g. <button title=delete><img src=... alt></button>, <button aria-label=delete><img src=...></button>). If the control element is labeled using aria-label or aria-labelledby, the contained graphic automatically becomes the layout graphic, and does not have to be marked separately as such.
- The screen readers provide the relevant information concerning the elements (label, role, status, value) which are located within <svg> and <canvas> if the respective <svg> and/or <canvas> element is not marked as a graphic. This means, for example, that the <svg> and/or <canvas> element can be labeled as a group with role=group and with aria-label or aria-labelledby. In this case, it is not directly perceptible with the screen reader that it is a graphic (this information can be communicated indirectly through the labeling of the group, however), but it is possible to communicate detailed and structured text alternatives within the graphic elements, in the case of diagrams, for example.

- Font icons or graphics that are shown before or after as pseudo elements using the CSS selectors may also be provided with an alternative text directly in the CSS in the future (1.2. Alternative Text for Accessibility - CSS Generated Content Module Level 3 (w3.org) (External Link)). At present, this is not yet reliably supported by the assistive technology, however.

Further information: 4.8.3 The img element - HTML Standard (whatwg.org) (External Link) , Images Tutorial | Web Accessibility Initiative (WAI) | W3C

Layout graphic

Online betrachten

Synonyms: Decoration graphic, decorative graphic, non-content-bearing graphic, possibly also background graphic

See also: Graphic

Layout graphics serve the purpose of the visual design of the application, but without conveying information. Layout graphics can be purely decorative, for example, or displayed in text form in parallel with information, but without communicating information in addition to the text.

Presentation

No.	Property	Description	Classification	Reference
310	Contrast	Layout graphics should not be used as background graphics for text, as this affects their readability. Note: This can lead to insufficient contrast, especially when users adjust the text color or font size according to their requirements.	Should	EN 301 549: 9.1.4.3, 11.1.4.3, 11.7
311	Animation	The layout graphic may not sparkle, flash or be visually changed in any other way (see Animation).	Must	EN 301 549: 9.2.3.1, 11.2.3.1, 9.2.2.2, 11.2.2.2

Operation

No.	Property	Description	Classification	Reference
312	Role	Layout graphics must not receive the focus.	Must	EN 301 549: 9.1.1.1, 11.1.1.1

Programming/interfaces

No.	Property	Description	Classification	Reference
313	Role	The layout graphic role must be communicated to the Accessibility API. Alternatively, the layout graphic should not be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.5

Practical tip: layout graphics in Web applications

Screen reader output

None

HTML and ARIA

The way in which layout graphics are marked depends on the method which is used for the presentation of the graphic:

- Graphics that are marked with the <img> element should be marked with a blank alternative text as a layout graphic (<img src=... alt="">).
- Graphics that are marked with the <svg> element should be marked with aria-hidden as a layout graphic (<svg aria-hidden=true>).
- Graphics that are marked with the <canvas> element should be marked with aria-hidden as a layout graphic (<canvas aria-hidden=true>).
- Font characters that are purely decorative should be marked as a layout graphic with aria-hidden (~~~~).
- Font icons or graphics that are shown before or after as pseudo-elements using the CSS selectors should be marked as layout graphics with aria-hidden (). In the future, it will be possible for these pseudo-elements to be directly marked as layout graphics in the CSS (1.2. Alternative Text for Accessibility - CSS Generated Content Module Level 3 (w3.org) (External Link)). At present, this is not yet reliably supported by the assistive technology, however.
- Font icons should be marked as a layout graphic with aria-hidden (i).
- Graphics for list characters (list-style-image) cannot be provided with an alternative text or marked as layout graphics and should not therefore be used.
- CSS background graphics (background-image) are automatically layout graphics.
- Other CSS graphics (e.g. those created with border) are automatically layout graphics.

Layout graphics should not receive the keyboard focus, i.e. the graphic itself or its descendant elements should not be marked with tabindex or as control elements (e.g. <button>).

Further information: Decorative Images | Web Accessibility Initiative (WAI) | W3C

Progress bar

Online betrachten

Synonyms: Progress display

See also: Slider, graphic

A progress bar shows how far a process has progressed (see DIN EN ISO 9241-161: 8.30). The progress can be displayed in text form, graphically (e.g. progress bar) or as a combination of graphics and text. The presentation of the progress bar changes automatically until the process is complete.



Figure 16: Progress bar

Presentation

No.	Property	Description	Classification	Reference
314	Contrast	The progress bar must have a contrast ratio of at least 3:1. This applies to the contrast of the progress bar with the background as well as to the contrasts within the progress bar (between the filled and unfilled bar).	Must	EN 301 549: 9.1.4.11, 11.1.4.11
315	Contrast	Text in and adjacent to the progress bar must have a contrast ratio of at least 4.5:1.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
316	Focus visibility	If the progress bar receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
317	Use of the keyboard	In applications that do not support the virtual cursor , it must be possible to access and exit the progress bar with the keyboard (see Use of the keyboard table). Exception: The progress bar is marked so that its updates can are perceptible without focusing with the use of assistive technology.	Must	EN 301 549: 9.1.1.1, 11.1.1.1
318	Use of the keyboard	In applications that support the virtual cursor, the progress bar should not receive the focus.	Should	EN 301 549: 9.4.1.4, 11.2.4.3

Use of the keyboard: progress bar

Note: The following table only applies if the progress bar has to be accessible with the keyboard (see above).

Action	Key	Classification
Focus on the progress bar	TAB	Required
Exit the progress bar	TAB	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
319	Role	The “progress bar” role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.5

No.	Property	Description	Classification	Reference
320	Name	The progress bar must have a concise and expressive Accessible Name. Note 1: The Accessible Name does not have to be visible. Note 2: Text that identifies the current process step is not the Accessible Name, but the value of the progress bar.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
321	Value	The value of the progress bar must be communicated to the Accessibility API (see Accessibility API). Note: The value of the progress bar is usually expressed as a percentage. The current process step can also be specified in text form (e.g. the name of the file that is currently being copied).	Must	EN 301 549: 11.4.1.2, 11.5.2.7
322	Desktop: Value range	The minimum and maximum value of the progress bar must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.7
323	Operation	In applications that do not support the virtual cursor , it must be possible to access and exit the progress bar with assistive technology (see Accessibility API). Exception: The progress bar is marked so that its updates can are perceptible without focusing with the use of assistive technology.	Must	EN 301 549: 9.1.1.1, 11.1.1.1

No.	Property	Description	Classification	Reference
324	Desktop: Position	The size and position of the progress bar must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: progress bar in Web applications

Screen reader output

Progress bar with value:

- JAWS: [label] [value] **progress bar** percent
- NVDA: [label] **progress bar** [value]
- Windows Narrator: [label] [value in %] percent **status bar** current value [value] minimum value [minimum value] maximum value [maximum value]

Progress bar without value:

- JAWS: [label] 0 **progress bar** indeterminate
- NVDA: [label] **busy indicator**
- Windows Narrator: [label] 0 percent **progress bar**

Please note:

- JAWS misleadingly outputs the value with the addition “percent”, without converting it into a percentage value.
- The updating of the progress bar is not automatically perceptible with JAWS and Windows Narrator.
- NVDA automatically outputs the progress bar update regardless of the focus position with short beeps whose pitch represents the level of the value.

HTML

The progress bar should be implemented with the HTML element <progress>.

The current value is set with the value attribute. If no value attribute is specified, the progress bar is indeterminate and only indicates that progress is being made without being able to show its extent.

The maximum value is set with the max attribute. It should be noted that its value is imperceptible with many forms of assistive technology. The minimum value is always 0.

The label should be linked to the progress bar with the <label for=ID> element.

The <progress> element can contain different child elements depending on the HTML specification. These are neither visually perceptible nor output by the assistive technology, however.

Further information: 4.10.13 The progress element - HTML Standard (whatwg.org)

ARIA

If the progress bar is not implemented with the HTML element, it is also necessary to take account of the following:

- The role is communicated with role=progressbar.
- The current value can be specified with aria-valuenow. If the value is not specified, the progress bar is indeterminate.
- aria-valuetext can also be used to specify a value in text form which should be then output by the assistive technology instead of the value in aria-valuenow.
- The minimum and maximum values can be specified with aria-valuemin and aria-valuemax.
- The labeling can take place with aria-label or aria-labelledby.
- The presentation of the progress bar should be verified in Windows High Contrast mode.

Further information: progressbar role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)

Structural elements

Online betrachten

Table of contents

- Desktop: Window
- Tooltip
- Form
- Toolbar
- Group
- List
- Table
- Hierarchical table
- Title bar
- Status bar
- Modal dialog

Desktop: Window

Online betrachten

Synonyms: Application window

See also: Title bar, status bar, modal dialog

A window contains all the currently visible elements of the application (see DIN EN ISO 9241-161: 8.51). A window can contain the following elements:

- Title
- Working area (with menu if necessary),
- Status bar.

Note: All the requirements concerning the window refer to desktop applications only. For Web applications, the browser is the window. The Web application itself does not contain any windows.

Examples:



Figure 17: Window with title bar, working area and status bar

Presentation

Only those requirements that are directly related to the window are described below. Requirements of elements within the window are described at the respective element.

No.	Property	Description	Classification	Reference
323	Resizing	All elements of the window must be perceptible and operable with a font size adjustment of up to 400% (and a resulting display width of 320 px), • in the sense that they wrap (i. e. they are shown in several successive rows), or • in the sense that the elements that cannot be displayed in a row can be called up using a menu button, or • in the sense that they can be scrolled.	Must	EN 301 549: 9.1.4.10, 11.1.4.10

Operation

No.	Property	Description	Classification	Reference
324	Use of the keyboard	It must be possible to access, operate and exit the window with the keyboard (see Use of the keyboard table, below). Note: This applies to both the control elements in the window and to the window itself (e.g. scaling and moving the window).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2

Use of the keyboard: window

Action	Key	Classification
Focusing of the window (first or last focused element)	ALT+TAB	Required
Exiting the window	ALT+TAB	Required
Navigating within the window	TAB	Required
Opening the system menu (with the window close, move, and scale functions)	ALT+SPACE	Required
Closing the application window	ALT+F4	Required
Resizing of the window (Minimized # Normal size # Full screen (if available))	WIN+UP ARROW	Required
Minimizing the window (Full screen (if available) # Normal size # Minimized)	WIN+DOWN ARROW	Required
Quick navigation between the page areas	F6	Recommended

Use of the pointing device: window

Action	Key	Classification
Focusing of the window	Clicking in the window	Required
Exiting the window	Click outside the window	Required
Scaling of the window (if possible)	Drag and drop on window border	Required
Enabling the buttons in the title	Left click	Required
Moving the application window	Drag and drop on the title bar Note: If the application is in full screen mode, it automatically switches to normal size	Required
Switching between normal size and full screen (if available)	Double click on the title bar	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
325	The window must have a concise and expressive Accessible Name.	Must	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5

Tooltip

Online betrachten

Synonyms: Quick info, info tip, mouse over

See also: Description, tooltip, modal dialog

The purpose of a tooltip is to display an additional piece of information dynamically, e.g. the label, or context-specific Help. (see DIN EN ISO 9241-161: 8.50). Tooltips are displayed during the focusing of the corresponding UI element. Tooltips do not contain any interactive elements.

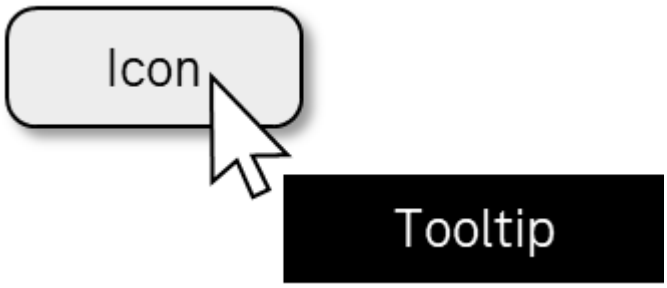

Figure 18: Tooltip

Presentation

No.	Property	Description	Classification	Reference
326	Visibility	The tooltip should be displayed at the respective element.	Should	DIN EN ISO 9241-161: 8.50
327	Contrast	The text in the tooltip must have a contrast ratio of at least 4.5:1 with respect to the background. With large font (from 24 px and /or from 18.7 px in the case of bold font), a contrast ratio of at least 3:1 is sufficient.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
328	Resizing	All content of the tooltip must be perceptible and operable with a font size adjustment of up to 400% (and a resulting display width of 320 px), in the sense that they wrap and can be vertically scrolled if necessary.	Must	EN 301 549: 9.1.4.10, 11.1.4.10

Operation

No.	Property	Description	Classification	Reference
329	Use of the keyboard	It must be possible to open and close the tooltip with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1
330	Use of the keyboard	The tooltip may not contain control elements as these cannot be keyboard-operable unless an alternative form of operation is documented in the Help option and the application (by keyboard shortcut, for instance).	Must	EN 301 549: 9.2.1.1, 11.2.1.1
331	Use of the keyboard	If the tooltip is shown while navigating with the keyboard, it must be possible to close the tooltip again with the keyboard without moving the keyboard focus away (e.g. with ESC), unless • the content consists of an error message, or • the content only hides white space or decorative non-text content . Note: This does not apply to tooltips that are shown by the platform software by default.	Must	EN 301 549: 9.1.4.13, 11.1.4.13
332	Use of the keyboard	If the tooltip is shown when navigating with the keyboard, the tooltip must be displayed until the keyboard focus is moved away from the triggering element and/or the tooltip, unless	Must	EN 301 549: 9.1.4.13, 11.1.4.13

No.	Property	Description	Classification	Reference
		• the tooltip was deliberately closed (e.g. with ESC), or • the content of the tooltip is no longer valid (e.g. an error message in the input field after entering a correct value). Note: This does not apply to tooltips that are shown by the platform software by default.		
333	Use of the pointing device	If the tooltip is shown when hovering with a pointing device, it must be possible to hide the tooltip again without moving the pointing device away, unless: • the content consists of an error message, or • the content only hides white space or decorative non-text content. Note 1: This does not apply to tooltips that are shown by the platform software by default. Note 2: The automatically-shown content can be hidden with ESC or clicking on the triggering element, for instance, as long as no other actions are triggered.	Must	EN 301 549: 11.1.4.13

No.	Property	Description	Classification	Reference
334	Use of the pointing device	If the tooltip is shown when hovering with a pointing device, the tooltip must be displayed until the pointing device is moved away from the triggering element and/or the tooltip, unless • the tooltip was deliberately closed (e.g. with ESC), or • the content of the tooltip is no longer valid (e.g. an error message in the input field after entering a correct value). Note: This does not apply to tooltips that are shown by the platform software by default.	Must	EN 301 549: 9.1.4.13, 11.1.4.13
335	Use of the pointing device	If a tooltip is shown when hovering with a pointing device, it must then be possible to move over the tooltip with the pointing device, i.e. the tooltip may not be hidden as soon as the pointing device is no longer positioned over the triggering element. Note: This does not apply to tooltips that are shown by the platform software by default.	Must	EN 301 549: 11.1.4.13

Use of the keyboard: tooltip

Action	Key	Classification
Opening the tooltip	Navigating to the element	Required
Closing the tooltip	ESC	Required

Use of the pointing device: tooltip

Action	Key	Classification
Opening the tooltip	Hovering	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
336	Name	If the tooltip contains a description, this must be communicated to the Accessibility API as the Accessible Description (see Description).	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 11.5.2.5
337	Name	If the tooltip contains a label of a graphical element, this should correspond to the Accessible Name or be contained in it.	Should	EN 301 549: 9.2.5.3, 11.2.5.3
338	Name	The tooltip should not contain any structured or long text content. Note: For structured or long text content, a display format should be chosen in which it is possible for the text to be read with the virtual cursor of the screen reader.	Should	EN 301 549: 9.1.3.1, 11.1.3.1
339	Keyboard shortcut	If the tooltip contains a keyboard shortcut for the respective control element, this must be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1

Form

Online betrachten

Synonyms: Form area

See also: Group

Forms have the purpose of the input of data. A form contains one or more form elements.

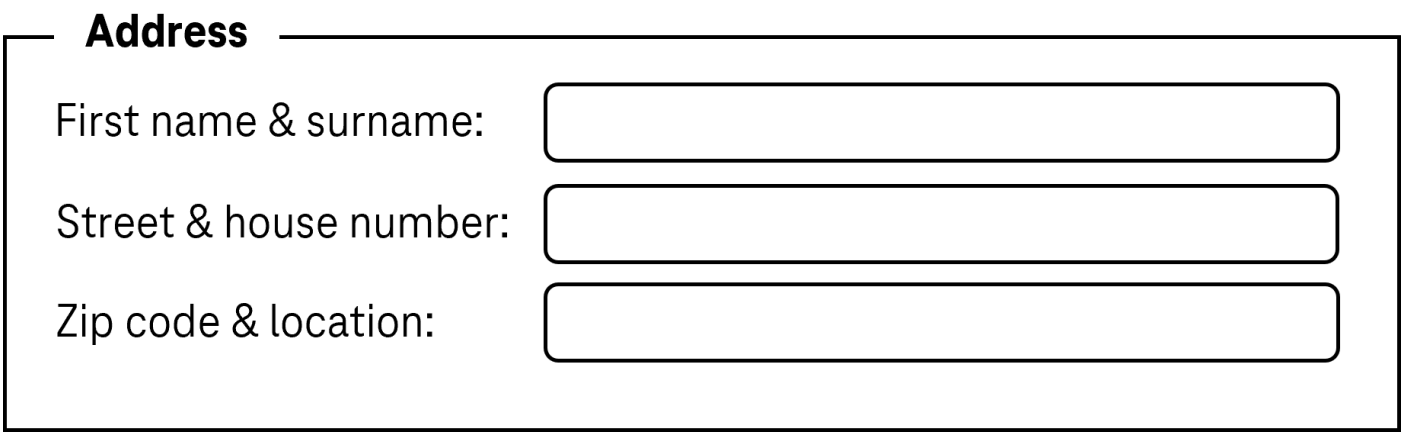

Figure 19: Address input form

Presentation

Only those requirements that are directly related to the form are described below. Requirements regarding interactive elements within the form are described at the respective element.

No.	Property	Description	Classification	Reference
340	Resizing	All elements of the form must be perceptible and operable with a font size adjustment of up to 400% (and a resulting display width of 320 px), in the sense that they wrap and do not have to be horizontally scrolled.	Must	EN 301 549: 9.1.4.10, 11.1.4.10
341	Error prevention	The form must be designed in such a way that errors can be prevented and corrected (also see Error prevention and correction und Required field label).	Must	EN 301 549: 9.3.3.1 bis 9.3.3.4, 11.3.3.1 bis 11.3.3.4
342	Complexity	The form should be clearly designed. The content of complex forms should be grouped programmatically and visually or split into different screens.	Should	DIN EN ISO 9241-125: 5.1.8

Operation

No.	Property	Description	Classification	Reference
343	Use of the keyboard	It must be possible to access, operate and exit the interactive elements in the form with the keyboard (see Use of the keyboard table).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
344	Use of the keyboard	Frequently used buttons on the form (e. g. the Submit button) should be accessible by keyboard shortcut The keyboard shortcuts should be documented in the Help option and application.	Should	DIN EN ISO 9241-171: 9.3.10
345	Navigation sequence	The navigation sequence in the form must be designed so that the contents can be perceived in a meaningful sequence and the control elements can be accessed according to their task-appropriate processing sequence. Note: This applies, for example, to the Submit button which must receive the focus at the end of the form.	Must	EN 301 549: 9.2.4.3, 11.2.4.3
346	Updates	When focusing and operating the interactive elements within the form, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1, 9.3.2.2, 11.3.2.2
347	Click area	The click area of the interactive elements in the form should total at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: form

Action	Key	Classification
Focusing of the form (first element)	TAB	Required
Desktop: Quick navigation between form areas	F6	Recommended
Submitting the form	ENTER	Recommended

Programming/interfaces

No.	Property	Description	Classification	Reference
348	Name	If the form has a visual label, it must be communicated as an Accessible Name.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
349	Desktop: Element hierarchy	The parent / child relationships of the elements within the form must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9

Toolbar

Online betrachten

Synonyms: Symbols bar, toolbox, command bar tab

See also: Menu, group, menu button

Toolbars serve the grouping of interactive elements for the editing of content or data (see DIN EN ISO 9241-161: 8.49).

A toolbar contains interactive elements (in most cases, buttons or toggle switches) that are grouped visually with a border, for example. The contents of the toolbar are usually arranged horizontally above or below the area, the contents of which are edited with the elements of the toolbar. The elements of the toolbar can be arranged in several lines. For toolbars with several buttons, icons are often used to label the buttons due to space-related constraints.

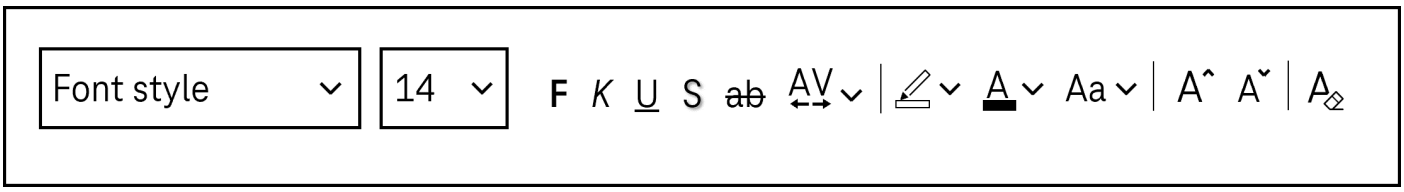

Figure 20: Toolbar for setting the font style properties

Presentation

Only those requirements that are directly related to the toolbar are described below. Requirements regarding interactive elements within the toolbar are described at the respective element.

No.	Property	Description	Classification	Reference
350	Resizing	All elements of the toolbar must be perceptible and operable with a font size adjustment of up to 400% (and a resulting display width of 320 px), • in the sense that they wrap (i. e. they are shown in several successive rows), or • in the sense that the elements that cannot be displayed in a row can be called up using a menu button, or • in the sense that they can be scrolled.	Must	EN 301 549: 9.1.4.10, 11.1.4.10
351	Grouping	To make the use of the keyboard visible, the toolbar should be designed in such a way that its elements can be identified as belonging together. Note: This can take place, for example, with a border or position and arrangement.	Should	DIN EN ISO 9241-125: 5.1.8

Operation

Nr.	Eigenschaft	Beschreibung	Klassifizierung	Referenz
352	Use of the keyboard	It must be possible to access, operate and exit the interactive elements in the toolbar with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2

Nr.	Eigenschaft	Beschreibung	Klassifizierung	Referenz
353	Use of the keyboard	The toolbar may not contain any control elements that are operated by the keys designed for navigation through the toolbar. Note 1: This can affect input fields and drop-down lists, for example, as they are operated with the arrow keys. Note 2: Alternatively, keyboard shortcuts must be implemented and documented with which it is possible to exit the control elements.	Must	EN 301 549: 11.2.1.1
354	Use of the keyboard	If the toolbar is only accessible by keyboard shortcut , keyboard shortcut must be documented in the application and the Help option.	Must	EN 301 549: 9.2.1.1, 11.2.1.1
355	Use of the keyboard	The toolbar should be accessible by keyboard shortcut. Frequently used interactive elements within the toolbar should also be given a keyboard shortcut. The keyboard shortcuts should be documented in the Help option and application.	Should	DIN EN ISO 9241-171: 9.3.10
356	Updates	When focusing and operating the interactive elements within the toolbar, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1
357	Updates	Changing the value of the form elements within the toolbar may not cause unexpected changes of context.	Must	EN 301 549: 9.3.2.2, 11.3.2.2

Nr.	Eigenschaft	Beschreibung	Klassifizierung	Referenz
358	Updates	No loss of focus may occur when the control elements within the toolbar are enabled. Note: Therefore, after operating a button, the focus must remain on the button or be placed on the element which is controlled by the button (e.g. input field of a Rich Text Editor or modal dialog which is opened).	Must	EN 301 549: 9.2.4.3, 11.2.4.3
359	Click area	The click area of the interactive elements in the toolbar should total at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: toolbar

Action	Key	Classification
Focusing of the toolbar (first or last focused element)	TAB Note: Alternatively, toolbars can be accessed by the area navigation or by keyboard shortcut.	Required
Exiting the toolbar	TAB Note: Alternatively, toolbars can be exited by the area navigation or an expected change of context after the operation of an element in the toolbar.	Required
Navigating within the toolbar	• Horizontal toolbar: RIGHT/LEFT ARROW, • Vertical toolbar: RIGHT/LEFT/UP /DOWN ARROW,	Required
Quick navigation to the first and/or last element within the toolbar	POS1, END	Recommended

Programming/interfaces

No.	Property	Description	Classification	Reference
360	Role	The toolbar role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
361	Desktop: Element hierarchy	The parent / child relationships of the elements within the toolbar must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9
362	Orientation	The orientation of the toolbar (vertical or horizontal) must be communicated to the Accessibility API.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
363	Name	If the toolbar has a label or description, these must be communicated as the Accessible Name and/or Accessible Description (see Label and Description). Note: If the page contains several toolbars, they must have a concise and expressive Accessible Name.	Must	EN 301 549: 11.2.4.6, 11.4.1.2, 11.5.2.5, 11.5.2.8
364	Keyboard shortcut	If the toolbar or control elements within the toolbar have visible keyboard shortcuts, these must be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
365	Operation	It must be possible to access, operate and exit the elements of the toolbar with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17

Practical tip: toolbar in Web applications

Screen reader output

With TAB navigation:

- JAWS: [label] **toolbar bar**
- NVDA: [label] **tool bar**
- Windows Narrator: [label] **tool bar**

With the virtual cursor:

- JAWS: **Toolbar** with [amount] controls ... **toolbar end**
- NVDA: **Toolbar** ... **out off the toolbar**
- Windows Narrator: **tool bar** ... -

HTML

There is no element for toolbars in HTML. Instead of this, buttons and other control elements can be grouped in a list (<menu> and <li>), a labeled region (e.g. <section>) or form field group (<fieldset>, labeled with <legend>). The navigation between the elements then takes place with the TAB key and not the arrow keys. To support the efficient keyboard navigation, it should be possible to skip the region and/or form field group (see practical tip on efficient navigation).

Further information: 4.4.7 The menu element - HTML Standard (whatwg.org) (External Link)
, 4.3.3 The section element - HTML Standard (whatwg.org) (External Link)
4.10.15 The fieldset element - HTML Standard (whatwg.org)

ARIA

When implementing toolbars, the following should be taken into account:

- The role is communicated with role=toolbar.
- The toolbar should be labeled with aria-label or aria-labelledby.
- A non-standard vertical orientation of the toolbar can be specified with aria-orientation=vertical. Frequently, the orientation is not output by the assistive technology, so that with a vertically oriented toolbar, operation with all arrow keys should be possible.
- The toolbar should also be visually recognizable as such, so that sighted keyboard users are able to recognize the operation with the arrow keys.
- The toolbar should contain at least three control elements.
- The visible elements within the toolbar and the programmatically focused element should have the same position and size.

Further information: toolbar role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, Toolbar Pattern | APG | WAI | W3C

Group

Online betrachten

Synonyms: Form field group, group field, grouping, groupbox

See also: Headings, form, toolbar

Groups are used to summarise elements that belong together in terms of their content (see DIN EN ISO 9241-161: 8.15). The group has a label which acts as a group label for the elements it contains.

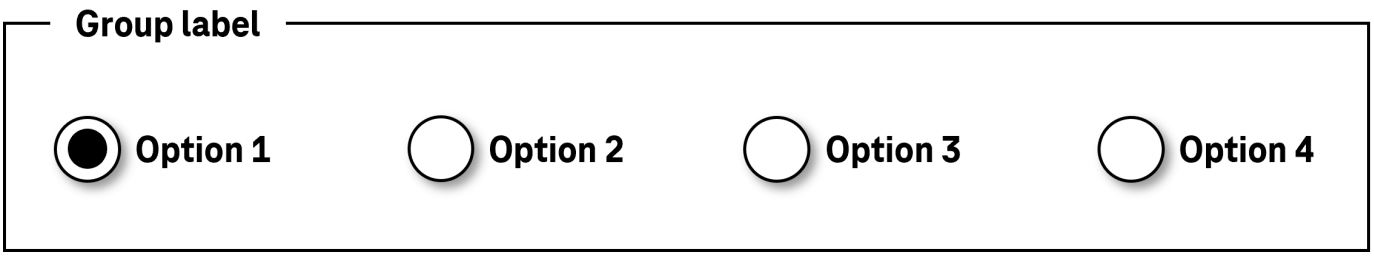

Figure 21: Grouped radio buttons with a group label

Presentation

No.	Property	Description	Classification	Reference
366	Contrast	The visual indicator for the group (e. g. the border around the group) must have a contrast ratio of at least 3:1 to the background or the neighboring colors. Exceptions: • The visual indicator has not been designed with color but spatially (e.g. using the appropriate spacing between the group and the content outside the group). • A visual indicator is not necessary because the group encompasses the entire screen, for instance.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

No.	Property	Description	Classification	Reference
367	Contrast	The label of the group must have a contrast ratio of at least 4.5:1 with respect to the background. Note: With large font (from 24 px and/or from 18.7 px in the case of bold font), a contrast ratio of at least 3:1 is sufficient.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
368	Label	The label of the group must be expressive. Note: To achieve this, the label of the group should be concise and unambiguous.	Must	EN 301 549: 9.2.4.6, 11.2.4.6
369	Label	The label of the group should be unambiguous and understandable within the context.	Should	DIN EN ISO 9241-171: 8.1.2, 8.1.3
370	Label	Form elements that belong together in terms of their content should be grouped and labeled. Note: This applies to groups of radio buttons and check boxes in particular .	Should	DIN EN ISO 9241-125: 5.1.8

Programming/interfaces

No.	Property	Description	Classification	Reference
370	Role	The “group” roles or, if applicable, a specific role for the respective group type, must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 11.5.2.5
371	Web: Structure	All visually perceptible areas of the pages shall also be programmatically perceptible, e.g. through • headings, • regions, • other groupings.	Must	EN 301 549: 9.1.3.1
372	Name	The group must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5
373	Desktop: Element hierarchy	The parent / child relationships of the elements within the group must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9

Practical tip: form field groups in Web applications

Screen reader output

- JAWS: [label] **group start** [label] ... **group end** [label]
- NVDA: [label] **grouping** ... **outside of grouping**
- Windows Narrator: [label] ...

Please note:

- Form field groups are not output by Windows Narrator.
- In JAWS and NVDA, the end of the form field group is only perceptible with the virtual cursor when reading. When navigating with TAB, the “group” and/or “grouping” role is only output with the first (and/or with SHIFT+TAB with the last) form element in the group, i.e. when accessing the group. Exiting the group is not perceptible when navigating with TAB. The element overviews of the screen readers display the label of the form field group for each form element in the group.
- For JAWS and NVDA to be able to identify the form field groups, they must be labeled, regardless of whether they are labeled with the HTML element <fieldset> or are marked with the ARIA group and /or radiogroup roles.

- If there is a heading within the <fieldset> and outside the <legend> element, the group label will not be output correctly by JAWS. Headings should therefore be avoided in <fieldset>.

HTML

Form field groups are marked with the <fieldset> element and serve the grouping of form fields that belong together, especially in the case of radio buttons. The form elements that belong to the group are nested in the source code within the <fieldsets>. The labeling of the form field group takes place with the <legend> element which should be the first child element in the <fieldset>. As the group label is output before the label of the form elements that it contains, it is necessary to take account of the following:

- The label of the group should be as concise as possible without losing any meaning.
- Form field groups should not be nested within each other.

According to the HTML specification, the <legend> element can contain control elements. Sometimes, however, this is not supported by the assistive technology; it is therefore recommended that the label of the form field group is only included in the <legend> element in text form. The <fieldset> element can be marked as disabled with disabled. This means that all the contained form elements are disabled (with the exception of form elements located in the <legend> element).

Further information: 4.10.15 The fieldset element - HTML Standard (whatwg.org), [4].10.16 The legend element - HTML Standard (whatwg.org)](https://html.spec.whatwg.org/multipage/form-elements.html#the-legend-element)

ARIA

If the form field group is not implemented with the HTML element, it is also necessary to take account of the following:

- The role of the list is communicated with role=group or role=radiogroup (with radio buttons only). When output by screen reader, no distinction is made between the group and radiogroup.
- The labeling of the group can take place with aria-label or aria-labelledby.
- The ARIA role radiogroup (in contrast to the role group) can be marked with the aria-readonly attributes as read-only, aria-required as a required field and aria-invalid as incorrect. In addition to this, aria-errormessage can be used to assign an error message to the ARIA radio buttongroup.
- Both the ARIA role radiogroup and the ARIA role group can be marked with aria-disabledas disabled. This means that all the contained form elements are output as disabled by the assistive technology (i.e. they should actually be disabled).

Further information: group role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, radiogroup role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)

List

Online betrachten

See also: Selection list, tree structure, table

A list is used for the structured display of data. A list contains several list entries. Lists can be sorted or unsorted. Lists can be nested inside each other. Lists often have a visual indicator at the beginning of each list entry, which is also known as a bullet point, e.g.

- a dash or an icon for an unsorted list
- a letter or a number for a sorted list.

List entries can contain control elements.

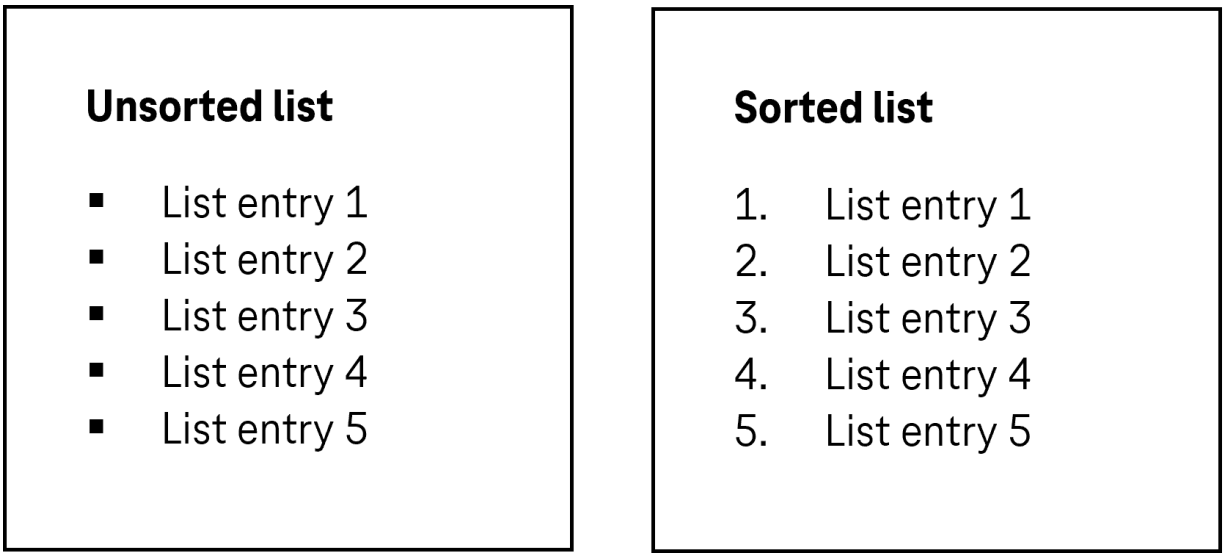

Figure 22: Unsorted and sorted list

Presentation

No.	Property	Description	Classification	Reference
374	Contrast	The text content of the list entries must have a contrast ratio of at least 4.5:1 with respect to the background. Note: In the case of sorted lists, this also applies to bullet points insofar as they convey information (e.g. if they have a number or letter).	Must	EN 301 549: 9.1.4.3, 11.1.4.11

No.	Property	Description	Classification	Reference
375	Contrast	If the list entries are only identifiable as such on the basis of their color design, the color must have a contrast ratio of at least 3:1 with respect to the neighboring colors. Note 1: A list entry might be recognizable as such on the basis of its bullet point or background color, for example. Note 2: The requirement does not apply if the list entries are clearly identifiable as such, because of the spacing between them, for example.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
376	Contrast	The graphic content of the list entries must have a contrast ratio of at least 3:1 with respect to the background. Note: In the case of sorted lists, this also applies to graphic bullet points insofar as they convey information.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
377	Resizing	To make its contents perceptible without horizontal scrolling, no list entry may be wider than 320 px with 400% zoom.	Must	EN 301 549: 9.1.4.10, 11.1.4.10
378	Hierarchy	If the list contains nested lists, the hierarchy should be clearly visible. Note: Nested lists are usually represented by indentation. Different bullet points can also be used depending on the hierarchy level.	Should	DIN EN ISO 9241-125: 6.1.2

No.	Property	Description	Classification	Reference
379	Focus visibility	If a list entry or an element in it receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
380	Use of the keyboard	It must be possible to access, operate and exit all control elements in the list with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
381	Click area	The click area of the control elements in the list should total at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: list (in an application that supports the virtual cursor)

With applications that support the virtual cursor, neither the list nor its list entries receive the keyboard focus. It should only be possible for interactive elements within the list entries to be accessed and operated with the keyboard.

Action	Key	Classification
Focusing interactive elements in the list	TAB	Required
Exiting interactive elements in the list	TAB	Required
Operating interactive elements in the list	Corresponding to the respective element	Required

Use of the keyboard: list (in an application that does not support the virtual cursor)

With applications that do not support the virtual cursor, each list entry must be able to receive the focus so that its contents are perceptible with assistive technology (e.g. screen reader).

Action	Key	Classification
Focusing of the list (first and/or last focused list entry)	TAB	Required
Exiting the list	TAB	Required
Cell-based navigation within the table in navigation mode	UP/DOWN ARROW	Required
Quick navigation (to the first and/or last list entry)	POS 1, END	Recommended
Quick navigation (with a defined increment)	PAGE UP/DOWN Note: The increment should match the amount of visible entries in the list.	Recommended

Note: In the programming languages for software, there is often no element for lists. Instead of this, selection lists are used for non-nested lists, and tree structures are used for nested lists. In this case, the selection list and/or tree structure should include a note in its label or description that it is only for the presentation of content and not for the selection.

Use of the pointing device: list

Action	Key	Classification
Operation of interactive elements	Corresponding to the respective element	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
382	Role	The (sorted and/or unsorted) list and list entry roles must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 9.4.1.2, 11.4.1.2, 11.5.2.5
383	Role	The list and list entry roles can only be used for lists. Layout lists that are solely for the visual design and not for the display of structured data must not be assigned these roles.	Must	EN 301 549: 9.1.3.1, 11.1.3.1

No.	Property	Description	Classification	Reference
384	Name	If the list has a label or description, these must be communicated as the Accessible Name and/or Accessible Description (see label and description).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
385	Operation	It must be possible to access, operate and exit the list and/or the interactive elements that it contains with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
386	Update	Updates regarding the contents of lists must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
387	Desktop: Position	The size and position of the list entries and the interactive elements they contain must be communicated to the Accessibility API (see Fokusindikator).	Must	EN 301 549: 11.5.2.5, 11.5.2.13
388	Element hierarchy	The parent / child relationships of the elements within the list must be communicated to the Accessibility API. Note 1: The assistive technology requires this information to determine, among others, the amount of list entries, their position within the list and the hierarchy of nested lists. Note 2: A list cannot be programmatically divided into different lists, for instance.	**Must**	EN 301 549: 9.1.3.1, 11.1.3.1, 11.5.2.9
389	Bullet points	The bullet point must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.1.3.1, 11.1.3.1

Practical tip: lists in Web applications

Screen reader output of sorted and unsorted lists

With the virtual cursor:

- JAWS:
  - **List** with [amount] items (nesting level [number])
  - [Bullet point] [list entry]
  - **List end** (nesting level [number])
- NVDA:
  - **List** with [amount] entries
  - [Bullet point] [list entry]
  - **Out of list**
- Windows Narrator:
  - **enter list**
  - [Position] of [amount] level [number] heading levell [number] [bullet point] [list entry]
  - **exit list**

Note: The Windows Narrator incorrectly outputs “heading level [number]“ for each list entry due to the implicit and/or explicit aria-level.

HTML

Unsorted lists are marked with <ul> and <li>. For control elements in a list, <menu> is also used instead of <ul>. The <menu> element is communicated to the Accessibility API with the semantics of the <ul> element.

Sorted lists are marked with <ol> and <li>. With sorted lists, it is possible to specify the start value with start , change the direction with reversed, and determine the type of bullet point with type.
Nested lists are implemented such that a list (<ol> or <ul>) is located within a <li> element of the parent list.

The difference between a sorted list and an unsorted list, i.e., between <ol> and <ul>, can only be identified visually and with the assistive technology on the basis of the bullet points. The bullet points should therefore match the list type.

Simple bullet points, such as disc for unsorted lists or decimal and/or lower-latin for sorted lists, should be chosen to ensure the correct output and to avoid lengthening it unnecessarily. The following sample values of the list-style-type CSS characteristic are output by the JAWS (first entry), NVDA (second entry, if different), and Windows Narrator (third entry, if different) screen readers as follows:

- none: [no output]
- disc: Bullet points
- circle: round hollow bullet point / white bullet point / empty bullet point
- square: black square / large filled black square / [no output]
- disclosure-open: black downward pointing small triangle / black downward pointing small triangle / [no output]
- disclosure-closed: filled right-pointing small triangle / black right-pointing small triangle / [no output]
- decimal: [number]

- lower-roman: [number; with numbers that only consist of one letter, just the letter; with numbers that consist of several letters, the individual letters or as a word] / [number; with numbers that consist of several letters, the individual letters] / [number; with numbers that consist of several letters, the letters as a word]
- lower-latin: [letter]
- lower-greek: [letter with the supplement “Greek lowercase”] / [no output] / [no output]
- georgian: [no output]

List characters with list-style-image are not output by JAWS and NVDA and are output as an “image” (without alternative text) by Windows Narrator.

Further information: 4.4.5 The ol element - HTML Standard (whatwg.org) (External Link)
, 4.4.6 The ul element - HTML Standard (whatwg.org) (External Link)
, 4.4.7 The menu element - HTML Standard (whatwg.org) (External Link)
, 4.4.8 The li element - HTML Standard (whatwg.org) (External Link)

ARIA

If the list is not implemented with the HTML element, it is also necessary to take account of the following:

- The role of the list is communicated with role=list.
- The role of the list entries is communicated with role=listitem.
- Nested lists can be implemented with aria-level according to the ARIA specification. This is only output correctly by Windows Narrator, however. For nested lists to be output correctly by JAWS and NVDA, nested lists should be nested correctly in the source code (for example: Entry 1 Entry 1.1 ...).

Further information: list role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)

Table

Online betrachten

See also: List, Hierarchical table

A table is used for the structured display of data (see DIN EN ISO 9241-161: 8.44).

A table consists of cells which are arranged in columns and rows. In most cases, the first row contains the column headings and the first column frequently contains the row headings. Tables can also contain various functionalities, e.g.

- sorting and filtering of the cell contents (e.g. through the column headings),
- scaling of the column and row size,
- showing or hiding of columns and/or changing the order of columns,
- showing and hiding of subordinate table rows (see: Hierarchical table),
- displaying sum total rows at the end of the table,
- in-line editing of cell contents (hereinafter referred to as editing mode),
- external editing of the cell content, e.g. using a toolbar above the table or a button within the table cells,
- Selection of table rows, e.g. by check box (multiple selection) or radio button (single selection),
- Selection of table cells, e.g. by mouse or keyboard selection,
- Browsing or scrolling through the records, if necessary, with pagination.

The requirements for the individual control elements within the table are described for the respective control element. Only the additional requirements for the whole of the element are described here.

Column heading 1	Column heading 2	Column heading 3	Column heading 4
Row heading 1			
Row heading 2			
Row heading 3			

Figure 23: Table with column and row headings

Presentation

No.	Property	Description	Classification	Reference
390	Contrast	The text content of the table cells must have a contrast ratio of at least 4.5:1 with respect to the background.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
391	Contrast	The graphic content of the table cells must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

No.	Property	Description	Classification	Reference
392	Contrast	Graphically communicated differences in status between the table cells shall have a contrast ratio of at least 3:1 compared with the background or the cells with a different status. Note: This applies to the status “selected”, “sorted” or “editable”, etc.	Should	EN 301 549: 9.1.4.1, 11.1.4.1, 9.1.4.11, 11.1.4.11
393	Contrast	If the table cells are only identifiable as such on the basis of their color design, the color must have a contrast ratio of at least 3:1 with respect to the neighboring colors. Note 1: A cell might be recognizable as such on the basis of its border or its background color, for example. Note 2: The requirement does not apply if the cells are clearly identifiable as such, because of the spacing between them, for example.	Should	EN 301 549: 9.1.4.11, 11.1.4.11
394	Label	The columns and rows must be labeled using column and row headings. Note: The whole of the table can also be labeled.	Must	EN 301 549: 11.5.2.6
395	Value	Tables should not have rows or columns that only have empty cells.	Should	EN 301 549: 11.5.2.7
396	Resizing	To make its contents perceptible without horizontal scrolling, no table cell may be wider than 320 px with 400% zoom.	Must	EN 301 549: 9.1.4.10, 11.1.4.10

No.	Property	Description	Classification	Reference
397	Focus visibility	If a table cell or an element in it receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
398	Use of the keyboard	It must be possible to access, operate and exit all control elements in the table with the keyboard (see Use of the keyboard table, below). Note 1: This also applies to functions that can be initialized from the column headings, such as the sorting of the cell contents or the adjusting of the column width through the section separators between the column headings. Note 2: Alternatively, the control elements outside the table can also be used to enable the table functionality. Note 3: If keyboard shortcuts are used to enable the use of the keyboard, they must be described in the application and the Help option. Note 4: The operation of the elements may not conflict with the navigation through the table. If the table can be navigated through with the use of the arrow keys, for example, the table may not contain any control elements that can be operated with the arrow keys unless it is possible to switch to the edit mode.	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
399	Click area	The click area of the control elements in the table should total at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: table (in an application that supports the virtual cursor)

With applications that support the virtual cursor, neither the tables nor their cells receive the keyboard focus. It should only be possible for interactive elements within the table cells to be accessed and operated with the keyboard. This only applies if the table is not marked with role=grid. Tables with role=grid are operated like tables in applications without the virtual cursor (see following section).

Action	Key	Classification
Focusing interactive elements in the table	TAB	Required
Exiting interactive elements in the table	TAB	Required
Operating interactive elements in the table	Corresponding to the respective element	Required

Use of the keyboard: table (in an application that does not support the virtual cursor)

With applications that do not support the virtual cursor and with tables that have been marked with role=grid, each table cell must be able to receive the focus so that its contents are perceptible with the assistive technology (e.g. screen reader). It is not sufficient if it is only possible to navigate through the table only one row at a time, for example. For tables in applications that do not support the virtual cursor, a distinction is made between the navigation and edit mode:

- In the navigation mode, the arrow keys can be used to navigate between the cells.
- In edit mode, the interactive elements within a cell are operable. If the cell contains multiple interactive elements, it is possible to navigate between the elements in the edit mode.

Action	Key	Classification
Focusing of the table	TAB	Required
Exiting the table	TAB	Required
Cell-based navigation within the table in navigation mode	UP/DOWN/LEFT/RIGHT ARROW	Required
Horizontal quick navigation in navigation mode (navigation to the first and/or last cell in the current row)	POS 1, END	Required with several columns
Vertical quick navigation in navigation mode (with a defined increment)	PAGE UP/DOWN Note: The increment should match the amount of visible rows.	Required with several rows

Action	Key	Classification
Quick navigation in navigation mode (navigation to the first and/or last cell in the current row)	CTRL+POS 1, CTRL+END	Required with several rows and columns
Quick navigation in navigation mode (navigation to the first and/or last cell in the table)	CTRL+POS 1, CTRL+END	Recommended
Change to the edit mode	F2, ENTER, [text input with input fields]	Required if edit mode available
Change to the navigation mode	F2, ENTER, ESC Note: With ESC, the changes made in the table cell should be discarded.	Required if edit mode available
Navigation within the cell in edit mode	TAB	Required if edit mode available
Operation of interactive elements in edit mode	Corresponding to the respective element	Required if edit mode available
Operation of an interactive element in navigation mode (if the cell only contains this element and the element is not operated using the arrow keys)	According to the respective element (e.g. ENTER for links and SPACE for buttons or check boxes) Note: Elements that are operated with the arrow keys cannot be located in tables without edit mode because the arrow keys are used for navigating through the table.	Required if no edit mode available
Operation of an interactive element in navigation mode (if the cell only contains this element and the element is not operated using the arrow keys)	According to the respective element (e.g. ENTER for links and SPACE for buttons or check boxes)	Recommended if edit mode available

Action	Key	Classification
Selecting table cells, rows, columns	• By check boxes and/or radio buttons or • SPACE Note: The selection of neighboring and non-neighboring cells, rows or columns is carried out as described at the multiple selection list element.	Required if selection is possible

Table 1: Use of the pointing device: table

Action	Key	Classification
Operation of interactive elements	Corresponding to the respective element	Required
Selecting table cells, rows, columns	• With left click on the check boxes and/or radio buttons or • Left click on a cell that does not contain a control element Note: The selection of neighboring and non-neighboring cells, rows or columns is carried out as described at the multiple selection list element.	Required if selection is possible

Programming/interfaces

No.	Property	Description	Classification	Reference
400	Role	The table, table label, table row, column header, row header and table cell roles must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
401	Role	The individual roles for tables can only be used for data tables. Layout tables that are solely for the visual design and not for the display of table data must not be assigned these roles.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
402	Role	Content which is associated with the table, but does not contain table data (such as the pagination) must not be located within the table.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
403	Name	The table must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5
404	Name	If the table has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
405	Status	The status of the table must be communicated to the Accessibility API (see Element status). Note: This applies, for example, to the “editable”, “sortable” or “selectable” status, provided that these functions are not perceptible using focusable control elements.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
406	Status	The status of the table cells must be communicated to the Accessibility API (see Element status). Note: This applies, for example, to the status “selected”, “sorted” or “disabled”.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
407	Value	The content of the table cells must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7
408	Column and row headings	If the table has column and row headings, these must be communicated to the Accessibility API for each cell.	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 11.5.2.6
409	Desktop: Element hierarchy	The parent / child relationships of the elements within the table must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9
410	Operation	It must be possible to access, operate and exit the table and/or the interactive elements that it contains with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
411	Update	Updates regarding the contents of tables and the status of the table cells must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
412	Desktop: Position	The size and position of the table cells and the interactive elements they contain must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5

Practical tip: tables in desktop applications

Tables in list form

Some programming languages for desktop applications do not allow for the creation of tables which fulfill the navigation requirements within the table (with the use of the arrow keys). Instead of this, the table can only be navigated one row at a time. Therefore, for example, the table cannot be perceived in a structured way with a screen reader (because the entire row is output, in some cases without the corresponding column headings). In addition, in most cases, the control elements in the table cannot be operated with the keyboard.
In the following exceptional cases, the use of these tables is acceptable if the technology used does not offer an alternative:

- the table only contains two columns or
- the table contains a maximum of five columns and the column headings and cell contents are short (maximum of one to two words or numbers), and

- the column headings are output when navigating through the rows, or the output is not necessary because the purpose of the columns is determined by the context (e.g. the cell content), and
- the rows and column headings do not contain control elements and/or all control elements are perceptible with assistive technology (role, purpose, status) and are operable with the keyboard and /or assistive technology.

If these requirements cannot be met, a different form of presentation must be chosen for the data.

Interactive elements in the table

Some programming languages for software do not allow for the creation of tables which fulfill the requirements referred to here regarding the operation of interactive elements within tables. In this case, interactive elements within the tables should be avoided. Instead of this, the tables should only be used to display data and the associated interactive elements should be located outside the table. Examples:

- One or more control elements can be inserted in front of the table to manage the sorting of the table contents.
- A form can be inserted above the table to filter the table contents.
- Above the table, a button can be added with which a modal dialog can be opened. In the modal dialog, the table can be fully configured using the keyboard (by adjusting the column widths, for instance).
- Above the table, there is a toolbar with which the selected table rows can be edited or deleted.

Alternatively, keyboard shortcuts or context menus can be used to enable the use of the keyboard. In this case, this alternative form of operation must be referred to in the application and the Help option.
With the control elements in the table that may be inaccessible, another option is to ensure the accessible operation with the use of control elements outside the table.

Hierarchical table

Online betrachten

Synonyms: Table with tree structure, tree grid, tree table

See also: Table, tree structure, accordion

A hierarchical table is designed to display hierarchical data in columns and rows in a structured way, in which subordinate data can be displayed and hidden row by row. An indicator on the rows shows whether the subordinate rows are shown or hidden.

In most cases, the first row contains the column headings and the first column frequently contains the row headings. Hierarchical tables can contain interactive elements, such as buttons for performing an action or check boxes for selecting a table row.

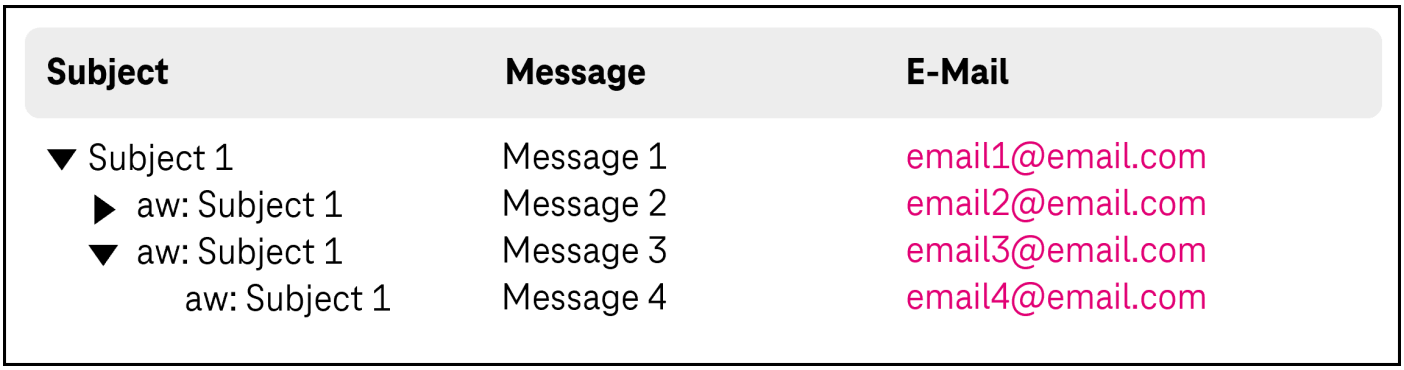

Figure 24: Hierarchical table

Presentation

The requirements regarding the table are described in the “table” section. Here, only the additional requirements are described which result from the ability to show and hide subordinate rows in the table.

No.	Property	Description	Classification	Reference
454	Contrast	Icons that display the status of table rows (shown or hidden) must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

Operation

The requirements regarding the table are described in the “table” section. Here, only the additional requirements are described which result from the ability to show and hide subordinate rows in the table.

No.	Property	Description	Classification	Reference
455	Click area	The click area of the elements for showing and hiding subordinate rows should total at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: hierarchical table

Action	Key	Classification
Show and hide subordinate rows	Not standardized, i.e. the operation should be described in the application and the Help option	Required
Show and hide subordinate rows	Double-click on parent rows	Recommended

Use of the pointing device: hierarchical table

Action	Key	Classification
Show and hide subordinate rows	Left click on the show and hide icon	Required
Show and hide subordinate rowsn	Double-click on parent rows	Recommended

Programming/interfaces

The requirements regarding the table are described in the “table” section. Here, only the additional requirements are described which result from the ability to show and hide subordinate rows in the table.

No.	Property	Description	Classification	Reference
413	Role	The hierarchical table role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 9.4.1.2, 11.4.1.2, 11.5.2.5
414	Status	The status of the table cells must be communicated to the Accessibility API (see Element status). Note: This also applies to the “open” or “closed” status (with respect to the subordinate rows).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
415	Desktop: Element hierarchy	The parent / child relationships of the elements within the hierarchical table must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9
416	Desktop: Element hierarchy	The hierarchy level of the table rows must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9

Title bar

Online betrachten

Synonyms: Title

See also: Heading, status bar

The title bar is the top element of the application window and contains the title (see DIN EN ISO 9241-161: 8.47).

The text in the title bar is usually one-line. The title bar can contain an application icon. The title of the window is also displayed by the icons in the task bar or by pressing ALT+TAB when switching between the applications. The title bar often contains interactive elements (for closing, scaling, and minimizing the application) which are not in the tab loop.

Only those requirements that are directly related to the title bar are described below. Requirements of elements within the title bar are described at the respective element.

Note: Most of the requirements concerning the title bar refer to desktop applications only. For Web applications, the browser is the window with the title bar. The Web application itself does not contain a title bar. The labeling of the title bar of the browser is defined using the <title> element, however.



Figure 25: Title bar

Presentation

No.	Property	Description	Classification	Reference
417	Label	The label of the title bar must be expressive. Note: The title bar should contain the application name and, if applicable, the document title /file name or the purpose /function of the window.	Must	EN 301 549: 9.2.4.2 11.2.4.6

Operation

No.	Property	Description	Classification	Reference
418	Desktop: Use of the keyboard	The control elements for closing, scaling and minimizing the application window must be operable with the keyboard (see Window).	Must	EN 301 549: 11.2.1.1
419	Desktop: Use of the keyboard	All other control elements within the title bar must be operable with the keyboard. The operation of these elements is described at the respective element.	Must	EN 301 549: 11.2.1.1
420	Click area	The click area of the control elements in the title bar should total at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Programming/interfaces

No.	Property	Description	Classification	Reference
421	Desktop: Operation	It must be possible to operate the buttons in the title bar with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17

Status bar

Online betrachten

Synonyms: Footer, status information

See also: Title bar

A status bar is used for the display of status information (see DIN EN ISO 9241-161: 8.40). The status bar can also contain control elements. The status bar is usually located at the lower border of the window. The use of a status bar within the application window is optional.

Only those requirements that are directly related to the status bar are described below. Requirements of elements within the status bar are described at the respective element.



Figure 26: Status bar

Operation

No.	Property	Description	Classification	Reference
422	Use of the keyboard	The control elements of the status bar must be operable with the keyboard (see following table on use of the keyboard).	Must	EN 301 549: 11.2.1.1
423	Click area	The click area of the control elements in the status bar should total at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: status bar

Action	Key	Classification
Navigating to the status bar	• Desktop: F6 or TAB • Web: TAB or documented keyboard shortcut	Required
Navigation from the status bar	• Desktop: F6 or TAB • Web: TAB or documented keyboard shortcut	Required
Navigating within the status bar	TAB	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
424	Role	The status bar role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.4.1.2, 11.5.2.5
425	Update	Important status messages must be marked so that they are output by the assistive technology without receiving the keyboard focus. Note: Important status messages are error messages, for instance. A status message about the amount of characters entered is considered unimportant in a word processing system, especially as this changes constantly following the input.	Must	EN 301 549: 11.4.1.3.1

Modal dialog

Online betrachten

Synonyms: Dialog window, message, pop-up, dialog, dialog box

See also: Window, tooltip, error message

A modal dialog is used to display important information and control elements in a separate area. A modal dialog blocks the operation of the application window in the background (see DIN EN ISO 9241-161: 8.10).

A modal dialog always contains a text and one or more buttons that can be used to close the dialog. The modal dialog can also contain a title bar and status bar as well as other control elements, graphics, etc.

The requirements regarding the content of the modal dialog are described at the respective element. Only specific requirements for modal dialog are described here.

Examples:

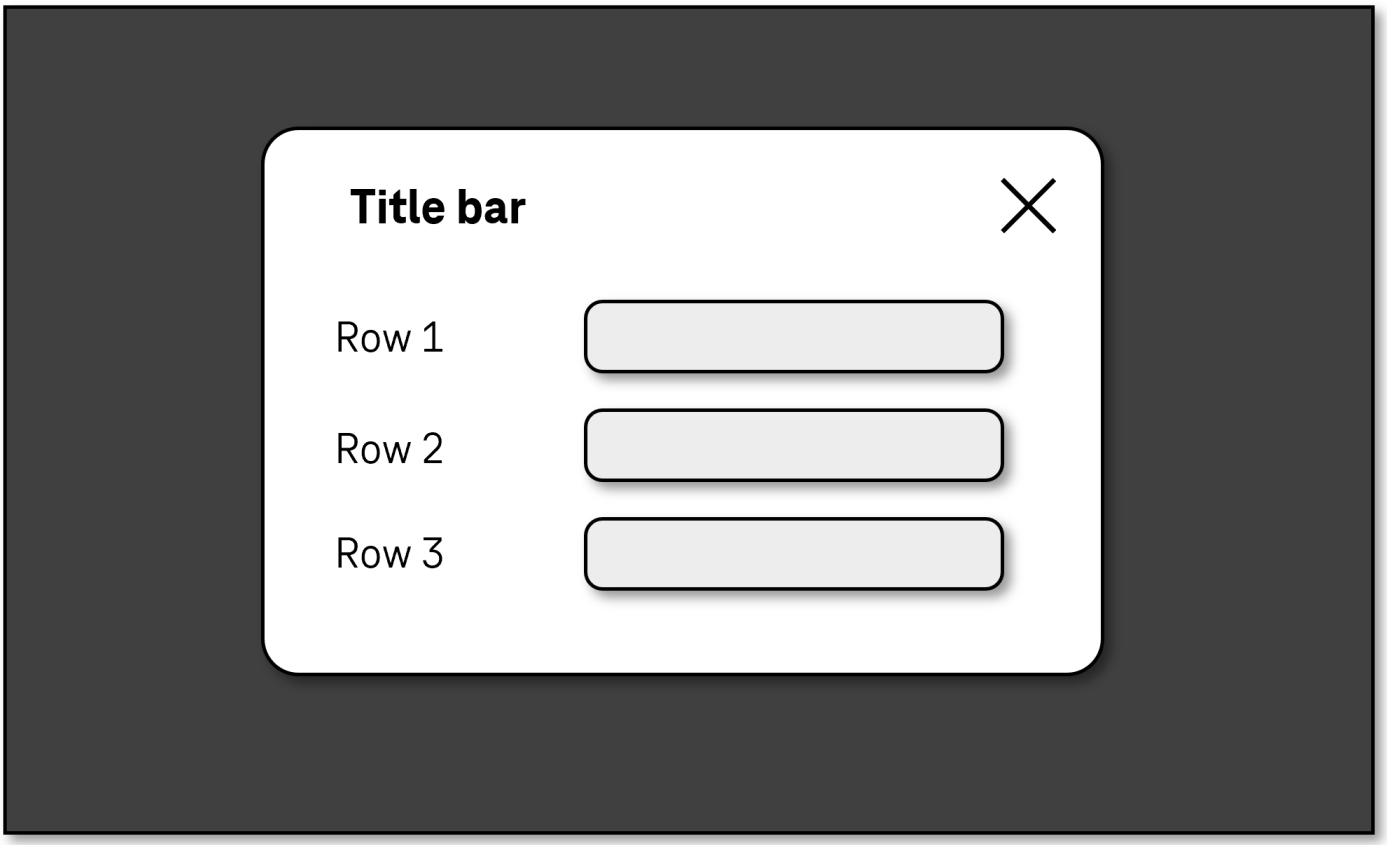

Figure 27: Modal dialog

Presentation

No.	Property	Description	Classification	Reference
426	Resizing	It must be possible to scale the modal dialog by up to 200%. During the scaling, no loss of content or functionality may occur (see Resizing).	Must	EN 301 549: 9.1.4.4, 11.1.4.4.1

No.	Property	Description	Classification	Reference
		Note: Fixed headers and footers in the modal dialog should be avoided, as they cause the main vertical scrollable area to become too small.		
427	Resizing	The modal dialog must be displayed in full without horizontal scrolling at a screen width of 320 px. If the modal dialog contains two-dimensional content such as tables, these should be horizontally scrollable (see Resizing).	Must	EN 301 549: 9.1.4.10, 11.1.4.10
428	Visibility	The dialog must stand out clearly from the background. Note 1: This can take place, for example, with a border around the dialog or the use of a gray background. Note 2: This also applies to the use of the Windows Contrast Adjustment .	Must	EN 301 549: 11.1.4.11; 9.1.4.11
429	Web: Label	If the modal dialog is displayed in a separate browser window, it must have an expressive label. Note: For the label, the <title> element is used.	Must	EN 301 549: 9.2.4.2
430	Label	The modal dialog must have an expressive label. Note: The label can be located in the title bar.	Must	EN 301 549: 9.2.4.5, 11.2.4.6

Operation

No.	Property	Description	Classification	Reference
431	Use of the keyboard	It must be possible to open, operate and close the modal dialog with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1
432	Use of the keyboard	When the modal dialog is opened, the keyboard focus must be placed in the dialog. Note: Typically, the keyboard focus should be placed at the beginning of the modal dialog. For simple modal dialogs, the focus can also be placed on a button at the end (e.g. “OK” or “Cancel”), provided that it is ensured that the title and text of the dialog are automatically output by the screen reader when the dialog is opened.	Must	EN 301 549: 9.2.4.3, 11.2.4.3
433	Use of the keyboard	As long as the modal dialog is open, the keyboard focus must remain within the dialog.	Must	EN 301 549: 9.2.4.3, 11.2.4.3
434	Use of the keyboard	When the modal dialog is closed, the keyboard focus must be reset to the triggering element or to an element with which the work can ultimately be continued. Note: The resetting to the triggering element is not possible, for instance, if it has been removed due to the operation of the dialog. In this case, it is usually a good idea to place the keyboard focus on an element before or after the triggering element.	Must	EN 301 549: 9.2.4.3, 11.2.4.3

Use of the keyboard: modal dialog

Action	Key	Classification
Navigating within the modal dialog	TAB	Required
Closing the modal dialog	• Desktop: ESC, ALT+F4 • Web: ESC	Required

Use of the pointing device: modal dialog

Action	Key	Classification
Closing the modal dialog	Left click on the corresponding button	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
435	Role	The modal dialog role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
436	Name	If the modal dialog has a label or description, these must be communicated as the Accessible Name and/or Accessible Description (see Label and Description). Note 1: The title bar or main heading of the dialog should be used as the Accessible Name. Note 2: If the modal dialog contains only a small amount of text, this can be communicated as the Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
437	Desktop: Element hierarchy	The parent / child relationships of the elements within the dialog must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9
438	Position	The size and position of the modal dialog must be communicated to the Accessibility API (see focus visibility).	Must	EN 301 549: 11.5.2.5

Control elements

Online betrachten

Table of contents

- Section separator
- Link
- Button
- Switch
- Toggle switch
- Split button
- Context menu
- Menu
- Menu button
- Tab group
- Input field (one-line)
- Input field (multi-line)
- Password input field
- Input field with autocomplete function
- Spin button
- Selection list
- Multiple selection list
- Tree structure
- Control point
- Combined input field
- Drop-down list
- Slider
- Scrollbars
- Radio buttons
- Check box

Section separator

Online betrachten

Synonyms: Separator, window splitter, splitter

See also: Control point, slider, scrollbar

A section separator is used to scale a page area or two neighboring page areas. Section separators are also used to scale table columns and rows.

A section separator is located between two page areas and consists of a bar and in some cases a control point. Section separators can have different manifestations:

- arranged horizontally or vertically,
- continuous or incremental scaling,
- scaling in the range of 0 to 100% (i.e., between hidden and full screen), or with a restricted area,
- scaling of both areas or only one of the two areas (the other area then moves, but remains unchanged in its size).

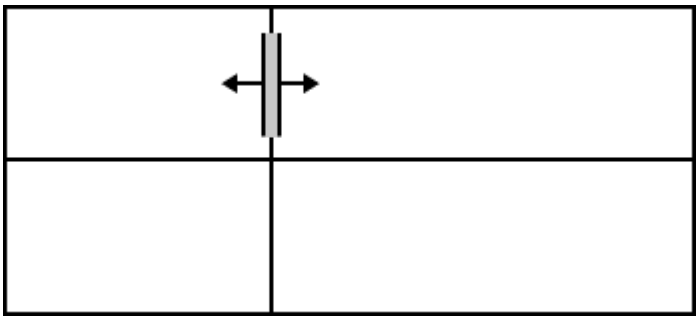

Figure 28: Section separator

Presentation

No.	Property	Description	Classification	Reference
438	Contrast	The bar or control point of the section separator must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
439	Focus visibility	If the section separator receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference

No.	Property	Description	Classification	Reference
440	Use of the keyboard	It must be possible to access, operate and exit the section separator with the keyboard (see Use of the keyboard table, below). Alternatively, it must be possible to scale the area with the keyboard when the focus is in the scalable area. In this case, the use of the keyboard for the section separator must be explained in the application and/or Help option. In addition, it must then be ensured that an area which is hidden by operation of the section separator can also be displayed again. Exception: If the section separator has no relevant function, it does not have to be keyboard-operable. This is the case, for example, if the section separator is used to scale page areas, if all content is fully perceptible in the standard display, and if the scaling does not add any value.	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2

No.	Property	Description	Classification	Reference
441	Use of the pointing device	The use of the pointing device for the section separator may not be complex. Please note: Complex use of the pointing device means • multipoint operation (e. g. swiping with several fingers), • path-based operation (where the start and end points of the use of the pointing device are not just relevant, but at least one intermediate point is).	Must	EN 301 549: 9.2.5.1, 11.2.5.1
442	Use of the pointing device	It should also be possible to operate the section separator without the dragging use of the pointing device. Note: This can be achieved, for example, by clicking on the section separator and then clicking on the target position.	Should	WCAG 2.2
443	Updates	When focusing and operating the section separator, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1, 9.3.2.2, 11.3.2.2
444	Click area	The click area of the section separator should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2: 2.5.8 (AA)

Use of the keyboard: section separator

Note: The following requirements only apply if the section separator receives the focus with the keyboard.

Action	Key	Classification
Focusing of the section separator	TAB	Required
Exiting the section separator	TAB	Required
Operation of the section separator	UP/DOWN ARROW, LEFT/RIGHT ARROW (depending on the orientation of the section separator)	Required
Operation of the section separator (minimum and maximum scaling)	POS1, END	Recommended
Switch between current, minimum and maximum scaling	ENTER SPACE	Recommended

Use of the pointing device: section separator

Action	Key	Classification
Operation of the section separator	Left click and drag on the bar or control point (drag and drop)	Required
Operation of the section separator	Left click to enable and move the pointing device, left click on the target position	Recommended

Programming/interfaces

No.	Property	Description	Classification	Reference
445	Role	The role of the section separator must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
446	Value	The value of the section separator must be communicated to the Accessibility API (see Accessibility API). Hinweis: Note: The value of the section separator is frequently given in the range 0% to 100%.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7
447	Desktop: Value range	The minimum and maximum value of the section separator must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.7

No.	Property	Description	Classification	Reference
448	Status	The status of the section separator must be communicated to the Accessibility API (see Element status).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
449	Orientation	The orientation of the section separator (vertical or horizontal) must be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2
450	Name	The section separator must have a concise and expressive Accessible Name. Note: A section separator typically has no visible label. The name of the section separator can contain the Accessible Name of the scalable areas.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
451	Operation	If the section separator receives the keyboard focus, it must be possible to access, operate and exit it with assistive technology. Otherwise, the alternative form of operation must be operable with assistive technology (see Accessibility API).	Must	EN 301 549: 11.4.1.2, 11.5.2.12, 11.5.2.17
452	Update	Updates concerning the Accessible Name, value or status of the section separator must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
453	Desktop: Position	The size and position of the control point (if available) and/or the section separator (if without control point) must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5

Link

Online betrachten

Synonyms: Cross-reference, reference, hypertext link, hyper link

See also: Button, Menu, Image Map

Links allow navigation to a defined location (see DIN EN ISO 9241-161: 8.23).

A link has a text or graphic label (e.g. an icon). Links usually have a visual indicator to identify the link as such, this applies to text links in particular. An underline and a different color are typically used as visual indicators for text links.

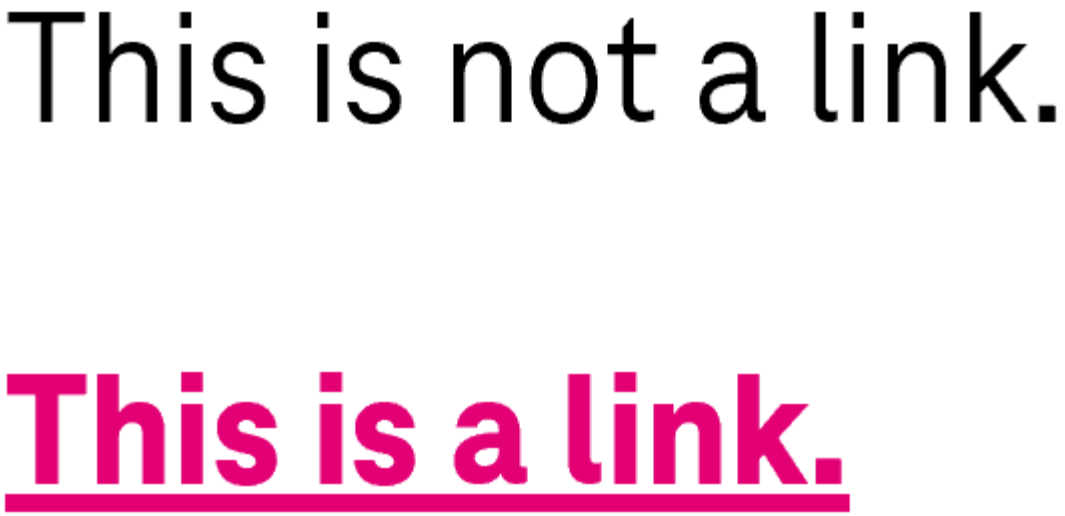

Figure 29: Text in normal font and black, link in bold, underlined and colored

Presentation

No.	Property	Description	Classification	Reference
454	Contrast	If the link has a text label, it must have a contrast ratio of at least 4.5: 1 with respect to the background.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
455	Contrast	If the link has a graphic label, it must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

No.	Property	Description	Classification	Reference
456	Contrast	If the link is only identifiable as a link on the basis of its text color, the contrast ratio between the color of neighboring text contents and the text color of the link must be at least 3:1.	Must	EN 301 549: 9.1.4.1, 11.1.4.1
457	Contrast	If the link is only identifiable as a link due to a graphic element (e.g. underline or icon), the contrast ratio between the graphic element and the background color must be at least 3:1.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
458	Contrast	If the status of the link is communicated in graphical form, the contrast ratio between the graphic element and the background color must be at least 3:1. Note: The status also refers to the following information: • linked document format (e.g. PDF), • link destination (e.g. internal or external links), • link type (e.g. link to a phone number or email address), • position in the application (for example, link to the current page).	Must	EN 301 549: 9.1.4.11, 11.1.4.11
459	Label	The visible label of the link must correspond to or be contained in the link name which is communicated to the Accessibility API (see Label).	Must	EN 301 549 9.2.5.3, 11.2.5.3

No.	Property	Description	Classification	Reference
460	Label	The purpose of the link must be determinable from the link label or the programmatically linked link context. Note 1: In the case of desktop applications, e.g. a tooltip, a label of a group or a description can be considered the link context as long as they are programmatically linked with the link. Note 2: In the case of Web applications, text in the same paragraph or sentence, in the same list entry or a parent list entry, in the section header, in the same table cell, or in an associated column or row heading is considered to be a programmatically determinable link context.	Must	EN 301 549: 11.2.4.4
461	Label	It should be possible to determine the purpose of the link from the link label. Note: This is important because the programmatically determinable link context often cannot be determined, or can be determined only with difficulty, using the screen reader. This applies to the link overview in particular which the screen reader is able to display.	Should	WCAG 2.1: 2.4.9 (AAA)
462	Label	If the link has a graphic label, the link should have a tooltip with a text label.	Should	WCAG 2.1: 3.3.5 (AAA); DIN EN ISO 9241-143: 9.6.11

No.	Property	Description	Classification	Reference
463	Label	If the link refers to a destination in another program or document format, reference should be made to it on the link.	Should	WCAG 2.1: 3.2.5 (AAA)
464	Web: Consistency	If a list of links is used for navigation, the links on each screen should be displayed in the same relative order and receive the keyboard focus (see Consistency).	Must	EN 301 549: 9.3.2.3
465	Desktop: Consistency	If a list of links is used for navigation, the links on each screen should be displayed in the same relative order and receive the keyboard focus (see Consistency).	Should	WCAG 2.1: 3.2.3 (AA)
466	Position	If a list of links is used for navigation, then the link to the current page or to the area in which the current page is located should be marked.	Should	WCAG 2.1: 2.4.8 (AAA)
467	Focus visibility	If the link receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
468	Use of the keyboard	It must be possible to access, operate and exit the link with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
469	Use of the keyboard	When enabling links within pages, the linked element must be focused. It is not sufficient if the linked element is only scrolled into the visible area.	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.4.3, 11.2.4.3

No.	Property	Description	Classification	Reference
470	Web: Use of the keyboard	It must be possible to skip areas of content with links that occur on several pages using the keyboard (see Practical tip: efficient keyboard navigation). Note: This applies to the navigation links at the top of the page in particular.	Must	EN 301 549: 9.2.4.1
471	Use of the keyboard	If a window contains areas with several links, a mechanism should be implemented for skipping these page areas with the keyboard (see Practical tip: efficient navigation).	Should	DIN EN ISO 9241-171: 9.3.16, 9.3.17
472	Use of the keyboard	Consecutive links with the same link destination should be avoided. Note: This applies, among others, to tiles, in which both the heading and a text at the end (e. g. “read more”) are frequently linked, as well as to links that have a text and graphic label.	Should	WCAG 2.1: 2.4.9 (AAA)
473	Updates	When a link is focused, no change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1
474	Click area	If the link is not located within continuous text, the click area of the link should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2: 2.5.8 (AA)

Use of the keyboard: link

Action	Key	Classification
Focusing of the link	TAB	Required
Exiting the link	TAB	Required
Enabling the link	ENTER	Required

Use of the pointing device: link

Action	Key	Classification
Enabling the link	Left click	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
475	Role	The role of the link must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
476	Status	The status of the link must be communicated to the Accessibility API (see Element status). Note: This also applies to information • about the linked document format (e.g. PDF), • about the link destination (e. g. internal or external links), • about the link type (e.g. link to a phone number or email address), • about the position in the application (for example, link to the current page), provided that this information is communicated in visual form.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
477	Name	The link must have a concise and expressive Accessible Name. Note: If the Accessible Name is not expressive, the programmatically perceptible link context can also be used to communicate the purpose of the link.	Must	EN 301 549: 9.2.4.4, 11.2.4.4, 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5
478	Name	If the link has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.2.4.4, 11.2.4.4, 9.4.1.2, 11.4.1.2, 11.5.2.5
479	Operation	It must be possible to access, operate and exit the link with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
480	Update	Updates concerning the Accessible Name or status of the link must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
481	Desktop: Position	The size and position of the link must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: links in Web applications

Screen reader output

- JAWS: [label] **link** | visited link
- NVDA: [label] **link** | visited link
- Windows Narrator: **link** [label] value [URL]

Note:

- JAWS and NVDA distinguish between visited and unvisited links.
- With JAWS, the link type of some links is perceptible (link to email and FTP addresses, page-internal links).

- With Windows Narrator, the link type is perceptible on most links, as the URL is output as the value of the link. The output of the URL is not correct, however, which means that “http://” is output as “http skeptical smiley”.

HTML

The link should be implemented with the HTML element <a href=URL>. The href attribute is required.

Label:

- The labeling of the link should take place using the text content.
- In the case of graphic links, the labeling takes place using the alternative text of the graphic or aria-label. It should be noted that the link label does not describe the graphic, but the purpose of the link. If the content of the graphic is also relevant, the graphic should not be linked, but the link should be located before or after the graphic.
- The labeling of the links should be concise. For example: In the case of a tile with a heading, graphic, teaser text, keywords, etc., it is not necessary for the entire tile to be linked, but just the heading.
- The labeling of the links should be unambiguous and expressive. Links labeled “Click here” or “Further information” are not considered expressive. The label should either be expressive (e.g. “Further information on accessibility”), or the description should explain the purpose of the link (by referring to the relevant context of the link with aria-describedby, for example).

Status and type:

- Links cannot be marked as disabled (disabled). Links which are not operable can be marked with the <a> element without the href attribute.
- In the case of links that are used for page navigation, aria-current=page can be used to mark the link that refers to the current page.
- Links that are used to show and hide areas should be marked with aria-expanded to communicate the status of the areas (shown or hidden).
- Links that open with a pop-up should be marked with aria-haspopup.
- Unless the link opens a new Web page in the current browser window, a visual reference should be made to the deviating link destination (e.g. the link opens a page in the new browser window, the link refers to a PDF document).
- If the links have a graphic reference that refers to the type of link (e.g. link to an external page, link opens a page in the new browser window, link refers to a PDF document, linked email address), this information should also be perceptible with assistive technology. The type of link is generally a secondary piece of information, which means that it should not be output as part of the label, but as a description (e.g. using title or aria-describedby attribute), or at least only at the end of the label.

Further information: 4.6 Links - HTML Standard (whatwg.org) (External Link)

ARIA

If the link is not implemented with the HTML element, in addition to the notes on HTML links, it is necessary to take account of the following:

- The role is communicated with role=link.
- Disabled links can be marked with aria-disabled.
- The communication of the information that a link has already been visited or that it consists of a page-internal link is not possible with ARIA links.

- The links should be visually marked (with an icon or underline, for example) so that their role is also perceptible when using Windows Contrast Adjustment.

Further information: link role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org), Link Pattern | APG | WAI | W3C

Button

Online betrachten

Synonyms: Command button, push button

See also: Menu button, link, toggle switch, switch, tab, accordion

Buttons are used to execute a command (see DIN EN ISO 9241-161: 8.32).

A button consists of a text or graphic label and a visual indicator in order to identify the button as such. A border is usually used as a visual indicator for buttons.

Buttons can be used within other elements, e.g. in tables, tree structures, toolbars, scrollbars, tab panels containing tabs, accordions, input fields, spin buttons, drop-down lists and combined input fields. Deviating requirements concerning the buttons are described in the sections about these elements



Figure 30: Confirm button

Presentation

No.	Property	Description	Classification	Reference
482	Contrast	If the button has a text label, this must have a contrast ratio of at least 4.5:1 with respect to the background.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
483	Contrast	If the button has a graphic label, this must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
484	Contrast	If the button is only identifiable as such on the basis of its color design, the color must have a contrast ratio of at least 3:1 with respect to the neighboring colors. Note 1: A button might be recognizable as an interactive element on the basis of its border or its background color, for example. Note 2: The requirement does not apply if the button is clearly identifiable as a button, because of its position or labeling, for example.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
485	Label	The visible label of the button must correspond to or be contained in the name of the button which is communicated to the Accessibility API (see Label).	Must	EN 301 549 9.2.5.3, 11.2.5.3
486	Label	If the button has a graphic label, the button should have a tooltip with a text label.	Should	WCAG 2.1: 3.3.5 (AAA); DIN EN ISO 9241-143: 9.6.11
487	Focus visibility	If the button receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
488	Use of the keyboard	It must be possible to access, operate and exit the button with the keyboard (see Use of the keyboard table).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
489	Use of the keyboard	Frequently used buttons should have a keyboard shortcut which is documented in the application and the Help option.	Should	DIN EN ISO 9241-171: 9.3.10
490	Web: Use of the keyboard	It must be possible to skip areas of content with buttons that occur on several pages using the keyboard (see Practical tip: efficient navigation).	Must	EN 301 549: 9.2.4.1
491	Use of the keyboard	If a window contains areas with several buttons, a mechanism should be implemented for skipping these page areas with the keyboard (see Practical tip: efficient navigation).	Should	DIN EN ISO 9241-171: 9.3.16, 9.3.17
492	Use of the pointing device	It must be possible to undo or cancel the accidental enabling of the button. To do this, the button may only be enabled when it is released, i.e. only upon the up event and not upon the down event (see Use of the pointing device).	Must	EN 301 549: 9.2.5.2, 11.2.5.2
493	Updates	When the button is focused, no change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1
494	Updates	If the enabling of the button causes a new window or a modal pop-up to open, the	Must	EN 301 549: 9.2.4.3, 11.2.4.3

No.	Property	Description	Classification	Reference
		focus must be placed in the window and/or the pop-up (see Navigation sequence).		
495	Updates	If an unannounced change of context is performed through the enabling of the button within the current program interface, this must take place according to the current position of the keyboard focus, or the focus must be placed at the beginning of the updated area.	Must	EN 301 549: 11.2.4.3
496	Updates	If an update is performed through the enabling of the button within the current program interface that requires an interaction on the part of the user, this should take place according to the current position of the keyboard focus. The keyboard focus should remain on the button or be placed at the beginning of the updated area (see Change of context). Note: User interaction can mean reading a piece of information or the operation of interactive elements.	Should	WCAG 2.1: 3.2.5 (AAA)
497	Updates	If the button is removed from the program interface as a result of its enabling, the focus should be placed on a neighboring control element or on a control element with which the work can be continued in a coherent way.	Should	WCAG 2.1: 3.2.5 (AAA)

No.	Property	Description	Classification	Reference
		Note: This applies, for example, to Delete buttons which refer to a single element.		
498	Click area	The click area of the button should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: button

Action	Key	Classification
Focusing of the button	TAB	Required
Exiting the button	TAB	Required
Enabling the button	SPACE, ENTER	Required

Use of the pointing device: button

Action	Key	Classification
Enabling the button	Left click	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
499	Role	The button role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 1.4.1.2, 11.5.2.5
500	Status	The status of the button must be communicated to the Accessibility API (see Element status).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
501	Name	The button must have a concise and expressive Accessible Name .	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.

No.	Property	Description	Classification	Reference
502	Name	If the button has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
503	Name	If the purpose of the button is determined by the visual context, this context must be communicated as part of the Accessible Name or Accessible Description. Note: The Accessible Name of a button cannot, therefore, be simply “Close” or “Delete” if only the visual context indicates which element is closed or deleted. Instead, the Accessible Name must be “Close pop-up” or “Delete file [file name]”, for instance. Alternatively, the Accessible Description must contain a reference to the element which is to be closed or deleted.	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 9.2.4.6, 11.2.4.6, 9.3.3.2, 11.3.3.2
504	Keyboard shortcut	If the button has a visually perceptible keyboard shortcut, it must be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
505	Operation	It must be possible to access, operate and exit the button with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
506	Update	Updates concerning the Accessible Name or status of the button must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15

No.	Property	Description	Classification	Reference
507	Desktop: Position	The size and position of the button must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: buttons in Web applications

Screen reader output

- JAWS: [label] **button** [note on operation with the Enter key]
- NVDA: [label] **button** | graphics button
- Windows Narrator: [label] **button**

Please note:

- The type of button (type=reset, type=submit, type=button) is not perceptible with the screen readers.
- Only NVDA outputs the element <input type=image> as a “graphic button”.

HTML

The button should be implemented with the HTML elements <button> or <input type=button>. For graphic buttons, <input type=image> can be used. In forms, <input type=submit> and/or <input type=reset> can be used for the buttons for sending and resetting the inputs.

Label:

- The labeling of the buttons that are labeled with <button> should take place using the text content.
- The labeling of the buttons that are labeled with <input> should take place using the value attribute.
- Buttons that are labeled with <input type=submit> or <input type=reset> do not need to be explicitly labeled because they are labeled automatically by the browser (with “Send” and “Reset”, for instance). The default labeling of the browser can be overridden with the value attribute, however. This is recommended if reference is made to these buttons in the application or the Help option, because the label of the buttons is otherwise different according to the browser.
- In the case of graphic buttons, the labeling takes place using the alternative text of the graphic (e. g. <input type=image alt=...> and/or <button><img alt=...></button>) or with aria-label. It should be noted that the button label does not describe the graphic, but the purpose of the button.
- The labeling of the buttons should be concise, unambiguous and expressive. Buttons labeled “Click here” or “Detailed view” are not considered expressive. The label should either be expressive (e.g. “Detailed view of Joe Bloggs”), or the description should explain the purpose of the button (with aria-describedby or title, for example).

Status and type:

- Buttons can be marked as disabled (disabled).
- With buttons that are used for navigation or pagination, aria-current can be used to mark the button which refers to the current element.
- Buttons that are used to show and hide areas should be marked with aria-expanded to communicate the status of the areas (shown or hidden).

- Buttons that open a pop-up should not be labeled aria-haspopup, even if the attribute is intended for that purpose. This is because, when combined with a button, aria-haspopup causes the element to be output as a menu button. Instead, links should be used with aria-haspopup r buttons that have a text note in the description.

Further information: 4.10.6 The button element - HTML Standard (whatwg.org), 4.10.5.1.21 Button state (type=button) - HTML Standard (whatwg.org), 4.10.5.1.19 Image Button state (type=image) - HTML Standard (whatwg.org), 4.10.5.1.18 Submit Button state (type=submit) - HTML Standard (whatwg.org), 4.10.5.1.20 Reset Button state (type=reset) - HTML Standard (whatwg.org)

ARIA

If the button is not implemented with the HTML element, in addition to the notes on HTML buttons, it is necessary to take account of the following:

- The role is communicated with role=button.
- Disabled buttons can be marked with aria-disabled.
- The buttons should be visually marked (with a border, for example) so that their role is also perceptible with the use of Windows Contrast Adjustment.

Further information: button role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org), Button Pattern | APG | WAI | W3C

Switch

Online betrachten

Synonyms: Switch button, toggle switch

See also: Button, check box, menu switch, toggle switch

A switch is used to select the values “On” or “Off”. The switch can also be used for other value pairs, provided that these values are communicated in text form.

A switch has a border that contains a visual indicator which displays the selected value on the basis of its position. The indicator is usually a circle which is positioned on the left (“Off” value) or right (“On” value) in the border.

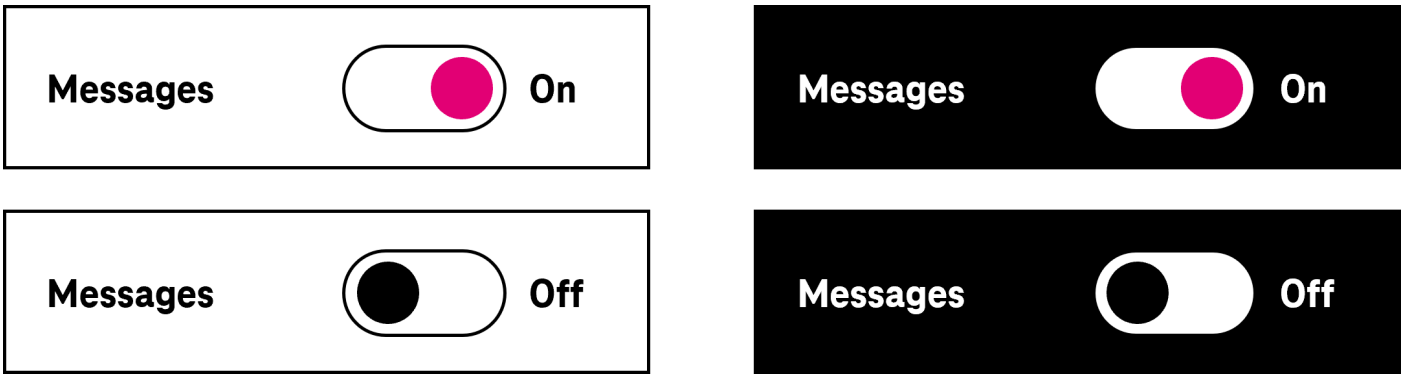

Figure 31: Switch in the On and Off state

Presentation

The requirements concerning buttons are described in the “buttons” section. Here, only the additional requirements are described which result from the fact that it is a button that is able to have two states.

No.	Property	Description	Classification	Reference
508	Contrast	The visual indicator for the value must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
509	Contrast	If the value of the switch is also communicated in text form, it must have a contrast ratio of at least 4.5: 1 with respect to the background.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
510	Value	The value of the switch should also be communicated in text form. Note: Although “On” is associated with the right position and “Off” is associated with the left position, it cannot be assumed that all users are familiar with this. In addition, the value is often color coded (“On” in green or a dark color, “Off” in red or a light color). The colors may not be perceivable by people who are disabled, however.	Should	DIN EN ISO 9241-143: 9.6.11

Operation

The requirements concerning buttons are described in the “buttons” section. Here, only the additional requirements are described which result from the fact that it is a button that is able to have two states.

No.	Property	Description	Classification	Reference
511	Updates	When focusing and operating the switch, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.2, 11.3.2.2

Use of the keyboard: switch

Action	Key	Classification
Operating the switch	SPACE	Required
Operating the switch	ENTER	Recommended

Use of the pointing device: switch

Action	Key	Classification
Operating the switch	Left click	Required

Programming/interfaces

The requirements concerning buttons are described in the “buttons” section. Here, only the additional requirements are described which result from the fact that it is a button that is able to have two states.

No.	Property	Description	Classification	Reference
512	Role	The switch role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
513	Value	The value of the switch must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7
514	Value	If the values of the switch do not represent “On” and “Off”, but are used for other states which are visible in text form on the switch, these values must be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7

Practical tip: switches in Web applications

Screen reader output

- JAWS: [label] **switch** [off | pressed on]
- NVDA: [label] **switch** [off | on]]
- Windows Narrator: [label] **button** [off | on]

Note:

- Although there is an individual role for switches with role=switch, it is not output by the screen readers. Instead, the switches are output as toggle switches and/or check boxes. This is not problematic, as the functionality is analog.
- The only problem is that the value in text form (if present) is imperceptible with the screen reader.

HTML

There is no element for switches in HTML. Toggle switches or check boxes can be used instead.

ARIA

In this respect, with switches, the following should be taken into account:

- The role is communicated with role=switch.
- The status is communicated with aria-checked=true|false and must be updated during operation.
- The labeling can take place by text content or aria-labelledby.
- The switch can be marked as read-only with aria-readonly and as disabled with aria-disabled.
- A switch can be marked as a required field with aria-required. This is only useful in cases where the switch has to be submitted in the “On” status (aria-checked=true). However, if at least one switch in a group of switches must be selected, the required field label should be carried out with the group (e.g. asterisk within the group label).
- The status of the switch should be displayed in text form, as color information (such as green = on and red = off) may be imperceptible for visually impaired people.
- The switch should only be used for the “on” and “off” status, because other status information is not communicated by the screen reader. Alternatively, other status information should be communicated as part of the label or description. In this case, however, it should be noted that the programmatic status is also output, which leads to a redundant output.
- The presentation of the switch should be verified in Windows High Contrast mode. The switch itself, for example, and the visual indicator of the status should have a border and not just be marked with background color.
- The visible switch and the programmatically focused element should have the same position and size.

Further information: switch role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, Switch Pattern | APG | WAI | W3C

Toggle switch

Online betrachten

Synonyms: Toggle button

See also: Button, check box, menu button, switch

Toggle switches are used to select the “pressed” or “not pressed” states (see DIN EN ISO 9241-161: 8.48).

A toggle switch consists of a text or graphic label and a visual indicator in order to identify the toggle switch as such. A border is usually used as a visual indicator for toggle switches. The toggle switch also has a visual indicator of the state, such as a different background color.

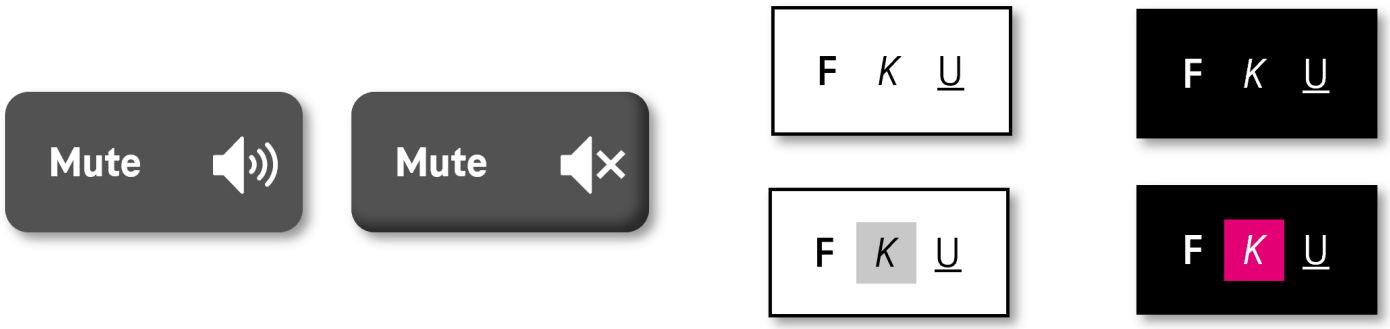

Figure 32: Toggle switch in the pressed and not pressed status

Presentation

The requirements concerning buttons are described in the “buttons” section. Here, only the additional requirements are described which result from the fact that it is a button that is able to have two states.

No.	Property	Description	Classification	Reference
515	Contrast	If the state of the toggle switch (pressed, not pressed) is only communicated with a different color, the contrast ratio of the colors must be at least 3:1. Note: To ensure that the status of the toggle switch is visible when using Windows Contrast Adjustment, it should not only be communicated in color. An icon or border effect can be used to communicate the status instead.	Must	EN 301 549: 9.1.4.1, 11.1.4.1

Operation

The requirements concerning buttons are described in the “buttons” section. Here, only the additional requirements are described which result from the fact that it is a button that is able to have two states.

No.	Property	Description	Classification	Reference
516	Updates	When focusing and operating the toggle switch, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.2, 11.3.2.2

Use of the keyboard: toggle switch

Action	Key	Classification
Operation of the toggle switch (status change between “pressed” and “not pressed”)	SPACE	Required
Operation of the toggle switch (status change between “pressed” and “not pressed”)	ENTER	Recommended

Use of the pointing device: toggle switch

Action	Key	Classification
Operation of the toggle switch (status change between “pressed” and “not pressed”)	Left click	Required

Programming/interfaces

The requirements concerning buttons are described in the “buttons” section. Here, only the additional requirements are described which result from the fact that it is a button that is able to have two states.

No.	Property	Description	Classification	Reference
517	Role	The toggle switch role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
518	Status	The status of the toggle switch must be communicated to the Accessibility API (see Element status). Note: This also applies to the “pressed” or “not pressed” status.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

Practical tip: toggle switch in Web applications

Screen reader output for toggle switch with aria-pressed

- JAWS: [label] **toggle button** [not pressed | pressed] [note on operation with the Space bar]
- NVDA: [label] **toggle button** [not pressed | pressed]
- Windows Narrator: [label] **button** [off | on]

Screen reader output for toggle switch with aria-expanded

- JAWS: [label] **button** [collapsed | expanded] [note on operation with the Enter key]
- NVDA: [label] **button** [collapsed | expanded]
- Windows Narrator: [label] **button** [collapsed | expanded]

HTML

There is no element for toggle switches in HTML. Instead of this, buttonsthat have alternative labeling (e.g. “Select” and/or “Cancel selection”), check boxes or the ARIA toggle switch can be used. Toggle switches which have the purpose of showing and hiding areas should be marked with <details> and <summary> (4.11.1 The details element - HTML Standard (whatwg.org) (External Link) , 4.11.2 The summary element - HTML Standard (whatwg.org) (External Link)). According to the HTML specification, the <summary> element may contain links, headings, input fields and many other elements – it is important to remember, however, that all the elements which are located within <summary> are imperceptible and cannot be operated with the screen reader, as the <summary> element is communicated to the Accessibility API as a button. Therefore, the <summary> element should only contain a concise and expressive label in text form.

ARIA

With toggle switches, the following should be taken into account:

- with a button (e.g. <button> or role=button), the role is communicated with the aria-pressed attribute.
- toggle switches that have the purpose of showing and hiding areas can instead be marked with the aria-expanded attribute. In this case, aria-controls can be used to make reference to the ID of the area that is shown or hidden.
- The status (aria-pressed=true|false or aria-expanded=true|false) must be updated during operation.
- The labeling can take place using text content or aria-labelledby.
- The toggle switch can be marked as disabled with aria-disabled.
- A toggle switch cannot be marked as read-only with aria-readonly or as a required field with aria-required because, in contrast to a switch, it is not considered to be a form element.
- The presentation of the toggle switch should be verified in Windows High Contrast mode. In this respect, the toggle switch should have a border and the visual indicator for the status should not just be conveyed with color.
- The visible toggle switch and the programmatically focused element should have the same position and size.

Further information: aria-pressed state - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link) , aria-expanded state - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)

, Button Examples | APG | WAI | W3C, Disclosure (Show/Hide) Pattern | APG | WAI | W3C

Split button

Online betrachten

Synonyms: Bifurcated switch

See also: Drop-down list, menu button, context menu, button

Split buttons are for the execution of a command that can be configured using a menu, a selection list or a dialog window.

Alternatively, split buttons are used for grouping related functions, where the primary function can be initialized directly using the button and the secondary functions can be initialized using the menu at the button.

A split button consists of a text or graphic label and a visual indicator in order to identify the button as such (usually a border). The split button also has additional buttons (with arrow icon), with which the secondary functions can be shown and hidden.

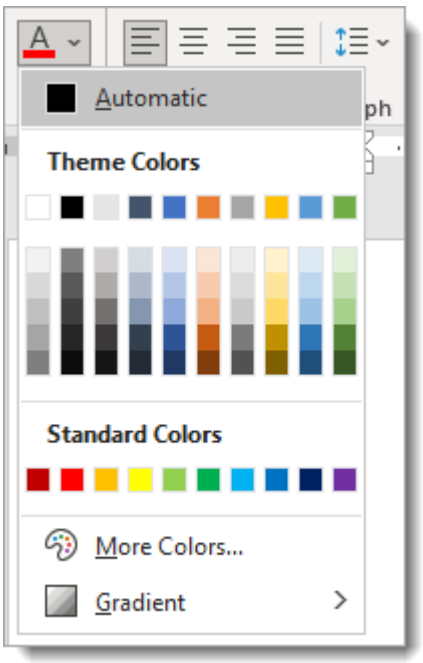

Figure 33: Split button for the selection of a color

Presentation

The requirements concerning buttons are described in the “buttons” section. Here, only the additional requirements are described which result from the fact that it is a button with primary and secondary functions. The requirements regarding the elements that can be shown are described at the respective element.

No.	Property	Description	Classification	Reference
519	Contrast	The arrow icon for opening and closing the menu or similar must have a contrast ratio of at least 3:1 with respect to the neighboring color.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
520	Contrast	If the selected entry only differs from the unselected entry in the open state due to its color (e.g. foreground or background color), the colors must have a contrast ratio of at least 3:1. Note: The selected entry does not have to be color coded or marked with color only. It can be marked with a check box, for example. In this case, the contrast requirements for the color coding are eliminated as long as the check box has sufficient contrasts.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

Operation

The requirements concerning buttons are described in the “buttons” section. Here, only the additional requirements are described which result from the fact that it is a button with primary and secondary functions. The requirements regarding the elements that can be shown are described at the respective element.

No.	Property	Description	Classification	Reference
521	Use of the keyboard	It must be possible to access, operate and exit the split button with the keyboard (see Use of the keyboard table, below). Note: The button for showing and hiding the secondary functions should not receive the keyboard focus separately.	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
522	Updates	If the function of the split button can be configured, no unexpected change of context may occur during or after the configuration.	Must	EN 301 549: 9.3.2.2, 11.3.2.2

Use of the keyboard: split button

Action	Key	Classification
Enabling the split button	• SPACE • ENTER	Required
Opening the menu or similar	ALT+DOWN ARROW	Required
Closing the menu or similar	• ESC • ALT+ UP ARROW • [Selection of a menu item or similar]	Required

Use of the pointing device: split button

Action	Key	Classification
Enabling the split button	Left click on the button label	Required
Opening the menu or similar	Left click on the arrow icon	Required
Closing the menu or similar	• Left click on the arrow icon • Left click on an entry in the menu or similar • Left click outside the split button	Required

Programming/interfaces

The requirements concerning buttons are described in the “buttons” section. Here, only the additional requirements are described which result from the fact that it is a button with primary and secondary functions. The requirements regarding the elements that can be shown are described at the respective element.

No.	Property	Description	Classification	Reference
522	Role	The split button role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
523	Value	If the current configuration of the split button is visible in the closed state, it must be communicated as a value to the Accessibility API. Note: If this is not possible, the value must be communicated as part of the label or description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

Practical tip: split buttons in Web applications

HTML

There is no element for split buttons in HTML. Instead, two buttons with expressive labeling (e.g. “Assign text color” and/or “Select text color”) can be used which receive the focus with TAB.

ARIA

There is no role for split buttons in ARIA. Instead, two separate buttons can be used as described above.

Context menu

Online betrachten

Synonyms: Pop-up menu, pie menu

See also: Menu button, menu

Context menus are used for displaying a context-specific menu with function for the element currently focused with the keyboard or moved over with the pointing device (see DIN EN ISO 9241-161: 8.29).

A context menu has one or more menu items which are mostly vertical. A menu item can have a sub-menu. The menu items of a sub-menu are also arranged vertically. Sub-menus can also be nested several times, i.e. a menu item in a sub-menu can also have a sub-menu.

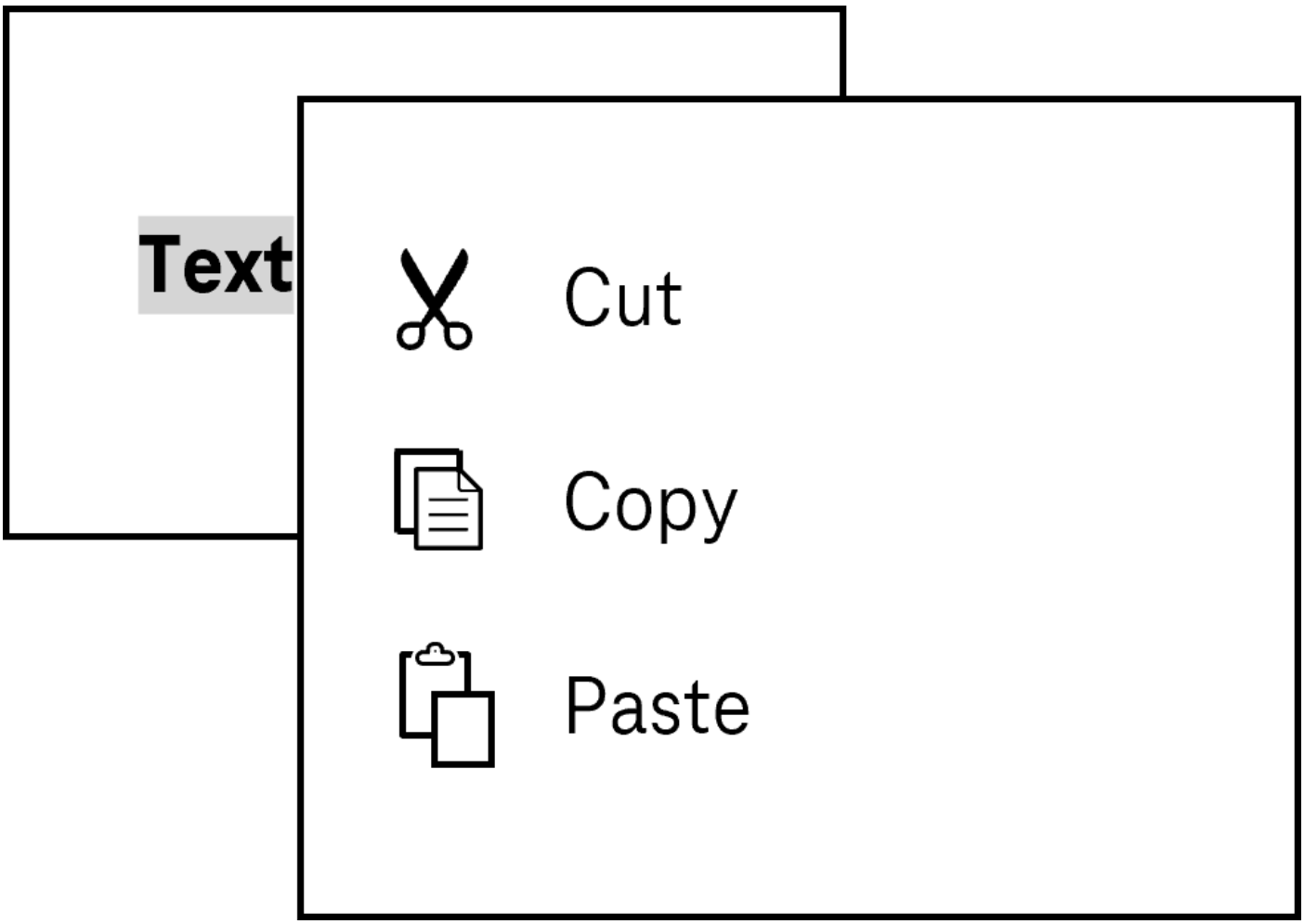

Figure 34: Context menu for the text editing

Presentation

The requirements concerning the menu are described in the “menu” section. Here, only the additional requirements are described which result from the fact that it is a context menu.

No.	Property	Description	Classification	Reference
524	Contrast	If the control element with context menu has a visual reference to the context menu, this must have a contrast ratio of at least 4.5:1 (text) and/or 3:1 (graphic) with respect to the background.	Must	EN 301 549: 9.1.4.3, 11.1.4.3, 9.1.4.11, 11.1.4.11

Operation

The requirements concerning the menu are described in the “menu” section. Here, only the additional requirements are described which result from the fact that it is a context menu.

No.	Property	Description	Classification	Reference
525	Use of the keyboard	The control element with context menu must be accessible with the keyboard, and it must be possible to open the context menu with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2

Use of the keyboard: context menu

Action	Key	Classification
Opening the context menu	CONTEXT MENU, SHIFT+F10	Required
Closing the context menu	ESC, SHIFT+F10, [to select a menu item]	Required

Use of the pointing device: context menu

Action	Key	Classification
Opening the context menu	Right click	Required
Closing the context menu	Left click on a menu item, click outside the context menu	Required

Programming/interfaces

The requirements concerning the menu are described in the “menu” section. Here, only the additional requirements are described which result from the fact that it is a context menu.

No.	Property	Description	Classification	Reference
526	Role	The context menu role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
527	Status	If the control element with context menu has a visual reference to the context menu, this reference must communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
528	Operation	The control element with context menu must be accessible with assistive technology, and it must be possible to open the context menu with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17

Menu

Online betrachten

Synonyms: Menu bar

See also: Context menu, menu button, toolbar, tree structure, tab group

Menus are used for selecting functions or for navigation (see DIN EN ISO 9241-161: 8.26).

A menu has several menu items which are usually arranged horizontally and adjacent to each other. A menu item can have a sub-menu. The menu items of a sub-menu are arranged vertically. Sub-menus can be nested several times, i.e. a menu item in a sub-menu can also have a sub-menu. It is only possible to display one sub-menu per hierarchy level. A menu can contain menu items that can be selected like check boxes or radio buttons. The menu items can be grouped. The label of the groups cannot be selected.

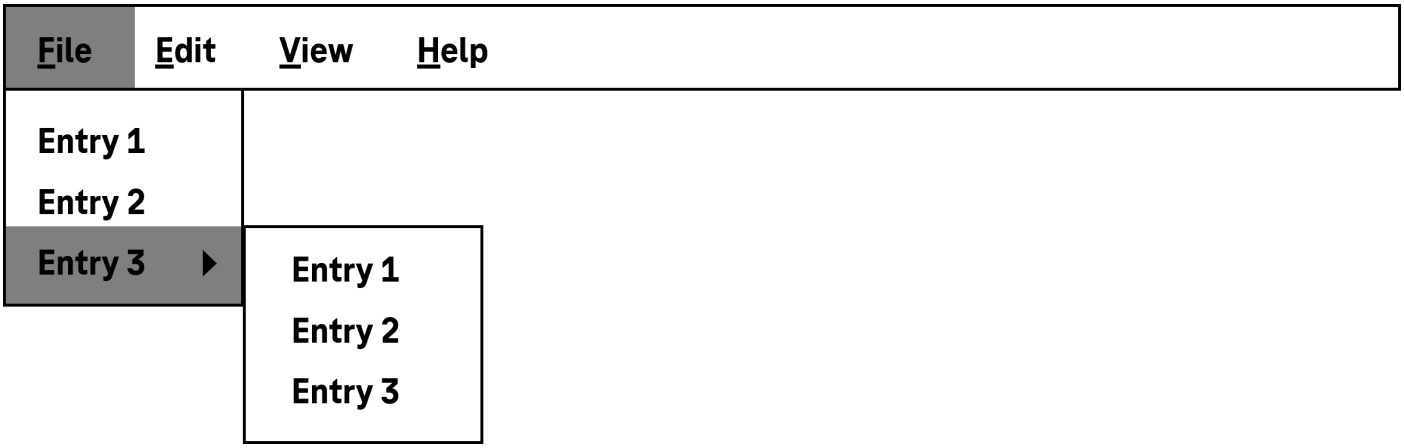

Figure 35: Application menu

Presentation

No.	Property	Description	Classification	Reference
529	Contrast	If the menu item has a text label, this must have a contrast ratio of at least 4.5:1 with respect to the background.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
530	Contrast	If the menu item has a graphic label, this must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
531	Contrast	The arrow icon which refers to a sub-menu must have a contrast ratio of at least 3:1 with respect to the neighboring color.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

No.	Property	Description	Classification	Reference
532	Contrast	If the selected menu item only differs from the unselected menu item due to its color (e.g. foreground or background color), the colors must have a contrast ratio of at least 3:1. Note: The selected menu item does not have to be color coded or marked with color only. It can be marked with a check box or a radio button, for example. In this case, the contrast requirements for the color coding are eliminated as long as the check box or the radio button has sufficient contrasts.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
533	Label	If the menu item has a graphic label, it should have a tooltip with a text label.	Should	WCAG 2.1: 3.3.5 (AAA); DIN EN ISO 9241-143: 9.6.11
534	Web: Consistency	If the menu is used for navigation, the menu items on each screen should be displayed in the same relative order and receive the keyboard focus (see Consistency).	Must	EN 301 549: 9.3.2.3
535	Desktop: Consistency	If the menu is used for navigation, the menu items on each screen should be displayed in the same relative order and receive the keyboard focus (see Consistency).	Should	WCAG 2.1: 3.2.3 (AA)
536	Focus visibility	If a menu item receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

No.	Property	Description	Classification	Reference
537	Resizing	The menu must be perceptible and operable with a font size adjustment of up to 400% (and a resulting display width of 320 px) (see Zoom). Note: The menu items can, for example, • wrap (i.e. they are shown in several successive rows), • be initialized with a menu button (e.g. with a hamburger icon (icon with three horizontal bars)), • be designed so they are scrollable horizontally (e.g. by scrollbar or two buttons).	Must	EN 301 549: 9.1.4.10, 11.1.4.10

Operation

No.	Property	Description	Classification	Reference
538	Use of the keyboard	It must be possible to access, operate and exit the menu with the keyboard (see Use of the keyboard table).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
539	Use of the keyboard	Frequently used functions should have a keyboard shortcut. Note 1: The keyboard shortcut should be displayed at the corresponding menu item. Note 2: The keyboard shortcuts should be documented in the Help option.	Should	DIN EN ISO 9241-171: 9.3.10; DIN EN ISO 9241-171: 9.3.11
540	Use of the keyboard	All the menu items should have a shortcut key . Note: The shortcut key should be marked by underlining the corresponding letter in the label.	Should	DIN EN ISO 9241-171: 9.3.10; DIN EN ISO 9241-171: 9.3.11
541	Updates	When focusing and operating the menu, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1
542	Click area	The click area of the menu items should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: menu

Action	Key	Classification
Focusing of the menu	Desktop:	Required

Action	Key	Classification
	• Application menu: ALT, F6 • Area menu: TAB Web: TAB	
Exiting the menu	Desktop: • Application menu: F6, ALT, selection of a menu item, ESC • Area menu: TAB Web: • TAB • possible selection of a menu item	Required
Opening a sub-menu of the menu	• Focus in the menu: UP/DOWN ARROW, ENTER • Focus in the application: ALT+ [shortcut key]	Required
Opening a sub-menu of a sub-menu	• RIGHT ARROW • ENTER	Required
Closing a sub-menu of the menu	ESC	Required
Closing a sub-menu of a sub-menu	• ESC • LEFT ARROW	Required
Navigating through the menu	RIGHT/LEFT ARROW Note: This must also work if the focus is in a sub-menu, and the use of the arrow key is not required in order to open or close a sub-menu due to the current focus position. In this case, the open sub-menu is closed automatically, and the next sub-menu is opened and the focus is placed on the first sub-menu item.	Required

Action	Key	Classification
Navigating through a sub-menu	UP/DOWN ARROW	Required
Navigating through the menu or sub-menu	[Shortcut key]	Recommended
Navigating through the menu (to a menu item before or afterwards with a defined increment)	PAGE UP, PAGE DOWN Note: The increment should match the amount of visible menu items.	Recommended
Selection of a menu item	ENTER, [shortcut key]	Required

Use of the pointing device: menu

Action	Key	Classification
Opening a sub-menu of the menu if no sub-menu is yet displayed	Left click on the parent menu item	Required
Opening a sub-menu of the menu if a sub-menu is already displayed	Hovering over the parent menu item	Required
Closing a sub-menu of the menu	Left click on the parent menu item, left click on a menu item in the sub-menu (which does not contain another submenu), click outside the menu	Required
Opening a sub-menu of a sub-menu	Hovering over the parent menu item	Required
Closing a sub-menu of a sub-menu	Hovering over another parent menu item, left click on a menu item in the sub-menu (which does not contain another sub-menu), click outside the menu	Required
Selection of a menu item	Left click on the menu item	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
543	Role	The menu role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
544	Role	The menu item, menu radio button or menu check box roles must be communicated to the Accessibility API for the menu items (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
545	Status	The status of the menu items must be communicated to the Accessibility API (see Element status). Note: This applies, for example, • to the “open” or “closed” status for menu items with a sub-menu. • to the “selected” or “unselected” status for menu radio buttons and menu check boxes.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
546	Orientation	The orientation of the menu (vertical or horizontal) must be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2
547	Name	If the menu has a label or description, these must be communicated as the Accessible Name and/or Accessible Description (see Label and Description). Note 1: If the page contains several menus, they must have a concise and expressive Accessible Name. Note 2: A sub-menu can be labeled as an Accessible Name by the parent menu item.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8

No.	Property	Description	Classification	Reference
548	Name	Each menu item must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5
549	Name	If a menu item has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5
550	Name	The group label of the menu items must (if available) be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
551	Operation	It must be possible to access, operate and exit the menu with assistive technology (see Accessibility API).	Must	EN 301 549: EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
552	Update	Updates concerning the Accessible Name or status of the menu items must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
553	Keyboard shortcut, shortcut key	If the menu item has a keyboard shortcut or a shortcut key , these must be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
554	Desktop: Position	The size and position of the menu and the menu items must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5, 11.5.2.13
555	Desktop: Element hierarchy	The parent / child relationships of the elements within the menu must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9

Menu button

Online betrachten

Synonyms: Switch with menu, list button

See also: Split button, drop-down list, menu, context menu, button

Menu buttons are for showing a menu (see DIN EN ISO 9241-161: 8.25).

A menu button consists of a text or graphic label and a visual indicator in order to identify the menu button as such (usually a border). The menu button also has a visual indicator which makes reference to the possibility for showing a menu (arrow icon).

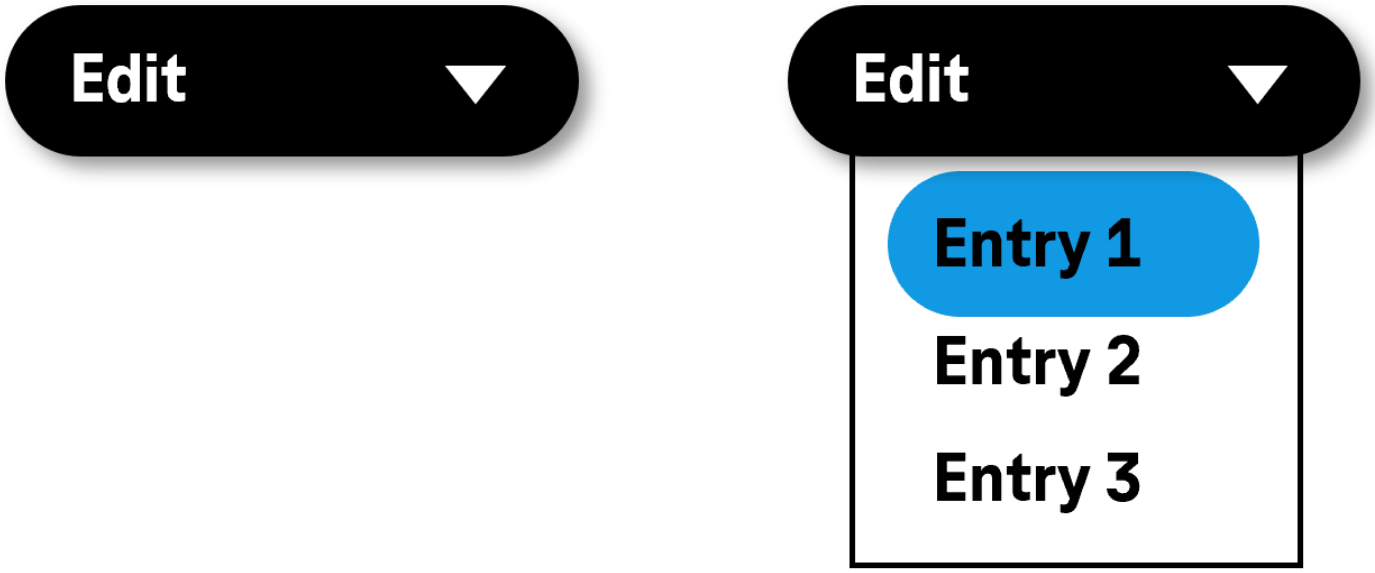

Figure 36: Menu button - closed left, open right

Presentation

The requirements concerning the button and the menu are described in the “button” and/or “menu” section. Here, only the additional requirements are described which result from the fact that a menu can be opened with the button. The requirements concerning the menu and the menu items that it contains only apply if the menu is open.

No.	Property	Description	Classification	Reference
556	Contrast	The arrow icon for opening and closing the menu must have a contrast ratio of at least 3:1 with respect to the neighboring color.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
557	Contrast	If the selected menu item only differs from the unselected menu item in the open state due to its color (e.g. foreground or background color), the colors must have a contrast ratio of at least 3:1. Note: The selected menu item does not have to be color coded or marked with color only. It can be marked with a check box, for example. In this case, the contrast requirements for the color coding are eliminated as long as the check box has sufficient contrasts.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

Operation

The requirements concerning the button and the menu are described in the “button” and/or “menu” section. Here, only the additional requirements are described which result from the fact that a menu can be opened with the button. The requirements concerning the menu and the menu items that it contains only apply if the menu is open.

Use of the keyboard: menu button

Action	Key	Classification
Opening the menu	SPACE ENTER ALT+DOWN ARROW	Required
Opening the menu	DOWN ARROW UP ARROW	Recommended
Closing the menu	ESC SPACE ENTER	Required

Use of the pointing device: menu button

Action	Key	Classification
Opening the menu	Left click on the menu button (label or arrow)	Required
Closing the menu	Left click on the menu button (label or arrow)	Required
Closing the menu	Left click on a menu item inside the open menu	Required
Closing the menu	Left click outside the menu button and the menu	Required

Programming/interfaces

The requirements concerning the button and the menu are described in the “button” and/or “menu” section. Here, only the additional requirements are described which result from the fact that a menu can be opened with the button. The requirements concerning the menu and the menu items that it contains only apply if the menu is open.

No.	Property	Description	Classification	Reference
558	Role	The menu button role must be communicated to the Accessibility API (see (Accessibility API)).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
559	Status	The status of the menu button must be communicated to the Accessibility API (see Element status). Note: This also applies to the “open” or “closed” status.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

Tab group

Online betrachten

Synonyms: Tab, tab panel

See also: Accordion, carousel

Tab groups are used to group and alternately display page areas (see DIN EN ISO 9241-161: 8.43).

A tab group consists of several tabs, the contents of which are displayed on an alternate basis. Each tab has an tab panel with a label which is displayed all the time. The individual tabs are selected using the tab panels. The tab panels are usually arranged horizontally above the tabs. Tab panels can contain interactive elements, such as buttons for removing the tab panel and the corresponding tab.

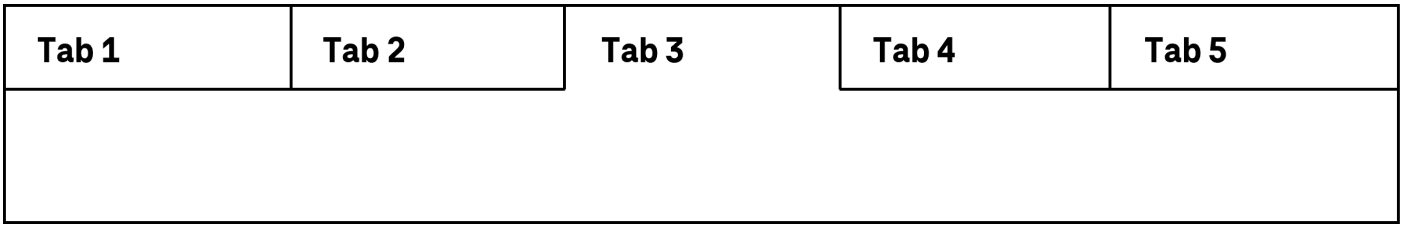

Figure 37: Tab group

Presentation

No.	Property	Description	Classification	Reference
560	Contrast	The label of the tab panel must have a contrast ratio of at least 4.5: 1 with respect to the background.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
561	Contrast	The marking of the selected tab panel must have a contrast ratio of at least 3:1 with respect to the background and/or with respect to the presentation of the unselected tab panel.	Must	EN 301 549: 9.1.4.1, 11.1.4.1, 9.1.4.11, 11.1.4.11

No.	Property	Description	Classification	Reference
562	Contrast	If the tab panels and the tabs are only identifiable as such on the basis of their color design, these colors must have a contrast ratio of at least 3:1 with respect to the neighboring colors. Note 1: A tab might be recognizable as an interactive element on the basis of its border or its background color, for example. Note 2: The requirement does not apply if the tab or the tab panel are clearly identifiable as such, because of their position, for example.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
563	Label	Each tab panel must have a visible label (see Label).	Must	EN 301 549 9.3.3.2, 11.3.3.2
564	Focus visibility	If a tab panel receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

No.	Property	Description	Classification	Reference
565	Resizing	All tab panels must be perceptible and operable with a font size adjustment of up to 400% (and a resulting display width of 320 px) (see Zoom). Note: The tab panels can, for example, • wrap (i.e. they are shown in several successive rows), • be initialized with a menu button , • be designed so they are scrollable horizontally (e.g. by scrollbar or two buttons).	Must	EN 301 549: 9.1.4.10, 11.1.4.10

Operation

No.	Property	Description	Classification	Reference
566	Use of the keyboard	It must be possible to access, operate and exit the tab panels with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
567	Use of the keyboard	If the tab panels contain other interactive elements, it must be possible to use these with the keyboard.	Must	EN 301 549: 9.2.1.2, 11.2.1.2
568	Use of the keyboard	The hidden tab fields and their contents must not receive keyboard focus.	Must	EN 301 549: 9.1.3.1, 11.1.3.1, 9.2.4.3, 11.2.4.3
569	Updates	When the tab panels are focused and navigated through, no change of context may occur. Note: This means that when focusing on the tab panel, the focus may not be automatically placed on the tab. It is considered permissible for the corresponding tab to be shown automatically when focusing on a tab panel, however.	Must	EN 301 549: 9.3.2.1, 11.3.2.1
570	Click area	The click area of the tab panels should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2
571	Click area	If the tab panels contain further interactive elements, their click area should total at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: tab group

Action	Key	Classification
Focusing of the selected tab panel	TAB	Required
Exiting the tab panel and navigating to the first element of the selected tab	TAB	Required
Navigating within the list of tab panels	Desktop: • UP/DOWN/RIGHT/LEFT ARROW • CTRL+TAB Web: UP/DOWN/RIGHT/LEFT ARROW	Required
Quick navigation to the first and/or last tab panel	POS1, END	Recommended
Switch between tabs (if the focus is in a tab)	CTRL+TAB	Recommended
Showing of the selected tab	ENTER, SPACE, or automatically when navigating through the list of tab panels	Required
Operation of interactive elements within the tab panel	The interactive elements within tab panels do not receive keyboard focus separately. The operation takes place • using either the context menu, which can be initialized with SHIFT+F10 and/or with the CONTEXT MENU, • or through a keyboard shortcut documented in the application and the Help option, such as DELETE in order to remove the tab panels and the corresponding tabs.	Required

Use of the pointing device: tab group

Action	Key	Classification
Showing a tab	Left click on the corresponding tab panel	Required
Enabling of interactive elements within the tab panels	Left click on the element	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
572	Role	The tab panel and tab role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
573	Role	If the tab panels contain further interactive elements, a note regarding their operability must be communicated to the Accessibility API (e.g. with the context menu or a keyboard shortcut).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
575	Desktop: Element hierarchy	The parent / child relationships of the tab panels within the tab group must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9
576	Status	The status of the tab panels and tabs must be communicated to the Accessibility API (see Element status). Note: This also applies to the tab panel status “selected” or “unselected” insofar as the corresponding tab is not automatically shown when navigating to a tab panel.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
577	Name	The tab panels must have a concise and expressive Accessible Name. Note 1: It is recommended that the tab is given the same Accessible Name as the corresponding tab panel. Note 2: If the tab panels and tabs have a parent label, this must also be communicated to the Accessibility API.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5
578	Name	If the tab panel has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
579	Operation	It must be possible to access, operate and exit the tab panels with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
580	Operation	If the tab panels contain other interactive elements, it must be possible to use these with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
581	Update	Updates concerning the Accessible Name or status of the tab panels must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
582	Desktop: Position	The size and position of the tab panels and tabs must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: tab groups in Web applications

Screen reader output for tab panel

- JAWS: [label of the tab bar] **group** [label of the tab panel] tab [| selected] [position] of [amount]

- NVDA: [label of the tab bar] **tab control** [label of the tab panel] tab [| selected] [position] of [amount]
- Windows Narrator: [label of the tab bar] **tab** [label of the tab panel] **tab item** [position] of [amount] [| selected]

Screen reader output for tabs

- JAWS: **tab panel start** [label]... **tab panel end**
- NVDA: -
- Windows Narrator: [label] **tab panel** ... **end** [label] tab panel

Note: With NVDA, the start and end of the tab is not perceptible. The tabs or content after this should be labeled as regions, so that which specific content is located inside and outside the tabs is perceptible with the screen reader.

HTML

There is no element for tab groups in HTML. Instead, the following can be used:

- Allocation of the information to different pages that are linked with each other,
- Accordion,
- Use of the corresponding ARIA roles.

ARIA

With tab groups, the following should be taken into account:

- The tab panel bar is marked with role=tablist, while the individual tab panels that are located inside the bar as child elements are marked with role=tab.
- The tab panel bar may not contain any elements apart from the tab panels.
- If the tab panels contain other control elements besides their labeling, it should be noted that these are imperceptible and not operable with the assistive technology. These elements should not be able to receive the keyboard focus. An alternative form of operation should be implemented and documented instead. If the tab panels contain a button for removing a tab, it is possible for the deletion to be enabled with the DELETE key.
- The tab panels are marked with role=tabpanel. The tabs are located in the source code immediately after the area which is marked with role=tablist. Due to this and the preceding reason, a tab group may not be used for accordions.
- The active tab panel, the tab of which is visible and which receives the focus with TAB, is marked with aria-selected=true. All of the other tab panels are marked with aria-selected=false. Reference can be made to the ID of the tab from the tab panel using aria-controls.
- The labeling of the tab panels should take place using text content. The bar of the tab panels and tabs can be labeled with aria-label or aria-labelledby.
- A non-standard vertical orientation of the tab bar can be specified with aria-orientation=vertical. Frequently, the orientation is not output by the assistive technology, so that with a vertically oriented tab bar, operation with all arrow keys should be possible.
- The tab bar should also be visually recognizable as such, so that sighted keyboard users are able to recognize the operation with the arrow keys.
- The tab group should contain at least two tabs.

- The tab panels can be marked as disabled with aria-disabled. It is recommended that disabled tab panels are left in the keyboard accessibility with the arrow keys.
- The selected tab (role=tabpanel) can be marked as tabindex=0 so that it receives the focus after the active tab panel, even if a specific tab is not a control element.
- The presentation of the tab group should be verified in Windows High Contrast mode. In this respect, the tab panels and tabs should have a border and the visual indicator for the active tab panel should not just be conveyed with color.
- The visible tab panels and the programmatically focused elements should have the same position and size.

Further information: tab role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link) , tablist role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link) , tabpanel role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link) , Tabs Pattern | APG | WAI | W3C

Input field (one-line)

Online betrachten

Synonyms: Text field, edit, input, input field, text box, one-line plain text edit control

See also: Input field (multi-line), password input field, spin button, input field with autocomplete function, combined input field

Input fields allow for the inputting and editing of characters (numbers, letters, special characters) (see DIN EN ISO 9241-161: 8.12).

An input field is usually a rectangular area of the page that contrasts with its surroundings due to a border, a line or a different background color, for example. The input field can contain further control elements which are related to the value input.



Figure 38: One-line input field with a label on the left

Presentation

No.	Property	Description	Classification	Reference
583	Contrast	The text in the input field must have a contrast of at least 4.5:1.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
584	Contrast	The border or line of the input field must have a contrast ratio of at least 3:1 with respect to the background of the page or the input field. Alternatively, the contrast ratio between the background color of the page and the input field must be at least 3:1.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

No.	Property	Description	Classification	Reference
585	Label	The input field must have a visible label (see Label) Note: The labeling can also take place implicitly, e.g. • in the case of input fields in tables, using the column and row headings, • in the case of a search field, with the search button located behind it.	Must	EN 301 549 9.3.3.2, 11.3.3.2
586	Label	The label should not be located in the input field, so that it is visible at all times and cannot be confused with a value.	Should	DIN EN ISO 9241-125: 5.1.14
587	Focus visibility	If the input field receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7
588	Focus visibility	The default text cursor must be displayed in the input field (see Text cursor).	Must	EN 301 549: 9.2.4.7, 11.2.4.7, 11.5.2.13
589	Value	If the input field may only contain certain characters or a particular input format is required and this is not apparent from the field label, it is necessary for an input instruction in the description or label to explain the permissible characters or the required input formats.	Must	EN 301 549: 9.3.3.2, 11.3.3.2

No.	Property	Description	Classification	Reference
590	Value	The input field should be long enough to display the maximum number of characters without scrolling. Note 1: For long text inputs, a multi-line input field should be used. Note 2: This does not apply to font size adjustments of up to 400%.	Should	DIN EN ISO 9241-143: 6.2.8

Operation

No.	Property	Description	Classification	Reference
591	Use of the keyboard	It must be possible to access, operate and exit the input field with the keyboard (see Use of the keyboard table, below).	Must	9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
592	Use of the keyboard	If the input field contains other interactive elements such as buttons, these must receive the focus with the tab key after the input field. Note: This does not apply to interactive elements for which a generally known use of the keyboard is implemented, such as a button for deleting the inputs in the input field. Nor does this apply to elements for which a keyboard shortcut was implemented, insofar as this keyboard shortcut is visually perceptible at the input field and with assistive technology.	Must	EN 301 549: 9.2.2.1, 11.2.1.1
593	Updates	When focusing and operating the input field, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1, 9.3.2.2, 11.3.2.2
594	Click area	The click area of the input field should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2
595	Click area	It should be possible to place the focus on the input field by clicking on the input field or by clicking on the label.	Should	

Use of the keyboard: input field

Action	Key	Classification
Focusing of the input field	TAB	Required
Exiting the input field	TAB	Required
Entering a value	[Inputting of printable characters] Note: If certain characters cannot be entered, an appropriate implicit or explicit input instruction must be available at the input field. An implicit input instruction might, for example, be the “phone number” label which indicates that no letters can be entered.	Required
Navigating in the input field	RIGHT/LEFT ARROW POS1, END	Required
Deleting text in the input field	DELETE, BACKSPACE	Required
Highlighting text in the input field	SHIFT+[any navigation key], CTRL+A	Required
Removing the highlighting	[any navigation key]	Required
Pasting of text from the clipboard	CTRL+V	Required
Copying or cutting highlighted text	CTRL+C, CTRL+X	Required

Use of the pointing device: input field

Action	Key	Classification
Placing the focus on a specific position in the input field	Left click in the input field	Required
Placing the focus in the input field	Left click on the label	Recommended
Highlighting text in the input field	Drag with the left mouse button pressed	Required
Highlighting a word in the input field	Double click	Recommended
Highlighting all content in the input field	Triple click	Recommended

Programming/interfaces

No.	Property	Description	Classification	Reference
596	Role	The input field role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
597	Role	If the technology used is able identify the input purpose of form fields, the purpose of the form fields must be marked for the data of the respective users (such as surname, date of birth, place of residence) according to https://www.w3.org/TR/WCAG21/#input-purposes" .	Must	EN 301 549: 9.1.3.5, 11.1.3.5.1
598	Value	The value of the input field must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7
599	Status	The status of the input field must be communicated to the Accessibility API (see Element status).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
600	Name	The input field must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
601	Name	If the input field has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
602	Operation	It must be possible to access, operate and exit the input field with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
603	Update	Updates concerning the Accessible Name, value or status of the input field must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
604	Desktop: Position	The size and position of the input field must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.5, 11.5.2.13
605	Desktop: Position	The position of the text cursor in the input field must be communicated to the Accessibility API (see Focus indicator)	Must	EN 301 549: 11.5.2.13

Practical tip: one-line input field in Web applications

Screen reader output

- JAWS: [label] **edit** [value] [instruction on text input]
- NVDA: [label] **edit** [value]
- Windows Narrator: [label] **edit** [value]

HTML

The input field should be implemented with the HTML element <input type=text>. For an input field which is used for entering search terms, <input type=search> can be used. The screen readers output the search field as a normal input field, i.e. the label must indicate the purpose which is the search. The initial value is communicated through the value attribute. The label of the input field should be linked to the input field

with the <label for=ID> element. The maximum and minimum required length of the value can be determined with maxlength and minlength. Neither of the values are perceptible visually or with the use of assistive technology. If they are used, a corresponding instruction should be provided at the input field. If a particular input format is required, instead of <input type=text>, the following elements can be used:

- <input type=tel> for telephone numbers,
- <input type=email> for email addresses,
- <input type=url> for Internet addresses,
- <input type=text pattern=...> or inputs that must correspond to the regular expression in the pattern attribute,
- <input type=number> for numbers (see Spin button),
- <input type=date>, <input type=time> etc. for date and time information (see Date picker).

These input fields with a predefined format are automatically validated by the browser. The error messages of the browsers, however, are not accessible, and should therefore be replaced or supplemented with specific error messages from the application (see also: Browser validation of the required required attribute in Practical tip: programmatic label of required fields in Web applications.

The input fields for telephone numbers, email and Internet addresses as well as input fields with the pattern attribute are not identifiable either visually or with assistive technology – they are displayed as normal input fields and/or output by the screen reader. They must therefore be labeled in an expressive way and should also have operating instructions for the required input format.

The inputmode attribute can be used to specify which virtual keyboard should be displayed on mobile devices because corresponding details are expected (e.g. inputting of numbers, text, URLs, email addresses). The use of inputmode does not result in an automatic browser validation. Insofar as the keyboard that is used does not result from the type attribute of the input field, the use of inputmode is recommended, as assistive technology is also able to use this information appropriately, to display corresponding virtual keyboards, for example. It should be noted that the inputmode attribute replaces unnecessary input instructions, however.
The placeholder attribute can be used for a placeholder. However, it is recommended to display input instructions next to the field and to link them to the input field using aria-describedby. The placeholder attribute has the following disadvantages:

- The contrasts are often insufficient (in respect to either the background or the text in the input field, which makes it difficult to distinguish between the placeholder and the value).
- The placeholder is not displayed as soon as the field contains a value.

An input field can be marked as disabled (disabled) and read-only (readonly).

An input field can be marked as a required field with required.

Further information: 4.10.5 The input element - HTML Standard (whatwg.org)

ARIA

If the input field is not implemented with the HTML element, it is also necessary to take account of the following:

- The role is communicated with role=textbox.

- For an input field which is used for entering search terms, role=searchbox can be used. The screen readers output the search field as a normal input field, i.e. the label must indicate the purpose which is the search.
- The labeling of the input field can take place with aria-label or aria-labelledby.
- The value of the input field is derived from the text content of the element which is marked with role=textbox.
- The input field can be marked as a required field with aria-required, as disabled with aria-disabled, and as read-only with aria-readonly.
- For a placeholder, aria-placeholder is used.
- The presentation of the input field should be verified in Windows High Contrast mode.
- The visible input field and the programmatically focused element should have the same position and size.

Further information: textbox role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)

Input field (multi-line)

Online betrachten

Synonyms: Text field, text area, multi-line text input, multiline plain text edit control

See also: Input field (one-line), Rich Text Editor

Multi-line input fields allow for the input of long text contents (see DIN EN ISO 9241-161: 8.45).

A multi-line input field is usually a rectangular area of the page that contrasts with the background due to a border, for example. The border can have a control point for the scaling of the input field. A multi-line input field can have scrollbars. The requirements of the control point and scrollbars are described in the appropriate sections.

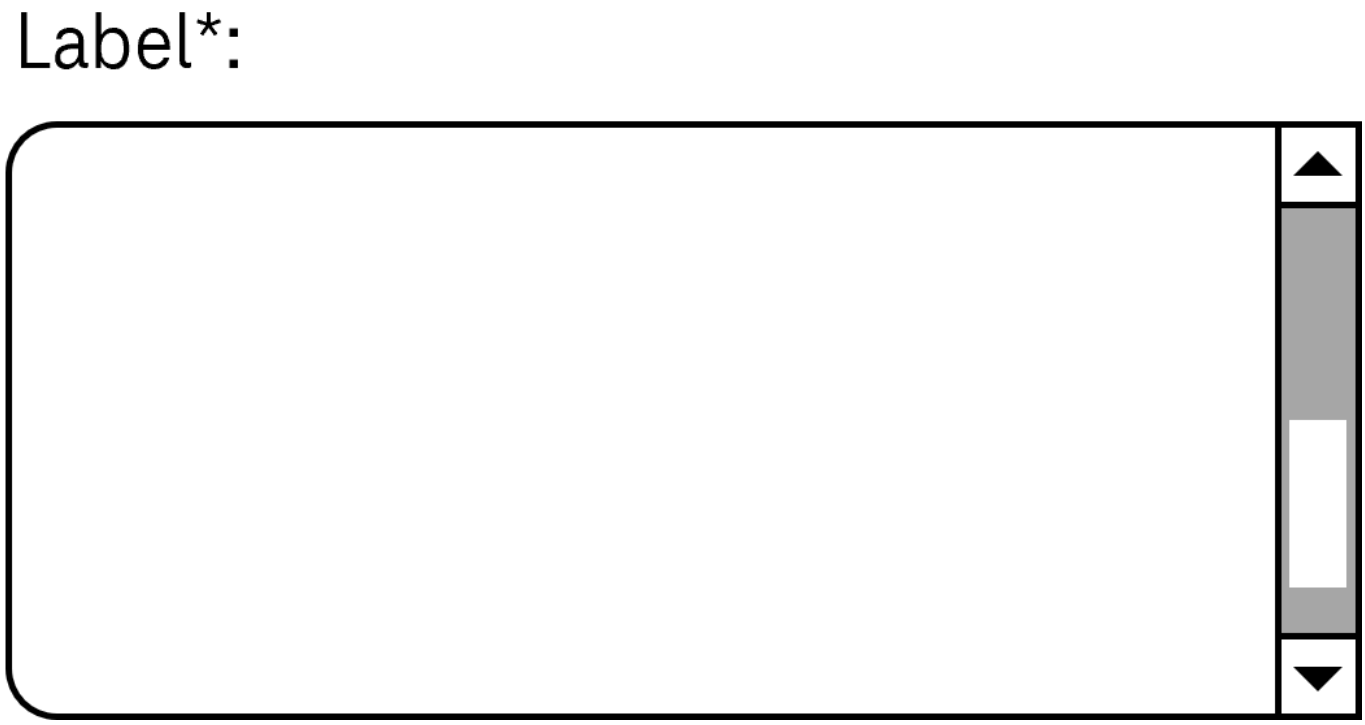

Figure 39: Multi-line input field with top label

Presentation

The requirements concerning the input field are described in the “input field” section. Here, only the additional requirements are described which result from the fact that it is an input field with several lines.

No.	Property	Description	Classification	Reference
606	Value	If the display width exceeds 320 px, the multi-line input field must be displayed in such a way that its text content can be read without having to scroll horizontally.	Must	EN 301 549: 9.1.4.10, 11.1.4.10

Operation

The requirements concerning the input field are described in the “input field” section. Here, only the additional requirements are described which result from the fact that it is an input field with several lines.

No.	Property	Description	Classification	Reference
607	Use of the keyboard	If the multi-line input field cannot be exited with the TAB key (because it means TAB steps are inserted in the text), a keyboard shortcut for exiting the field must be implemented and documented in the application.	Must	EN 301 549: 9.2.1.2, 11.2.1.2

Use of the keyboard: multi-line input field

Action	Key	Classification
Exiting the multi-line input field	TAB or a documented keyboard shortcut	Required
Entering a value	[Inputting of printable characters and ENETER and/or TAB] Note: If certain characters cannot be entered, an appropriate input instruction must be available at the input field.	Required
Navigating in the multi-line input field	UP/DOWN/RIGHT/LEFT ARROW	Required
Quick navigation in the multi-line input field	POS1, END, PAGE UP, PAGE DOWN	Required

Programming/interfaces

The requirements concerning the input field are described in the “input field” section. Here, only the additional requirements are described which result from the fact that it is an input field with several lines.

No.	Property	Description	Classification	Reference
608	Role	The multi-line input field role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
609	Operation	It must be possible to access, operate and exit the multi-line input field with assistive technology (see Accessibility API). Note: If it is not possible to exit the multi-line input field with TAB, in order to exit, the keyboard shortcut must be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
610	Update	Updates concerning the Accessible Name, value or status of the multi-line input field must be communicated to the Accessibility API (see Accessibility API). Note: This applies to an updated description of the amount of allowed characters that have been entered or are remaining, for example.	Must	9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17

Practical tip: multi-line input field in Web applications

Screen reader output

- JAWS: [label] **input field** [instruction on contained value] [instruction on text input]
- NVDA: [label] **multi-line input field** [contents of the current line]
- Windows Narrator: [label] **edit** [value]

Please note:

- With JAWS, the value of the input field is imperceptible when focusing with TAB. The content is only output line by line when navigating through the field.
- With NVDA, the value of the input field is only partially perceptible when focusing with TAB, because only the current line is output. The content is only output line by line when navigating through the field.
- If the input field is empty, with JAWS and Windows Narrator, the difference between a one-line and multi-line input field is imperceptible.

- If the value of the input field contains empty lines and the text cursor is in an empty line, when focusing with TAB with NVDA, it is imperceptible that the input field contains a value because “empty” is output only.

HTML

Note: Multi-line input fields should only be used if very long inputs are expected or if the inputting of paragraph breaks (ENTER key) must be enabled. Otherwise, a one-line input field should be used, because the accessibility of the multi-line input fields is poorer for users of screen readers.

The input field should be implemented with the HTML element <textarea>.
The initial value is communicated through the text content of the element.
The label of the input field should be linked to the input field with the <label for=ID> element.
The maximum and minimum required length of the value can be determined with maxlength and minlength. Neither of the values are perceptible visually or with the use of assistive technology. If they are used, a corresponding instruction should be provided at the input field.
The visual size of the input field can be set using the cols and rows attributes. It is recommended that the size is defined using CSS instead, so that the multi-line input field can be responsively designed to avoid horizontal scrolling at a screen width of 320 px. The initial size of the input field should be sufficiently large, as keyboard users are not able to change the size of the field – the control point automatically displayed by the browser can only be operated with a pointing device.
The placeholder attribute can be used for a placeholder. However, it is recommended to display input instructions next to the field and to link them to the input field using aria-describedby. The placeholder attribute has the following disadvantages:

- The contrasts are often insufficient (in respect to either the background or the text in the input field, which makes it difficult to distinguish between the placeholder and the value).
- The placeholder is not displayed as soon as the field contains a value.

An input field can be marked as disabled (disabled) and read-only (readonly). An input field can be marked as a required field with required.

Further information: 4.10.11 The textarea element - HTML Standard (whatwg.org)

ARIA

If the input field is not implemented with the HTML element, it is also necessary to take account of the following:

- The role is communicated with role=textbox and the ARIA attribute aria-multiline=true.
- The labeling of the input field can take place with aria-label or aria-labelledby.
- The value of the input field is derived from the text content of the element which is marked with role=textbox.
- The input field can be marked as a required field with aria-required, as disabled with aria-disabled, and as read-only with aria-readonly.
- For a placeholder, aria-placeholder is used.
- The presentation of the input field should be verified in Windows High Contrast mode.
- The visible input field and the programmatically focused element should have the same position and size.

Further information: textbox role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)

Password input field

Online betrachten

Synonyms: Password field

See also: Input field (one-line), authentication

Password input fields allow for the entry of a password.

A password input field is usually a rectangular area of the page that contrasts with its surroundings due to a border, a line or a different background color, for example. The entered value is only displayed in a masked format. A password input field can contain a button to unmask the value.

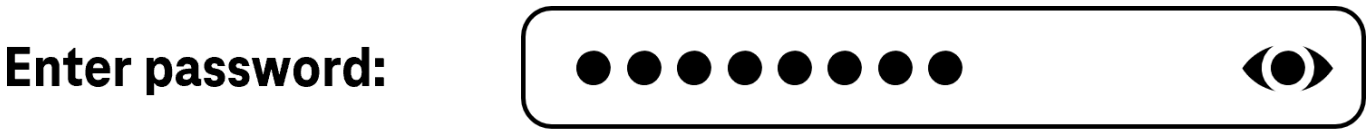

Figure 40: Password input field

Presentation

The requirements concerning the input field are described in the “input field” section. Here, only the additional requirements are described which result from the fact that it is an input field for entering a password.

No.	Property	Description	Classification	Reference
611	Contrast	The masked and unmasked text in the password input field must have a contrast of at least 4.5:1.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
612	Value	If the password input field may only contain certain characters or a particular input format is required, it is necessary for this to be explained in the input instruction. Note: This only applies when a new password is given, and not when the password is entered for authentication purposes or if the password is entered repeatedly.	Must	EN 301 549: 9.3.3.2, 11.3.3.2

Operation

The requirements concerning the input field are described in the “input field” section. Here, only the additional requirements are described which result from the fact that it is an input field for entering a password.

No.	Property	Description	Classification	Reference
613	Use of the keyboard	If the password input field contains other interactive elements such as a button for unmasking the input, these must receive the focus with the tab key after the input field. Note: This does not apply to elements for which a Tastaturkürzel has been implemented, insofar as this keyboard shortcut is visually perceptible at the password input field and with assistive technology.	Must	EN 301 549: 9.2.1.1, 11.2.1.1

Programming/interfaces

The requirements concerning the input field are described in the “input field” section. Here, only the additional requirements are described which result from the fact that it is an input field for entering a password.

No.	Property	Description	Classification	Reference
614	Role	The password input field role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
615	Value	The value of the password input field may not be communicated to the Accessibility API unless the masking of the entry was removed. Instead of this, a masked character string must be communicated as a value to the Accessibility API (see Accessibility API). Note: The masked character string to be used for the Accessibility API is a character string consisting of an amount of “black dot” (•) points corresponding to the length of the input.	Must	EN 301 549: 4.2.11, 9.4.1.2, 11.4.1.2, 11.5.2.7
616	Status	The status of the password input field must be communicated to the Accessibility API (see Element status). Note: This also applies to the “masked” or “unmasked” status. Accordingly, the password input field can be communicated without masking to the Accessibility API with the input field role.	Must	EN 301 549: 9.4.2.1, 11.4.2.1

Practical tip: password input field in Web applications

Screenreader output

- JAWS: [label] **input field for passwords** [masked value] [instruction on text input]
- NVDA: [label] **protected input field** [masked value]
- Windows Narrator: [label] **edit** [value]

Please note:

- With Windows Narrator, the difference between an input field and a password input field is not perceptible.
- When entered, the masked value, the “black dot” (•) is output as “asterisk” (JAWS, NVDA) and/or “hidden” (Windows Narrator). A value that has already been entered is output as a sequence of “bullet points”, whereby JAWS shortens the amount of characters to three by default.

HTML

To prevent the entered password from being output by the assistive technology, the password input field must be implemented with the HTML element <input type=password>.
The label of the password input field should be linked to the password input field with the <label for=ID> element.
A password input field can be marked as disabled (disabled) and read-only (readonly). A password input field can be marked as a required field with required.
If specific rules apply when assigning the password, the following should be observed:

- The rules should be explained at the password input field.
- If the rules are short and unstructured, the password input field should be programmatically linked with the rules (with aria-describedby, for example).
- If the rules are long or structured, they should be placed in the source code before the password input field, to ensure that the reading sequence is correct. The rules can be given a heading (e.g. “Instructions on password entry”). At the input field, the Accessible Description should make reference to the fact that rules are available (e.g. “Note the input instructions for passwords before this field”).
- If the rules are validated automatically upon input, this should also be perceptible when using assistive technology, e.g. by marking the result of the validation as a live region.

Further information: 4.10.5.1.6 Password state (type=password) - HTML Standard (whatwg.org)

ARIA

There is no role for password input fields in ARIA. Instead, the appropriate HTML element must be used for password input fields so that the password to be entered is not output by the assistive technology.

Input field with autocomplete function

Online betrachten

Synonyms: Input field with suggestion list, autocomplete search, auto suggestion box

See also: selection list, drop-down list, menu button, input field (one-line),, combined input field

Input fields with the autocomplete function allow for text to be entered freely and the selection of options from a list, whereby the list is only opened after the text has been input.

In the closed state, an input field with autocomplete function consists of a single input field. In the opened state, a selection list is displayed underneath (possibly with a scrollbar). The options of the selection list can be grouped. The labeling of the groups cannot be selected.

Input fields with the autocomplete function can also add the first hit to the input field automatically, so that this can be adopted immediately (Inline-Autocomplete). In this case, the text cursor remains at the position of the last character entered.

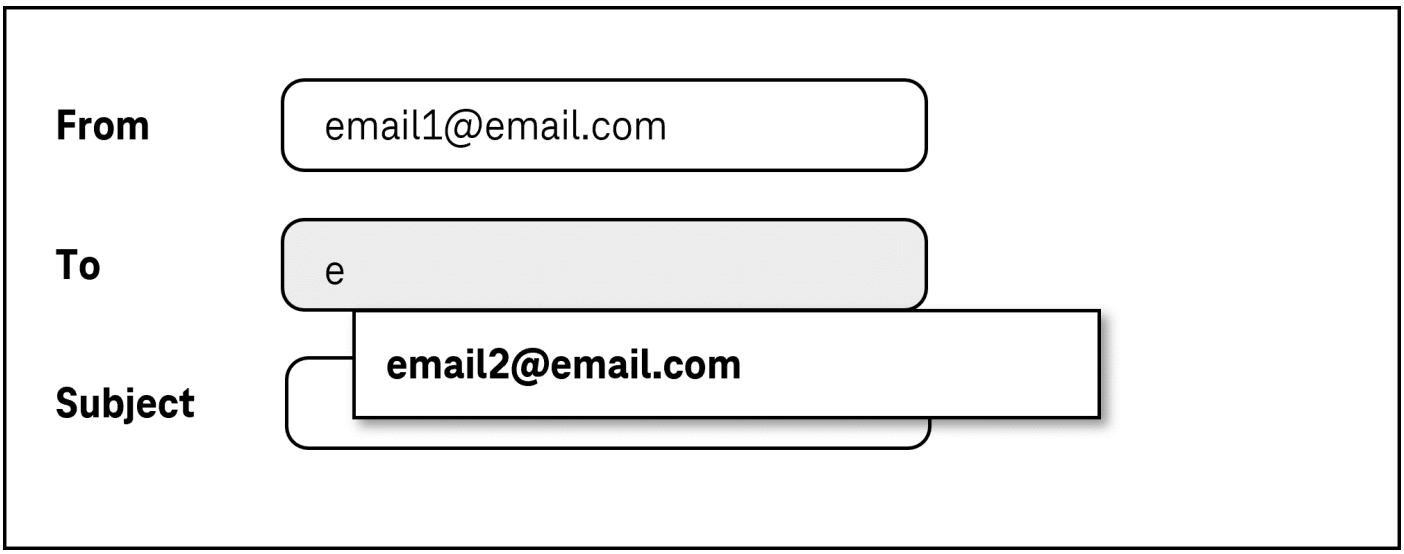

Figure 41: Input field with autocomplete function

Presentation

The requirements concerning the input field are described in the “input field” section. Here, only the additional requirements are described which result from the fact that it is an input field with autocomplete function.

No.	Property	Description	Classification	Reference
617	Contrast	The labeling of the options in the selection list of the autocomplete function must have a contrast of at least 4.5:1.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
618	Contrast	The selected option in the selection list of the autocomplete function must have a contrast of at least 3:1 with respect to the neighboring options.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
619	Focus visibility	In the case of keyboard navigation through the list entries, the current option must be displayed in the visible area.	Must	EN 301 549: 11.2.4.7

Operation

The requirements concerning the input field are described in the “input field” section. Here, only the additional requirements are described which result from the fact that it is an input field with autocomplete function.

No.	Property	Description	Classification	Reference
620	Use of the keyboard	It must be possible to access, operate and exit the options in the selection list of the autocomplete function with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
621	Updates	When focusing and operating the input field with the autocomplete function, no unexpected change of context may occur.	Must	EN 301 549: 11.3.2.1 und 11.3.2.2
622	Click area	The click area of the list entries in the selection list should be at least 24 x 24 px.	Should	WCAG 2.2: 2.5.8 (AA)

Use of the keyboard: input field with autocomplete function

Action	Key	Classification
Opening of the selection list	Text input	Required
Closing the selection list	ENTER ESC TAB Text input that does not lead to any results from the autocomplete function	Reauired
Operation of the selection list (selection of the previous or following value)	UPP ARROW, DOWN ARROW	Required
Operation of the selection list (selection of a value before or after with a defined increment)	PAGE UP, PAGE DOWN Note: The increment should correspond to the amount of visible options.	Recommended

Use of the pointing device: input field with autocomplete function

Action	Key	Classification
Closing the selection list	Left click on a value within the open list	Required
Closing the selection list	Left click outside the combined input field (consisting of the input field and the selection list)	Required
Value selection within the selection list	Left click on a value	Required

Programming/interfaces

The requirements concerning the input field are described in the “input field” section. Here, only the additional requirements are described which result from the fact that it is an input field with autocomplete function.

No.	Property	Description	Classification	Reference
623	Role	The input field role must be communicated to the Accessibility API (see Accessibility API)	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
		Note: If the technology used does not know the input field with autocomplete function role, the input field role must be used. In this case, the description should make reference to the autocomplete function.		
624	Desktop: Element hierarchy	The parent / child relationships of the elements within the list must be communicated to the Accessibility API.	Must	EN 301 549: 911.5.2.9
624	Status	The status of the selection list of the autocomplete function must be communicated to the Accessibility API (see Elementstatus). Note: This applies to the “open” or “closed” status, for example.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5, 1.5.2.9
625	Name	The group label of the list entries must (if available) be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
626	Operation	It must be possible to access, operate and exit the selection list of the autocomplete function with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
627	Update	Updates concerning the Accessible Name, value or status of the combined input field must be communicated to the Accessibility API (see Accessibility API). Note: This applies in particular to the displaying of the selection list of the autocomplete function, as otherwise it is not perceptible that values are available for selection.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15

No.	Property	Description	Classification	Reference
628	Desktop: Position	The size and position of the selected elements in the selection list must be communicated to the Accessibility API (see Focus indicator))	Must	EN 301 549: 11.5.2.13

Spin button

Online betrachten

Synonyms: Rotary button, step forward button, spinner, spin control

See also: Input field, slider, drop-down list, combined input field, button

Spin buttons allow for the selection of a value from a range of values with continuous data (e.g., days of the week, years). A spin button consists of two buttons that allow for the selection of the previous value and the following value as well as an input field in order to display the value. The input field for displaying the value can be read-only or allow for the direct text input. The value range of the spin button can be limited or unlimited, i.e. it can have a minimum value, a maximum value or both (see DIN EN ISO 9241-161: 8.41).



Figure 42: Spin button

Presentation

No.	Property	Description	Classification	Reference
629	Contrast	The text in the spin button must have a contrast ratio of at least 4.5: 1 with respect to the background.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
630	Contrast	The border of the spin button must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
631	Contrast	The arrow icons for the selection of the previous and following value must have a contrast ratio of at least 3:1 with respect to the neighboring color.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
632	Label	The spin button must have a visible label (see label).	Must	EN 301 549 9.3.3.2, 11.3.3.2
633	Fokus visibility	If the spin button receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7
634	Focus visibility	If it is possible to enter text in the spin button, the default text cursor must be displayed (see Textcursor).	Must	EN 301 549: 9.2.4.7, 11.2.4.7, 11.5.2.13

Operation

No.	Property	Description	Classification	Reference
635	Use of the keyboard	It must be possible to access, operate and exit the spin button with the keyboard (see Use of the keyboard table, below). Note: The button for the selection of the previous and following value should not receive the keyboard focus separately	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
636	Updates	When focusing and operating the spin button, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1 und 9.3.2.2, 11.3.2.2
637	Klickbereich	If it is not possible for a value to be entered, the click area of the buttons for the selection of the previous and following values should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: spin button

Action	Key	Classification
Focusing of the spin button	TAB	Required
Leaving the spin button	TAB	Required
Entering a value in the input field	Text input	Required (if text input is possible)
Navigating in the input field	RIGHT/LEFT ARROW POS1, END	Required (if text input is possible)
Operation of the spin button (selection of the previous or following value)	UP ARROW, DOWN ARROW	Required
Operation of the spin button (selection of a value before or after with a defined increment)	PAGE UP, PAGE DOWN Note: The increment depends on the amount of possible values. An increment of 10 for instance, would make sense with 100 values.	Recommended
Operation of the spin button (selection of the first and last value)	POS1, END	Recommended (if no text input is possible and a range of values exists)

Use of the pointing device: spin button

Action	Key	Classification
Placing the focus on a specific position in the input field	Left click in the input field	Required (if text input is possible)
Selection of the previous or following value	Left click on the corresponding button with the arrow icon	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
638	Role	The spin button role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
639	Value	The value of the spin button must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7
640	Desktop: Value range	If the spin button has a minimum and maximum value, it must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.7
641	Status	The status of the spin button must be communicated to the Accessibility API (see Elementstatus). Note: This also applies to the "text input possible" or "text input not possible" status.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
642	Name	The spin button must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
643	Name	If the spin button has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
644	Operation	It must be possible to access, operate and exit the spin button with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
645	Update	Updates concerning the Accessible Name, value or status of the spin button must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15

No.	Property	Description	Classification	Reference
646	Desktop: Position	The size and position of the spin button must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5
647	Desktop: Position	If the input field is not read-only, the position of the text cursor in the input field must be communicated to the Accessibility API (see Focus visibility)	Must	EN 301 549: 11.5.2.13

Practical tip: spin button in Web applications

Screen reader output

- JAWS: [label] **edit spin box** [value] [instruction on text input and operation with the arrow keys]
- NVDA: [label] **spin button** | spin button editable [value]
- Windows Narrator: [label] **spinner** | **spin button** [value] minimum [minimum value] and maximum [maximum value]

Please note:

- The Windows Narrator outputs the HTML spin buttons as a “net” and the ARIA spin buttons as a “spin button”.
- The minimum and maximum values are only perceptible with the Windows Narrator.
- The difference between spin buttons with and without text input is only perceptible with NVDA.

HTML

The spin button should be implemented with the HTML element <input type=number> The initial value is communicated through the value attribute
The increment as well as the minimum and maximum values can be communicated with the step, min und max attributes.
The label of the spin button should be linked to the spin button with the <label for=ID> element.
A spin button can be marked as disabled (disabled) and read-only (readonly) ausgezeichnet werden. A spin button can be marked as a required field with required.

Further information: 4.10.5.1.12 Number state (type=number) - HTML Standard (whatwg.org)

ARIA

Please note: On mobile devices, an ARIA spin button without text input may not be operated with assistive technology, because no gesture commands for the operation of non-native spin buttons have been implemented. This applies, in particular, if the buttons for increasing and/or decreasing the value are child elements of the spin button. In addition to this, the differences between spin buttons with and without text input are not perceptible with most screen readers. It is therefore recommended that only spin buttons with a text input option are implemented.

If the spin button is not implemented with the HTML element, it is also necessary to take account of the following:

- The role is communicated with role=spinbutton.
- If the spinbutton role is located at an input field or at an element which contains an input field, it is a spin button with text input. Otherwise, it is a spin button without text input.
- The labeling of the spin button can take place with aria-label or aria-labelledby. If the ARIA spinbutton role is located at an HTML input field, it can also be labeled with <label for=ID>.
- The current value must be specified with aria-valuenow. BefindetIf the ARIAspinbutton role is located at an HTML input field, its value is used as the current value of the spin button.
- aria-valuetext can also be used to specify a value in text form which should be then output by the assistive technology instead of the value inaria-valuenow.
- The minimum and maximum values can be specified with aria-valuemin and aria-valuemax.
- With ARIA spin buttons, specifying an increment is not possible.
- The spin button can be marked as disabled with aria-disabled and with aria-readonly as read-only.
- The switches used for increasing and/or decreasing the value should not receive keyboard focus separately.
- The presentation of the spin button should be verified in Windows High Contrast mode.
- The visible spin button and the programmatically focused element should have the same position and size.

Further information: spinbutton role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, Spinbutton Pattern | APG | WAI | W3C

Selection list

Online betrachten

Synonyms: List field, multi-line selection list, list box

See also: Multiple selection list, drop-down list, radio button group, tree structure

Selection lists allow the selection of an option from a list (see DIN EN ISO 9241-161: 8.39).

All the available options are displayed in the selection list (with scrollbars)if necessary). The current value is highlighted. The options can be grouped. The labeling of the groups cannot be selected. The focused option is identical to the selected option.

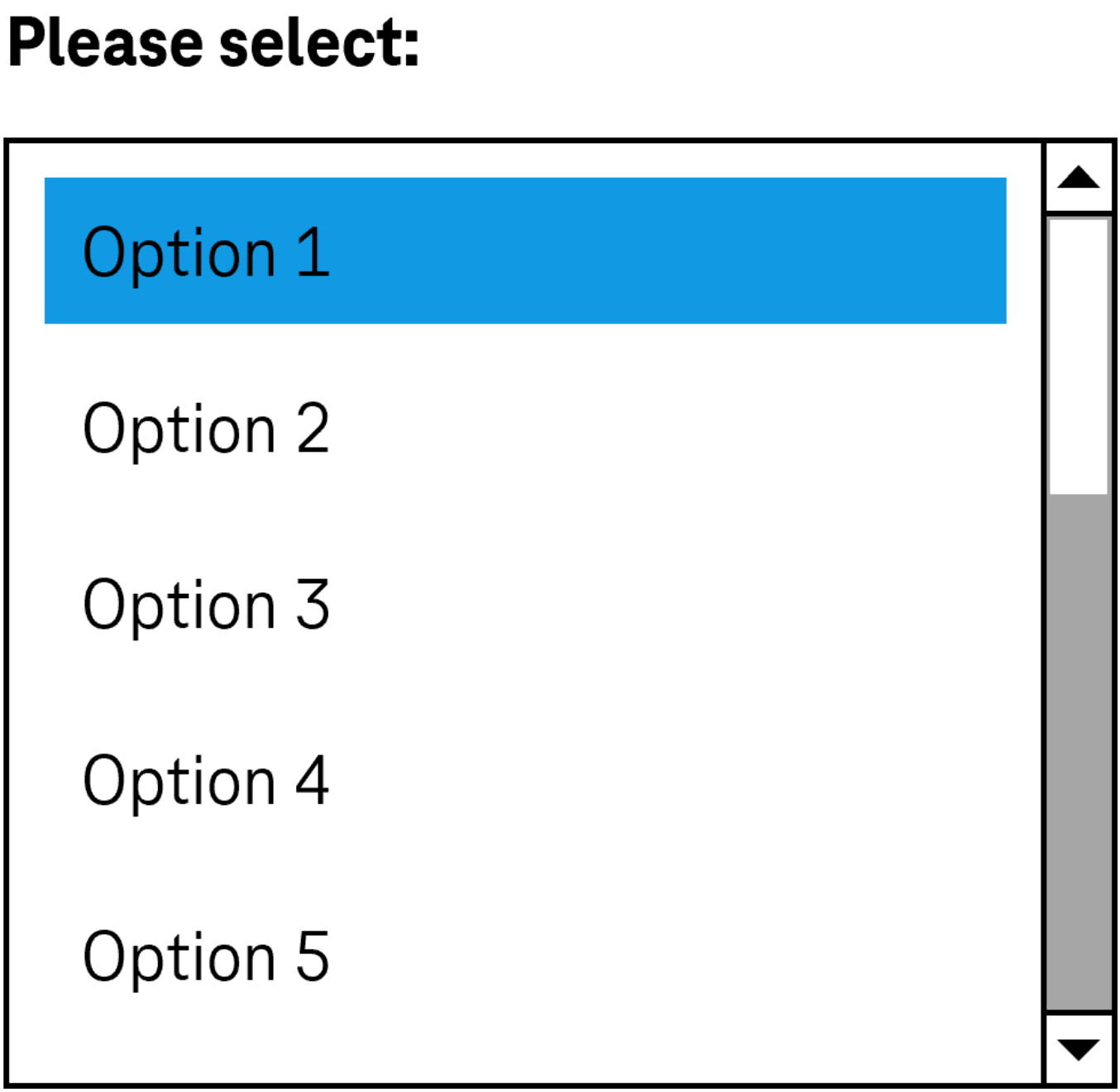

Figure 43: Selection list

Presentation

No.	Property	Description	Classification	Reference
648	Contrast	The labeling of the options in the selection list must have a contrast of at least 4.5:1. Note 1: This applies to both the selected and unselected options. Note 2: If the options are labeled with graphics rather than text, the contrast of the graphics with respect to the background and the content-bearing areas of the graphics with respect to each other must be at least 3:1.	Must	EN 301 549: 9.1.4.3, 11.1.4.3, 9.1.4.11, 11.1.4.11
649	Contrast	If the selected option only differs from the unselected option due to its color (e.g. foreground or background color), a contrast ratio between the colors of at least 3:1 must be complied with. Note: The selected list entry does not have to be color coded or marked with color only. It can be marked with a check box, for example. In this case, the contrast requirements for the color coding are eliminated as long as the check box has sufficient contrasts.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
650	Contrast	The border of the selection list must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
651	Label	The selection list must have a visible label (see Label).	Must	EN 301 549 9.3.3.1, 11.3.3.2

No.	Property	Description	Classification	Reference
652	Focus visibility	If the selection list receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7
653	Focus visibility	When navigating through the options, the current option must be displayed in the visible area and as focused.	Must	EN 301 549: 9.2.4.7, 11.2.4.7
654	Options list	It should not be necessary to scroll the list with the options horizontally, i.e. it should be at least as wide as the longest entry.	Should	DIN EN ISO 9241-143: 9.3.4
655	Options list	Die Optionen sollen so formuliert werden, dass die relevante, zur Unterscheidung dienende Information am Anfang steht.	Should	DIN EN ISO 9241-143: 9.3.4

Operation

No.	Property	Description	Classification	Reference
656	Use of the keyboard	DIt must be possible to access, operate and exit the selection list with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.25
657	Updates	When focusing and operating the selection list, no unexpected change of context may occur. Note: In particular, the change in value may not result in a loss of focus or the opening of a new screen.	Must	EN 301 549: 9.3.2.1, 11.3.2.1, 9.3.2.2, 11.3.2.2
658	Click area	The click area of the list entries in the selection list should be at least 24 x 24 px.	Should	WCAG 2.2

Use of the keyboard: selection list

Action	Key	Classification
Focusing of the selection list	TAB	Required
Exiting the selection list	TAB	Required
Operation of the selection list (selection of the previous or following value)	UP ARROW, DOWN ARROW	Required
Operation of the selection list (selection of the first and last value)	POS1, END	Required
Operation of the selection list (selection of a value before or after with a defined increment)	PAGE UP, PAGE DOWN Hinweis: Note: The increment should correspond to the amount of visible options.	Required with several list entries
Operation of the selection list (selection of a value which starts with a specific character string)	Entry of one or more characters (within a short time) Note: If two entries start with the same character string, the entries are navigated to one another in sequence.	Required with several list entries

Zeigeinstrumentbedienung Auswahlliste

Action	Key	Classification
Value selection within the selection list	Left click on a value	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
659	Role	The selection list role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
660	Value	The value of the selection list must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7

No.	Property	Description	Classification	Reference
661	Desktop: Element hierarchy	The parent / child relationships of the elements within the selection list must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9
662	Status	The status of the selection list must be communicated to the Accessibility API (see Elementstatus).	Must	EN 301 549: 69.4.1.2, 11.4.1.2, 11.5.2.5
663	Name	The selection list must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
664	Name	If the selection list has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
665	Name	The group label of the list entries must (if available) be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
666	Operation	It must be possible to access, operate and exit the selection list with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
667	Update	Updates concerning the Accessible Name, value or status of the selection list must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
668	Position	The size and position of the selection list must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15

No.	Property	Description	Classification	Reference
669	Desktop: Position	The size and position of the selected elements in the selection list must be communicated to the Accessibility API (see Focus visibility)	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: selection list in Web applications

Screen reader output

During focusing of the selection list:

- JAWS:
  - Without selected list entry: [label] **list box** [instruction on operation with the arrow keys]
  - With selected list entry: [label] **list box** [value] [position] of [amount] [instruction on operation with the arrow keys]
- NVDA:
  - Without selected list entry: [label] **list**
  - With selected list entry: [label] **list** [value] [position] of [amount]
- Windows narrator:
  - Without selected list entry: [label] selected **requires selection** contains [amount] items
  - With selected list entry: [label] list [value] [position] of [amount] selected

When navigating through the selection list with the arrow keys:

- JAWS: [value] [position] of [amount]
- NVDA: [value] [position] of [amount]
- Windows Narrator: [value] [position] of [amount] selected

On reading with the virtual cursor:

- JAWS:
  - Without selected list entry: [label] **list box**
  - With selected list entry: [label] [value] **list box items** selected [position] of [amount]
- NVDA:
  - Without selected list entry: [label] **list** clickable [first list entry]
  - With selected list entry: [label] **list** clickable [value]
- Windows Narrator: [label] [position] of [amount] [| selected] [list entry]

Please note:

- JAWS and NVDA do not output the grouping of an HTML selection list with <optgroup>. Windows Narrator only outputs the grouping with <optgroup> on reading with the virtual cursor.
- JAWS and NVDA do not output the grouping of an ARIA selection list correctly with role=group group. Windows Narrator only outputs the grouping with role=group on reading with the virtual cursor.
- When reading with the virtual cursor, JAWS only outputs the selected list entry, NVDA outputs the selected or first list entry, and Windows Narrator outputs all list entries.

HTML

The selection list should be implemented with the HTML elements <select> and <option> with a value greater than 1 with the size attribut und without the multiple attribute.
The list entries can be grouped with the <optgroup> element. The grouping should be avoided, however, as many screen readers do not output the group label (which is specified with the label attribut).
The initially selected list entry can be set with the selected attribut. An initial list entry should be selected in each selection list, as keyboard users automatically make a selection when navigating through the selection list which cannot be undone. The list entry should be initially selected which is either most likely to be selected or which contains a neutral option (e.g. “not applicable”).
The label should be linked to the selection list with the <label for=ID> element.
A selection list, a group of list entries and the individual list entries can be marked as disabled (disabled), but not read-only (readonly).
A selection list can be marked as a required field with required.

Further information: 4.10.7 The select element - HTML Standard (whatwg.org)

ARIA

If the selection list is not implemented with the HTML element, it is also necessary to take account of the following:

- The selection list is marked with role=listbox and contains the list entries that are marked with role=option.
- The list entries can be grouped within an element which is marked with role=group. The group is labeled with aria-label or aria-labelledby.
- The selected list entry is marked with aria-selected=true, all others with aria-selected=false. The selected list entry can also be communicated with aria-checked – this is not recommended for selection lists without multiple selection, however.
- Alternatively, the selected list entry is not explicitly marked, but is determined automatically by the focused list entry during the operation. This is not recommended, though, because the assistive technology is not able to reliably determine the selected list entry in the non-focused state.
- The labeling of the selection list can take place with aria-label or aria-labelledby.
- When navigating through the list entries in the selection list, these must either actually receive the focus, or reference is made to the selected list entry with aria-activedescendant. The first variant is to be preferred.
- The selection list can be marked as disabled with aria-disabled.
- The selection list can be marked as read-only with aria-readonly.
- The selection list can be marked as a required field with aria-required.
- A non-standard horizontal orientation of the selection list can be specified with aria-orientation=horizontal. Frequently, the orientation is not output by the assistive technology, so that with horizontally oriented list entries, operation with all the arrow keys should be possible.
- The presentation of the selection list should be verified in Windows High Contrast mode.
- The visible selection list and the programmatically focused element should have the same position and size.
- When navigating through the list entries, the focused list entry should be displayed in the visible area.

Further information: listbox role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)

, Listbox Pattern | APG | WAI | W3C

Multiple selection list

Online betrachten

Synonyme: List field, multi-line selection list, list box

See also: Selection list, drpo-down list, check box group, tree structur

Multiple selection lists allow the selection of several options from a list (see DIN EN ISO 9241-161: 8.39).

All the selectable options are displayed in the multiple selection list (with scrollbrars if necessary). The selected options are highlighted. The options can be grouped. The labeling of the groups cannot be selected. The focused option is not automatically identical to the selected option.

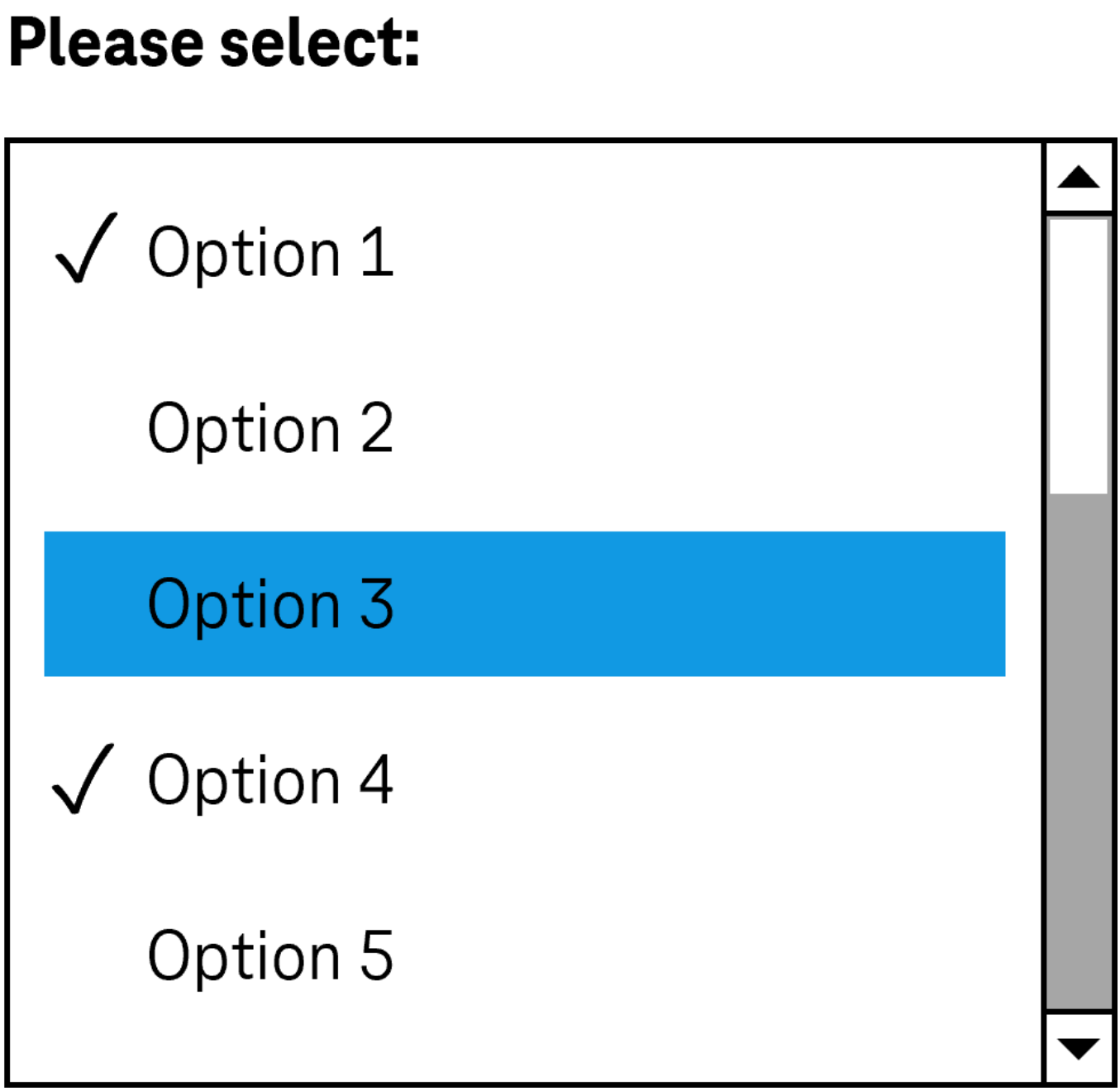

Figure 44: Multiple selection list (the check mark indicates selected options, the blue background color indicates the currently focused option)

Presentation

The requirements concerning the selection list are described in the “selection list” section. Here, only the additional requirements are described which result from the fact that it is a selection list with multiple selection.

No.	Property	Description	Classification	Reference
670	Label	The multiple selection list must have a visible label (see Label). Note: Users should be made aware of the possibility for multiple selection.	Must	EN 301 549 9.3.3.2, 11.3.3.2
671	Description	If the multiple selection is not possible by simply activating the list entries, the operation with the keyboard and pointing device should be explained (see Practical tip: simplified use of multiple selection).	Should	WCAG 2.1: 3.3.5 (AAA), DIN EN ISO 9241-143: 9.6.11

Operation

The requirements concerning the selection list are described in the “selection list” section. Here, only the additional requirements are described which result from the fact that it is a selection list with multiple selection.

No.	Property	Description	Classification	Reference
672	Use of the keyboard	It must be possible to access, operate and exit the multiple selection list with the keyboard (see Use of the keyboard table, below). Note: This applies to the selection of neighboring and non-neighboring list entries.	Must	EN 301 549: 9.2.1.2, 11.2.1.2

Use of the keyboard: multiple selection list

Note: A simplified but currently less established use of the multiple selection feature is described in the practical tip on the simplified use of multiple selection. It is recommended that the implemented operation concerning multiple selection is described in the application or the Help option, regardless of the method chosen

Action	Key	Classification
Focusing of the multiple selection list	TAB	Required
Value selection (all other values are deselected)	[Navigation key]	Required
Selection of neighboring values	SHIFT+[Navigation key]	Required
Selection of non-neighboring values	CTRL+[Navigation key], gefolgt von CTRL+SPACE	Required
Deselect a selected value	CTRL+SPACE	Required
Select all values	CTRL+A	Required

Use of the pointing device: multiple selection list

Note: A simplified but currently less established use of the multiple selection feature is described in the practical tip: simplified use of multiple selection. It is recommended that the implemented operation concerning multiple selection is described in the application or the Help option, regardless of the method chosen.

Action	Key	Classification
Value selection (all other values are deselected)	Left click on a valuet	Required
Selection of neighboring values	SHIFT+Left click	Required
Selection of non-neighboring values	CTRL+Left click	Required
Deselect a selected value	CTRL+Left click	Required

Programming/interfaces

The requirements concerning the selection list are described in the “selection list” section. Here, only the additional requirements are described which result from the fact that it is a selection list with multiple selection.

No.	Property	Description	Classification	Reference
673	Role	The multiple selection list role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.511.4.1.2, 11.5.2.5
674	Operation	It must be possible to access, operate and exit the multiple selection list with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17.5.2.17

Practical tip: alternatives to the selection list with multiple selection

The possibility for multiple selection is not generally identifiable, as the multiple selection lists do not differ visually from the selection list. Moreover, multiple selection (in particular the selection of non-neighboring values) with the keyboard and the pointing device is difficult. In addition, the selected elements are often imperceptible with the screen reader. It is therefore recommended to replace multiple selection lists with other elements, to implement them in a simple form, or to describe their function and operation. Alternatives to multiple selection lists include:

- group of check boxes, or
- two selection list or multiple selection lists side by side, with the first list showing the available values and the second list showing the selected values. Buttons can be used to move values from the first list to the second list or to remove them (see screenshot).

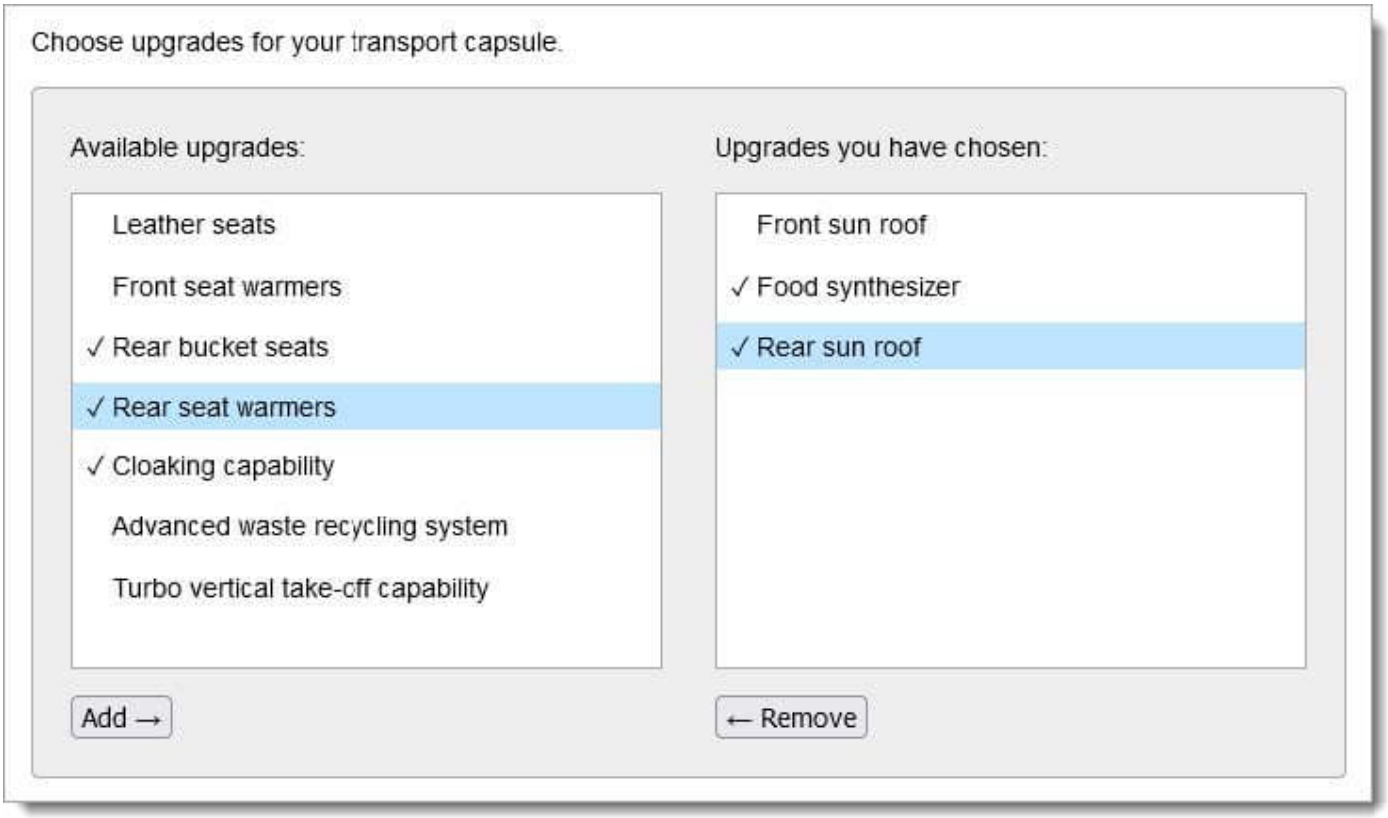

Figure 45: Multiple selection list (WAI-ARIA Authoring Practices 1.2: Listbox > Example Listboxes with Rearrangeable Options)

A simple use of multiple selection lists can be implemented as follows:

- with the pointing device: Click on a value changes the status (selected or unselected)
- with the keyboard: Enabling of a value with the SPACE bar changes the status (selected or unselected).

Practical tip: simplified use of multiple selection

A simple use of multiple selection lists can be implemented as follows:

- with the pointing device: Click on a value changes the status (selected or unselected)
- with the keyboard: Enabling of a value with the SPACE bar changes the status (selected or unselected).

Practical tip: multiple selection list in Web applications

Screen reader output

- JAWS:
    - JAWS:
        - Without selected list entry: [label] **ExtendedSelect ListBox** [instruction on operation with the arrow keys]
        - With selected list entries: [label] **ExtendedSelect ListBox ** [focused list entry] [position] of [amount] [instruction on operation with the arrow keys]
- NVDA:
    - Without selected list entry: [label] **list**
    - With selected list entries: [label] **list** [focused list entry] [position and amount]
- Windows narrator:
    - Without selected list entry: [label] **selected supports multiple selection** contains [amount] items
    - With selected list entries: [label] list [focused list entry] [position and amount] selected.

Please note: In the form mode, the selected list entries are imperceptible with the screen reader. The possibility for multiple selection is partly imperceptible. The use of an alternative element is therefore recommended (see Practical tip: alternatives to the selection list with multiple selection).

HTML

Please note: The use of the keyboard is also difficult for the following reasons:

- Modifier keys, which users may not be familiar with, are required for multiple selection. With old browsers, multiple selection with the keyboard is not possible and/or it is necessary for different modifier keys to be used.
- In the case of multiple selection, the selected list entries are visible, but the currently focused list entry is not.

The use of an alternative element is therefore recommended (see Practical tip: alternatives to the selection list with multiple selection).

The implementation instructions for the HTML selection list are described in the “selection list” section. Here, only the additional instructions are given, which result from the fact that it is a selection list with multiple selection.

- The multiple selection list should be implemented with the multiple attribute.
- The initially selected list entry is marked with the selected attribute.

ARIA

The implementation instructions for the ARIA selection list are described in the “selection list” section. Here, only the additional instructions are given, which result from the fact that it is a selection list with multiple selection.

- The possibility for multiple selection is communicated with aria-multiselectable=true.
- The selected list entries are marked with aria-selected=true or aria-checked=true, all others with aria-selected=false and/or aria-checked=false.

- When navigating through the list entries in the selection list, these must either actually receive the focus, or reference is made to the focused list entry with aria-activedescendant. The first variant is to be preferred. In both cases, care must be taken to ensure that the navigation through the multiple selection list does not automatically change the status of the list entries (selected or unselected)

Tree structure

Online betrachten

Synonyms: Hierarchical list, tree, tree view, structure view

See also: selection list, multiple selection list, hierarchical table, menu

Tree structures enable the presentation and operation of hierarchically structured lists (see DIN EN ISO 9241-161: 8.17). The nested lists can be shown and hidden. A button with an indicator on the list entries shows whether the subordinate list is shown or hidden. Tree structures can be used for different purposes, for example:

- the selection of one or more list entries within a form (as is the case with a selection list or multiple selection list),
- navigation (as is the case with a Link list or a menu),
- the showing of data (as is the case with a list).

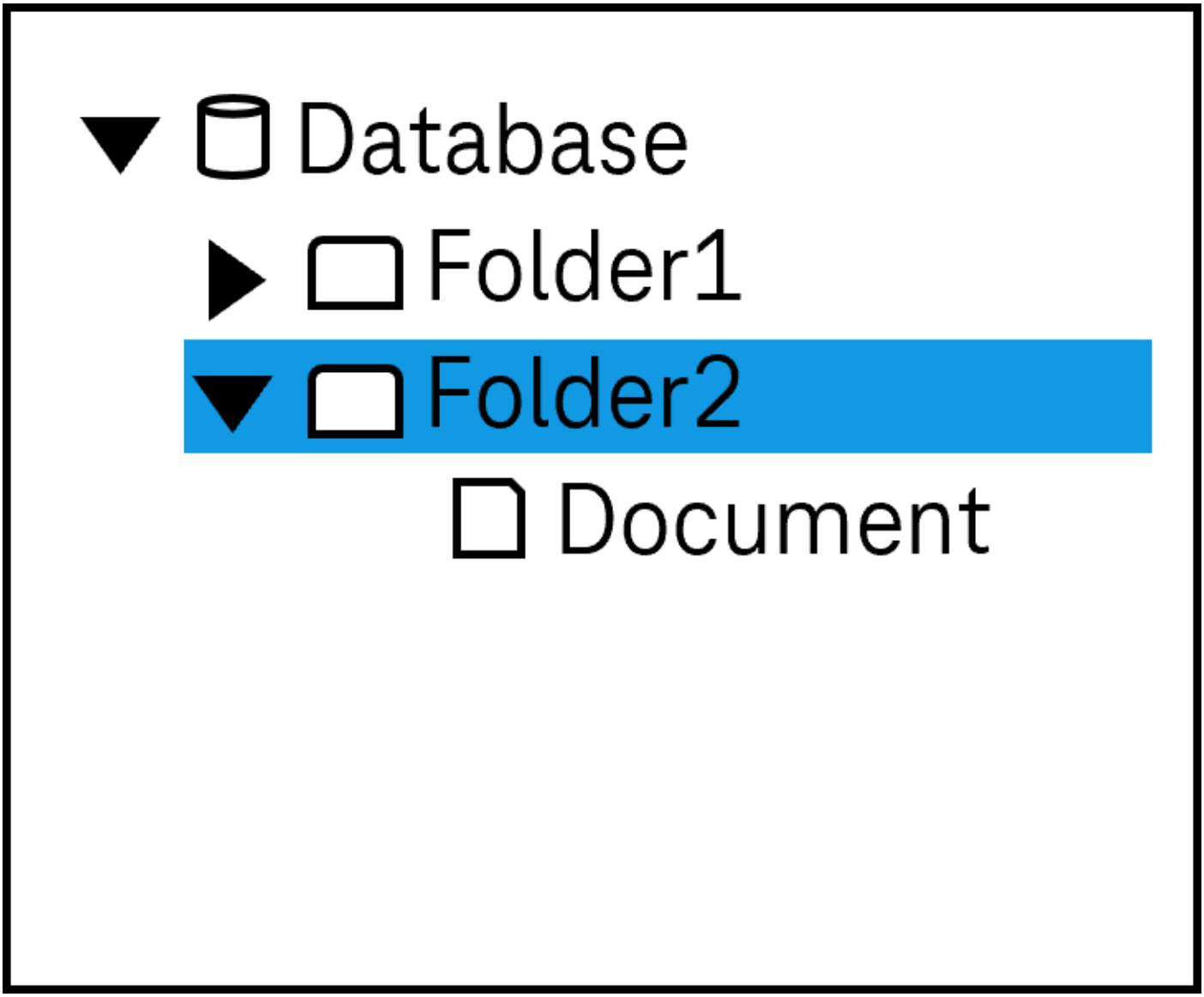

Figure 46: Tree structure

Presentation

No.	Property	Description	Classification	Reference
675	Contrast	The labeling of the list entries in the tree structure must have a contrast of at least 4.5:1. Note 1: This applies to both the selected and unselected entries. Note 2: If the list entries are labeled with graphics rather than text, the contrast of the graphics with respect to the background and the content-bearing areas of the graphics with respect to each other must be at least 3:1.	Must	EN 301 549: 9.1.4.3, 11.1.4.3, 9.1.4.11, 11.1.4.11
676	Contrast	If the selected list entry only differs from the unselected list entry due to its color (e.g. foreground or background color), the colors must have a contrast ratio of at least 3:1. Note: The selected list entry does not have to be color coded or marked with color only. It can be marked with a check box, for example. In this case, the contrast requirements for the color coding are eliminated as long as the check box has sufficient contrasts.	Must	EN 301 549: 9.1.4.1, 11.1.4.1, 9.1.4.11, 11.1.4.11
677	Contrast	Icons that display the status of list entries (shown or hidden) must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
678	Label	If it serves as a form element, the tree structure must have a visible label (see Label).	Must	EN 301 549 9.3.3.2, 11.3.3.2

No.	Property	Description	Classification	Reference
679	Focus visibility	If the tree structure receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7
680	Focus visibility	When navigating through the list entries, the current list entry must be displayed in the visible area.	Must	EN 301 549: 9.2.4.7, 11.2.4.7
681	Option list	It should not be necessary to scroll the tree structure horizontally, i.e. it should be at least as wide as the longest entry.	Must	EN 301 549: 9.2.1.1, 11.2.1.1
682	Option list	The list entries should be formulated in such a way that the information that serves the purpose of differentiation is at the beginning.	Should	DIN EN ISO 9241-143: 9.3.4

Operation

No.	Property	Description	Classification	Reference
683	Use of the keyboard	It must be possible to access, operate and exit the tree structure with the keyboard (see Use of the keyboard table, below). Note: The buttons for showing and hiding the subordinate list entries should not receive the keyboard focus separately.	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
684	Update	When focusing and operating the tree structure, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1, 9.3.2.2, 11.3.2.
685	Click area	The elements for showing and hiding subordinate lists should be at least 24 x 24 px in size.	Should	WCAG 2.2
686	Click area	The click area of the list entries in the tree structure should be at least 24 x 24 px.	Should	WCAG 2.2

Use of the keyboard: tree structure

Action	Key	Classification
Focusing of the tree structure	TAB	Required
Exiting the tree structure	TAB	Required
Navigating to the previous or following value	UP ARROW, DOWN ARROW	Required
Quick navigation to the first and last value	POS1, END	Required
Quick navigation to a value before or after with a defined increment	PAGE UP, PAGE DOWN Note: The increment should match the amount of visible entries in the list.	Recommended
Navigating to the first entry in the nested list	RIGHT ARROW	Required
Show subordinate list	RIGHT ARROW	Required
Hide subordinate list	LEFT ARROW	Required
Select value, enabling list entry	SPACE, ENTER	Required

Note 1: In the case of tree structures with multiple selection, moreover, the use of the keyboard for the multiple selection should be implemented as described for the multiple selection list.
Note 2: The use of the keyboard for tree structures may differ from the standard described here depending on the programming language or framework to be used (e.g. showing and hiding of subordinate lists with PLUS and MINUS). The alternate use of the keyboard should be described in the application and Help option.

Use of the pointing device: tree structure

Action	Key	Classification
Select value, enabling list entry	Left click	Required
Show and hide subordinate list	Left click on the show and hide icon	Required
Show and hide subordinate list	Double-click on list entry	Recommended

Note: In the case of tree structures with multiple selection, moreover, the use of the pointing device for the multiple selection should be implemented as described for the multiple selection list.

Programming/interfaces

No.	Property	Description	Classification	Reference
687	Role	The tree structure role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
688	Value	The values of the tree structure must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7
689	Desktop: Element hierarchy	The parent / child relationships of the elements within the tree structure must be communicated to the Accessibility API. Note: If this is not possible, the hierarchy of the list entries must be communicated to the Accessibility API in another way, e.g. in text form	Must	EN 301 549: 11.5.2.9
690	Status	The status of the tree structure and the list entries must be communicated to the Accessibility API (see Element status). Note: With the list entries, this also applies to the “open” or “closed” status.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
691	Name	The tree structure must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.
692	Name	If the tree structure has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
693	Operation	It must be possible to access, operate and exit the tree structure with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
694	Update	Updates concerning the Accessible Name, value or status of the tree structure must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
695	Position	The size and position of the tree structure and its list entries must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: tree structures in Web applications

Screen reader output

During focusing of the tree structure:

- JAWS: [label] **tree view** [list entry] [| open | closed] [instruction on navigation and operation with the arrow keys]
- NVDA: [label] **tree** [list entry] [position] of [amount] level [number] [| expanded | collapsed] Heading level [number]
- Windows Narrator: [label] **tree** [list entry] [position] of [amount] level [number] [| expanded | collapsed] Heading level [number]

When navigating through the tree structure with the arrow keys:

- JAWS:
  - When changing the level: Level [number] [list entry] [| open | closed] [position] of [amount]
  - Within the same level: [list entry] [| open | closed]
- NVDA:
  - When changing the level: Level [number] [list entry] [| expanded | collapsed] [position] of [amount]
  - Within the same level: [list entry] [| expanded | collapsed] [position] of [amount] level [number]
- Windows Narrator: [list entry] [position] of [amount] level [number] [| expanded | coollapsed] Heading level [number]

On reading with the virtual cursor:

- JAWS:
  - without selected element: [label] **tree view**

    - with selected element: [label] [list entry] **tree view item** selected [| expanded | collapsed] [position] of [amount]]
- NVDA: [label] **tree view** [| expanded | collapsed] level [number] [list entry]
- Windows Narrator: [label] [position] of [amount] level [number] [| expanded | collapsed] heading level [number] [list entry]

Please note:

- When reading with the virtual cursor, JAWS only outputs the selected elements (i.e., elements marked with tabindex=0 or aria-selected=true, for example).
- When reading with the virtual cursor, NVDA outputs the first list entry and all the visible nested list entries of the first list entry, but not the selected list entries.
- Windows Narrator outputs all the visible list entries when reading with the virtual cursor.
- Windows Narrator incorrectly outputs “heading level [number]” for each list entry due to the implicit and/or explicit aria level.

HTML

There is no element for tree structures in HTML. Instead, the following can be used:

- nested lists (containing the elements <ul>, <li>) with buttons for showing and hiding subordinate list entries which receive focus with the TAB key, or
- Use of the corresponding ARIA roles.

ARIA

With tree structures, the following should be taken into account:

- The tree structure is marked with role=tree and contains the list entries that are marked with role=treeitem.
- The nested list entries are grouped within an element which is marked with role=group. The group itself is located within the superordinate list entry, which is marked with role=treeitem.
- The status of the list entries (open or closed) is communicated with aria-expanded.
- The current list entry is marked with aria-selected=true, all others with aria-selected=false.
- List entries can also be marked as enabled and/or not enabled with aria-checked as long as the tree structure offers this functionality (e.g. visually, with check boxes at the list entries). If aria-checked allows for a multiple selection of list entries, this is communicated with aria-multiselectable=true.
- The labeling of the tree structure can take place with aria-label or aria-labelledby. The labeling of the list entries should take place using text content. The groups are not labeled.
- When navigating through the list entries in the tree structure, these must either actually receive the focus, or reference is made to the selected list entry with aria-activedescendant. The first variant is to be preferred.
- The tree structure may not contain any elements apart from the groups and list entries.
- The tree structure can be marked as a required field with aria-required and as disabled with aria-disabled.
- The presentation of the tree structure should be verified in Windows High Contrast mode. In this respect, the icons that communicate the status of the list entries should be easy to see, for example.
- The visible list entries and the programmatically focused elements should have the same position and size.

Further Information: tree role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link) , treeitem role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link) , Tree View Pattern | APG | WAI | W3C

Control point

Online betrachten

Synonyms: Drag point, handle

See also: Section separator

A control point is used for the spatial editing of an element such as a graphic, a text block or an application window (see DIN EN ISO 9241-161: 8.16). The control point can, for instance, be used to scale, rotate, distort, or move the element.

A control point often consists of a small circle or square at the corners of the editable element. The editing options can also be shown on the control point. Control points are often only displayed when the corresponding element has focus. With application windows, the control point can also be invisible. A control point can have a context menu for further functions.

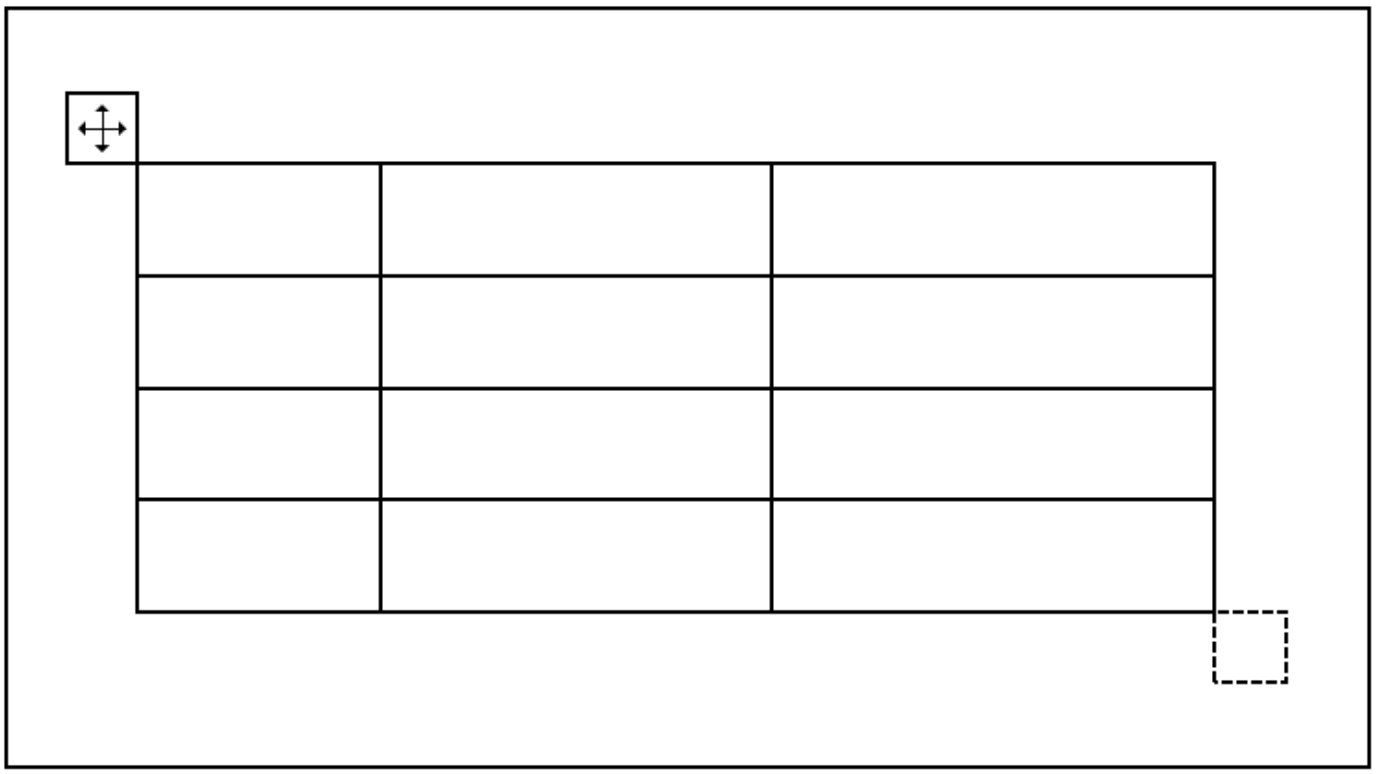

Figure 47: Control point

Presentation

No.	Property	Description	Classification	Reference
696	Contrast	The control point must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
697	Focus visibility	If the control point receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
698	Use of the keyboard	It must be possible to access, operate and exit the control point with the keyboard (see Use of the keyboard table, below). Alternatively, it must be possible to run all the functions of the control point with the keyboard. In this case, the use of the keyboard for the control point must be explained in the application and/or Help option. This does not apply to the control points of the application window if default operation has been implemented for them. Exception: If the control point has no relevant function, it does not have to be keyboard-operable. This is the case, for example, if the control point is used to scale a display area, if all content is fully perceptible in the standard display, and if the scaling does not add any value.	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2

No.	Property	Description	Classification	Reference
699	Use of the pointing device	The use of the pointing device for the control point may not be complex. Please note: Complex use of the pointing device means • multipoint operation (e. g. swiping with several fingers), • path-based operation (where the start and end points of the use of the pointing device are not just relevant, but at least one intermediate point is).	Must	EN 301 549: 9.2.5.1, 11.2.5.1
700	Use of the pointing device	It should also be possible to operate the control point without the dragging use of the pointing device. Note: This can be achieved, for example, by clicking on the control point and then clicking on the target position.	Must	WCAG 2.2
701	Click area	The click area of the control point should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: control point

Note: The following requirements only apply if the control point receives the focus with the keyboard.

Possible alternative forms of operation for keyboard users can be, for instance:

- The scaling and moving of the application windows using the Windows methods (ALT+SPACE > move /resize > ARROW UP/DOWN/LEFT/RIGHT),
- The use of keyboard shortcuts, as described in the application and Help option,

- Entry of explicit values for the size, position or rotation of an object in a dialog box, which is initialized by a keyboard shortcut or the context menu, for example (this alternative form of operation should be described in the application and the Help option).

Action	Key	Classification
Focusing of the control point	TAB	Required
Exiting the control point	TAB	Required
Operating the control point	UP/DOWN/LEFT/RIGHT ARROW	Required
Operating the control point with defined increment	CTRL+UP/DOWN/LEFT/RIGHT ARROW	Recommended
Proportional scaling	SHIFT+UP/DOWN/LEFT/RIGHT ARROW	Recommended

Use of the pointing device: control point

Action	Key	Classification
Operating the control point	Dragging the control point (drag and drop)	Required
Operating the control point	Left click to enable and move the pointing device, left click on the target position	Recommended
Proportional scaling	SHIFT + dragging of the control point	Recommended

Programming/interfaces

Note: The following requirements only apply if the control point receives the focus with the keyboard.

No.	Property	Description	Classification	Reference
702	Role	The control point role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
703	Value	The value of the control point (e.g. rotation in degrees) must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7
704	Desktop: Value range	The minimum and maximum value of the control point must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.7

No.	Property	Description	Classification	Reference
705	Status	The status of the control point must be communicated to the Accessibility API (see Element status).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
706	Name	The control point must have a concise and expressive Accessible Name. Note: A control point typically has no visible label. The name of the control point should describe the function (for example, “lower vertical scaling”).	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5
707	Name	If the control point has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
708	Operation	It must be possible to access, operate and exit the control point with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
709	Update	Updates concerning the Accessible Name, value or status of the control point must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
710	Desktop: Position	The size and position of the control point must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5

Combined input field

Online betrachten

Synonyms: Combination field, combo box

See also: Selection list, drop-down list, menu button, input field with autocomplete function

Combined input fields allow for text input and the selection of options from a list, where the list can be opened and closed (see DIN EN ISO 9241-161: 8.7).

In the closed state, a combined input field consists of an input field and a button (with arrow icon) for opening the list, which is located to the right of the input field. In the opened state, a selection list is displayed underneath (possibly with a scrollbar). The options of the selection list can be grouped. The labeling of the groups cannot be selected.

Combined input fields can be implemented very differently. The implementation variants include:

- As long as the input field is empty, in the open state, a selection list with all the options is displayed below the input field. If the input field contains text, only options that contain or begin with the text you have already entered appear in the selection list below the input field. If there are no options matching the entered text, no selection list will be displayed.
- The selection list contains certain commonly used values, regardless of the text input.
- The combined input field has two selection lists. The list entries of one list are independent of the text input; in the second list, the matching list entries are only displayed after the text input.
- The text input is used only to filter the options in the selection list. It is not possible to enter text that does not match any of the default options.

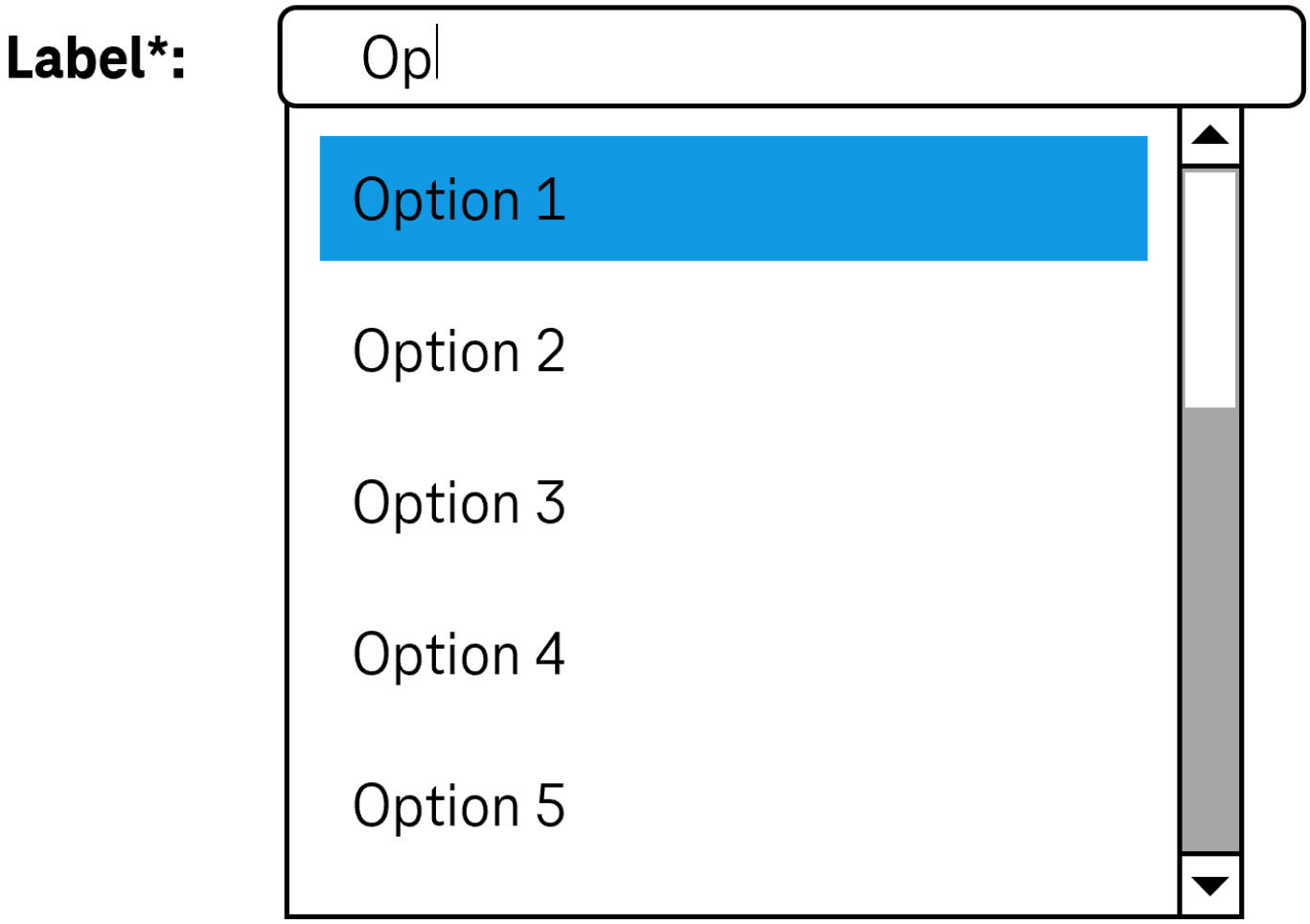

Figure 48: Combined input field

Presentation

The requirements concerning the input field and the selection list are described in the “input field” and/or “selection list” section. Here, only the additional requirements are described which result from the fact that a selection list can be opened with the input field.

No.	Property	Description	Classification	Reference
711	Contrast	The arrow icon for opening and closing the list must have a contrast ratio of at least 3:1 with respect to the neighboring color.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

Operation

The requirements concerning the input field and the selection list are described in the “input field” and/or “selection list” section. Here, only the additional requirements are described which result from the fact that a selection list can be opened with the input field.

No.	Property	Description	Classification	Reference
712	Use of the keyboard	It must be possible to access, operate and exit the combined input field with the keyboard (see Use of the keyboard table, below). Note: The button for showing and hiding the selection list should not receive the keyboard focus separately.	Must	EN 301 549: 11.2.1.1 und 11.2.1.2; ISO 9241-171: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
713	Click area	The click area of the arrow for opening and closing the selection list should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: combined input field

Action	Key	Classification
Opening of the selection list	Desktop: • ALT+DOWN ARROW • UP/DOWN ARROW • F4 • Text input Web: • ALT+DOWN ARROW • UP/DOWN ARROW • Text input	Required
Closing the selection list	Desktop: • ALT+UP ARROW • ENTER • F4 • ESC • TAB Web: • ALT+UP ARROW • ENTER • ESC • TAB	Required

Use of the pointing device: combined input field

Action	Key	Classification
Opening of the selection list	Left click on the arrow	Required
Closing the selection list	Left click on the arrow	Required
Closing the selection list	Left click on a value within the open list	Required
Closing the selection list	Left click outside the combined input field (consisting of the input field and the selection list)	Required

Programming/interfaces

The requirements concerning the input field and the selection list are described in the “input field” and/or “selection list” section. Here, only the additional requirements are described which result from the fact that a selection list can be opened with the input field.

No.	Property	Description	Classification	Reference
714	Role	The combined input field role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.4.1.2, 11.5.2.5
715	Status	The status of the combined input field must be communicated to the Accessibility API (see Element status). Note: This also applies to the “open” or “closed” status.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

Practical tip: combined input field in Web applications

Screen reader output

- JAWS: [label] **edit combo** collapsed | expanded [value] [instruction on text input and operation with the arrow keys]
- NVDA: [label] **combo box** collapsed | expanded [has autoc omplete] editable [value]
- Windows Narrator: [label] [value] **combo edit** collapsed | expanded

HTML

The combined input field should be implemented with the HTML elements <input type=... list=ID> and <datalist id=ID> and <option>.

- The list attribute is suitable to be used for the following values in the type attribute: text, search, url, tel, email.
- The list attribute may also be used according to the HTML specification with the following values in the type attribute. This is not supported by most browsers, however, and sometimes leads to misleading output by the screen readers: date, month, week, time, datetime-local, number, range, color.

The initial value is communicated through the value attribute at the <input> element.

The label should be linked to the combined input field with the <label for=ID> element.

The combined input field can be marked as a required field (required), as disabled (disabled) and/or as read-only (readonly).

The list entries cannot be marked as disabled or read-only. The list entries cannot be grouped.

Further information: 4.10.8 The datalist element - HTML Standard (whatwg.org), 4.10.5.3.9 The list attribute - HTML Standard (whatwg.org)

ARIA

Please note: As the ARIA specification regarding combobox has changed fundamentally several times in recent years, it cannot be guaranteed that the combined input fields implemented with ARIA will be correctly output by all screen readers. The use of the native combined input field is therefore recommended instead. Alternatively, the ARIA implementation should be thoroughly tested with differing browsers and screen readers.
Please note: The ARIA pattern for combined input fields is often used for input fields with autocomplete function or drop-down lists. These three elements differ in their meaning and operation, however, and should not be mixed up.

If the combined input field is not implemented with the HTML elements, it is also necessary to take account of the following:

- The role is communicated with role=combobox. The role must be located at an input field (<input type=text>).
- The value of the input field (value attribute) is communicated as the value of the combined input field.
- The labeling of the combined input field can take place with aria-label or aria-labelledby.
- The status of the drop-down list (closed or open = selection list visible) must be communicated with aria-expanded.
- Reference is made to the selection list from the element with role=combobox using aria-controls.
- The autocomplete behavior of the combined input field is communicated with aria-autocomplete.
- The selection list is marked with role=listbox and its list entries with role=option.
- When navigating through the list entries in the selection list, these must either actually receive the focus, or reference is made to the selected list entry with aria-activedescendant. The first variant is to be preferred.
- The button with the arrow icon which is able to open the selection list does not receive the keyboard focus (tabindex=-1), but should be marked so that it is output by the assistive technology when navigating with the virtual cursor. The button should be labeled expressively, refer to the selection list with aria-controls, and communicate the status of the selection list with aria-expanded.
- The combined input field can be marked as disabled with aria-disabled.
- The combined input field can be marked as read-only with aria-readonly.
- The combined input field can be marked as a required field with aria-required.
- The presentation of the combined input field should be verified in Windows High Contrast mode.
- The visible combined input field and the programmatically focused element should have the same position and size.
- When navigating through the list entries, the focused list entry should be displayed in the visible area.

Further information: combobox role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, Combobox Pattern | APG | WAI | W3C

Drop-down list

Online betrachten

Synonyms: Dropdown, dropdown list field, combobox

See also: Selection list, combined input field, menu button, input field with autocomplete function

Drop-down lists allows for the selection of an option from a list, whereby the list can be opened and closed (see DIN EN ISO 9241-161: 8.11).

In the closed status, the current value (the selected option in the list) is displayed and to the right of it, a button (with arrow icon) for opening the list. In the open status, a selection list with all the options is also shown (possibly with a scrollbar). The current value is highlighted. The options can be grouped. The labeling of the groups cannot be selected.

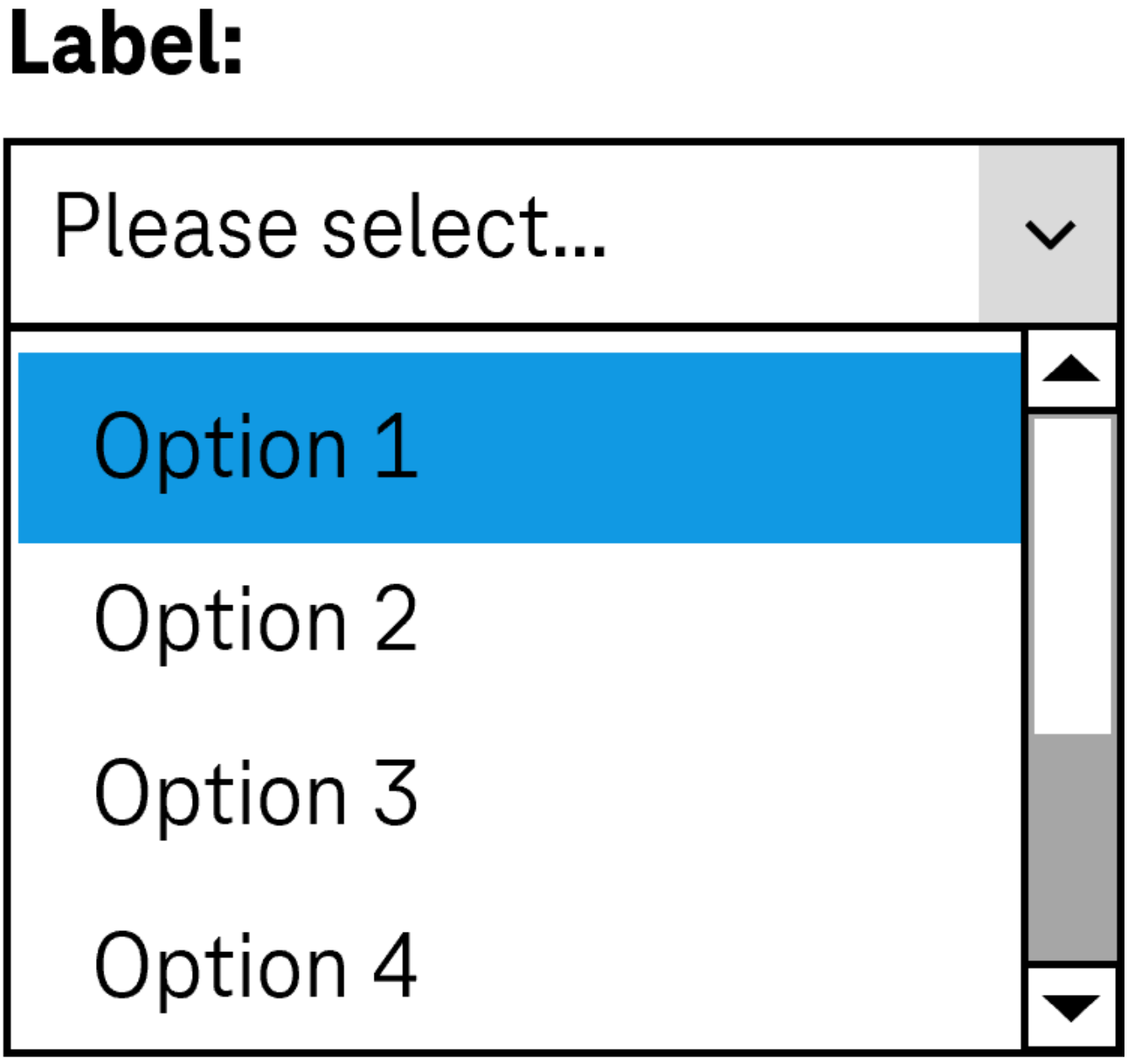

Figure 49: Drop-down list

Presentation

No.	Property	Description	Classification	Reference
716	Contrast	The labeling of the options in the drop-down list must have a contrast of at least 4.5:1. Note 1: This applies to the opened and closed status of the drop-down list and to the selected and unselected options. Note 2: If the options are labeled with graphics rather than text, the contrast of the graphics with respect to the background and the content-bearing areas of the graphics with respect to each other must be at least 3:1.	Must	EN 301 549: 9.1.4.3, 11.1.4.3, 9.1.4.11, 11.1.4.11
717	Contrast	If the selected list entry only differs from the unselected list entry in the open state due to its color (e.g. foreground or background color), the colors must have a contrast ratio of at least 3:1. Note: The selected list entry does not have to be color coded or marked with color only. It can be marked with a check box, for example. In this case, the contrast requirements for the color coding are eliminated as long as the check box has sufficient contrasts.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

No.	Property	Description	Classification	Reference
718	Contrast	If the drop-down list is only identifiable as such on the basis of its color design, the color must have a contrast ratio of at least 3:1 with respect to the neighboring colors. Note 1: A drop-down list might be recognizable as an interactive element on the basis of its border or its background color, for example. Note 2: The requirement does not apply if the drop-down list is clearly identifiable as such, because of its position or labeling, for example.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
719	Contrast	The arrow icon for opening and closing the list must have a contrast ratio of at least 3:1 with respect to the neighboring color.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
720	Label	The drop-down list must have a visible label (see Label).	Must	EN 301 549 9.3.3.2, 11.3.3.2
721	Focus visibility	If the drop-down list receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7
722	Focus visibility	When navigating through the list entries, the current option must be displayed in the visible area and as focused.	Must	EN 301 549: 9.2.4.7, 11.2.4.7
723	Options list	The options should be formulated in such a way that the information that serves the purpose of differentiation is at the beginning.	Should	DIN EN ISO 9241-143: 9.3.4

Operation

No.	Property	Description	Classification	Reference
724	Use of the keyboard	It must be possible to access, operate and exit the drop-down list with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
725	Updates	When focusing and operating the drop-down list, no unexpected change of context may occur. Note: This means that no loss of focus must occur during or after the operation of the drop-down list.	Must	EN 301 549: 9.3.2.1, 11.3.2.1 und 9.3.2.2, 11.3.2.2
730	Click area	The click area of the list entries in the selection list should be at least 24 x 24 px. Note: It should be possible to open the closed drop-down list by both clicking on the current value and clicking on the arrow (see Use of the pointing device).	Should	WCAG 2.2.

Use of the keyboard: drop-down list

Action	Key	Classification
Focusing of the drop-down list	TAB	Required
Exiting the drop-down list	TAB	Required

Action	Key	Classification
Opening the drop-down list	Desktop: • ALT+DOWN ARROW • UP/DOWN ARROW • F4 • Text input Web: • ALT+DOWN ARROW • UP/DOWN ARROW • Text input	Required
Closing the drop-down list	Desktop: • ALT+UP ARROW • ENTER • F4 • ESC • TAB Web: • ALT+UP ARROW • ENTER • ESC • TAB Note: In Web applications, it is also recommended that the drop-down list can be opened with F4.	Required
Operation of the drop-down list (selection of the previous or following value)	UP ARROW, DOWN ARROW	Required
Operation of the drop-down list (selection of the first and last value)	POS1, END	Required with several list entries
Operation of the drop-down list (selection of a value before or after with a defined increment)	PAGE UP, PAGE DOWN Note: The increment should correspond to the amount of visible options.	Required with several list entries

Action	Key	Classification
Operation of the drop-down list (selection of a value which starts with a specific character string)	Entry of one or more characters (within a short time) Note: If two entries start with the same character string, the entries are navigated to one another in sequence.	Required with several list entries

Note: The operation of the drop-down list should be equally possible in both the open and closed states.

Use of the pointing device: drop-down list

Action	Key	Classification
Opening the drop-down list	Left click on the drop-down list (value or arrow)	Required
Closing the drop-down list	Left click on the drop-down list (value or arrow)	Required
Closing the drop-down list	Left click on a value within the open list	Required
Closing the drop-down list	Left click outside the drop-down list	Required
Value selection within the open drop-down list	Left click on a value	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
731	Role	The drop-down list role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.1.4.2, 11.4.1.2, 11.5.2.5
732	Value	The value of the drop-down list must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.1.4.2, 11.4.1.2, 11.5.2.7
733	Desktop: Element hierarchy	The parent / child relationships of the elements within the list must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9

No.	Property	Description	Classification	Reference
734	Status	The status of the drop-down list must be communicated to the Accessibility API (see Element status). Note: This also applies to the “open” or “closed” status.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
735	Name	The drop-down list must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
736	Name	If the drop-down list has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
737	Name	The group label of the list entries must (if available) be communicated to the Accessibility API.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
738	Operation	It must be possible to access, operate and exit the drop-down list with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
739	Update	Updates concerning the Accessible Name, value or status of the drop-down list must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
740	Desktop: Position	The size and position of the drop-down list and – if open – the selection list and its list entries must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: drop-down list in Web applications

Screen reader output

- JAWS: [label] **combo box** [value] [instruction on operation with the arrow keys]
- NVDA: [label] **combo box** [value] collapsed
- Windows Narrator: [label] [value] **combo box** collapsed

HTML

The drop-down list should be implemented with the HTML elements <select> and <option> (without the multiple and size attributes).

The list entries can be grouped with the <optgroup> element. The grouping should be avoided, however, as many screen readers do not output the group label (which is specified with the label attribute).

The initially selected list entry can be set with the selected attribute.

The label should be linked to the drop-down list with the <label for=ID> element. A drop-down list, a group of list entries and the individual list entries can be marked as disabled (disabled), but not read-only (readonly).

A drop-down list can be marked as a required field with required. In this case, however, the first list entry must be empty.

Further information: 4.10.7 The select element - HTML Standard (whatwg.org)

ARIA

Please note: As the ARIA specification regarding combobox has changed fundamentally several times in recent years, it cannot be guaranteed that the ARIA drop-down list will be correctly output by all screen readers. The use of the HTML drop-down list is therefore recommended instead. Alternatively, the ARIA implementation should be thoroughly tested with differing browsers and screen readers.

Please note: Depending on the operating system, an HTML drop-down list is communicated to the Accessibility API differently. Under MacOS, a drop-down list is a menu button which opens a menu. If an ARIA drop-down list is used, no cross-platform solution is possible, as it would be necessary to use either the combobox role for Windows or the button role with aria-haspopup=menufor MacOS. The use of HTML drop-down lists is therefore recommended.

If the drop-down list is not implemented with the HTML element, it is also necessary to take account of the following:

- The role is communicated with role=combobox. The role may not be located at an input field. Instead of this, a element should be used.
- The text content of the element with role=combobox is communicated as the value of the drop-down list.
- The button with the arrow icon which is able to open the selection list is marked so that it does not receive the keyboard focus and is not output by the assistive technology (e.g. with aria-hidden=true).
- The labeling of the drop-down list can take place with aria-label or aria-labelledby.
- The status of the drop-down list (closed or open = selection list visible) must be communicated with aria-expanded.

- Reference is made to the selection list from the element with role=combobox using aria-controls.
- The selection list is marked with role=listbox and its list entries with role=option.
- When navigating through the list entries in the selection list, these must either actually receive the focus, or reference is made to the selected list entry with aria-activedescendant. The first variant is to be preferred. -The drop-down list can be marked as disabled with aria-disabled.
- The drop-down list can be marked as read-only with aria-readonly.
- The drop-down list can be marked as a required field with aria-required.
- The presentation of the drop-down list should be verified in Windows High Contrast mode.
- The visible drop-down list and the programmatically focused element should have the same position and size.
- When navigating through the list entries, the focused list entry should be displayed in the visible area.

Further information: combobox role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, Select-Only Combobox Example | APG | WAI | W3C

Slider

Online betrachten

Synonyms: Analog form element, range slider, range control

See also: Spin button, radio buttons, drop-down list, scrollbar, separator, progress bar

A slider is used to select a value from a continuous range of values (see DIN EN ISO 9241-161: 8.2). Sliders are mainly used for numerical values.

A slider consists of a bar that represents the entire range of values and a drag point which displays the selected value and through which the value is changed. Sliders can also feature an unlabeled or labeled scale of values on the bar. A slider can be aligned vertically or horizontally.

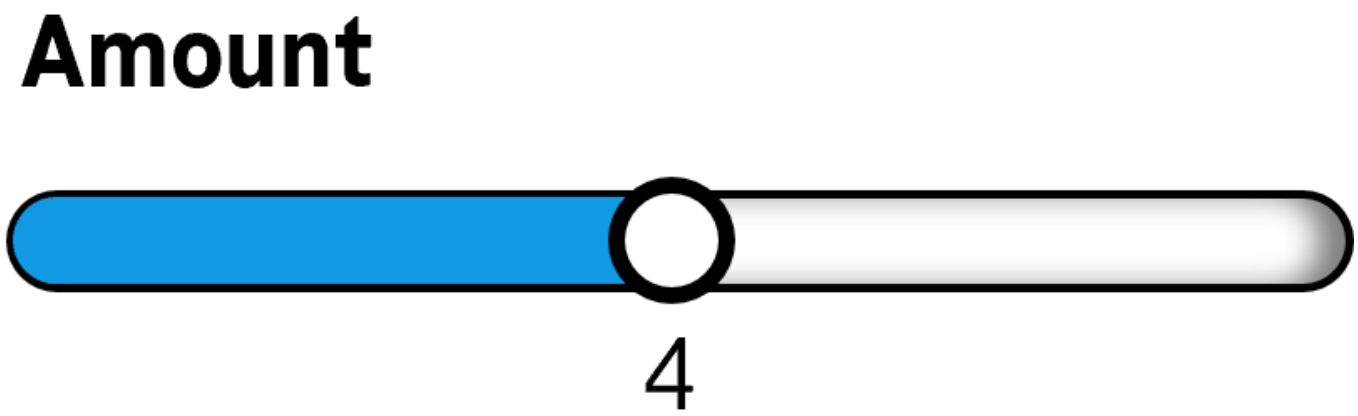

Figure 50: Slider

Presentation

No.	Property	Description	Classification	Reference
741	Contrast	The bars of the slider must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
742	Contrast	The drag point of the slider must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
743	Contrast	If the drag point is only located within the bar, each must have a contrast ratio of at least 3:1 with respect to each other. Note: The contrast ratio of the drag point and the bar can also be maintained with a corresponding border around the drag point.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
744	Contrast	All the text elements of the slider (e. g. display of current and possible values) must have a contrast ratio of at least 4.5:1.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
745	Label	The slider must have a visible label (see Label). Note: If the purpose of the slider is clear from the visual context, the labeling can be omitted. This applies to sliders that are used for time control of videos.	Must	EN 301 549 9.3.3.2, 11.3.3.2
746	Focus visibility	If the slider receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7
747	Value	The current value of the slider should be visually perceptible in text form.	Should	DIN EN ISO 9241-161: 8.2.2

Operation

No.	Property	Description	Classification	Reference
748	Use of the keyboard	It must be possible to access, operate and exit the slider with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
749	Use of the pointing device	The use of the pointing device to operate the slider must not be complex, i.e. if the value change of the slider is only possible by dragging the drag point, the value change must also be possible if the pointing device is no longer on the bar. Note 1: Complex use of the pointing device means • multipoint operation (e. g. swiping with several fingers), • path-based operation (where the start and end points of the use of the pointing device are not just relevant, but at least one intermediate point is). Note 2: The slider can be combined with an input field or spin button to provide an alternative method of value selection which does not require the complex use of the pointing device. In this case, the current value of both form elements must be synchronized automatically.	Must	EN 301 549: 9.2.5.1, 11.2.5.1

No.	Property	Description	Classification	Reference
750	Use of the pointing device	It should also be possible to operate the slider without the dragging use of the pointing device. Note: This can be done enabling the value selection by clicking on the bar, for example.	Should	WCAG 2.2
751	Updates	When focusing and operating the slider, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1, 9.3.2.2, 11.3.2.2
752	Click area	The click area of the drag point of the slider should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2
753	Click area	Sliders should only be used when it is not necessary to enter an exact value. Alternatively, the slider should be designed so that the individual selectable values have a spacing of at least 24 px.	Should	WCAG 2.2

Use of the keyboard: slider

Action	Key	Classification
Focusing of the slider	TAB	Required
Exiting the slider	TAB	Required
Operation of the slider (selection of the previous or following value)	DOWN/LEFT ARROW, UP/RIGHT ARROW Note: Two arrow keys should function in each direction regardless of the orientation, as the orientation is often not correctly perceptible with AT.	Required
Operation of the slider (selection of the first and last value)	POS1, END	Required
Operation of the slider (selection of a value before or after with a defined increment)	PAGE UP, PAGE DOWN Note: The increment depends on the amount of possible values. An increment of 10 for instance, would make sense with 100 values.	Recommended

Use of the pointing device: slider

Action	Key	Classification
Value change	Dragging of the drag point (drag and drop)	Required
Value change	Left click on an area of the bar	Recommended

Programming/interfaces

No.	Property	Description	Classification	Reference
754	Role	The role of the slider must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
755	Value	The value of the slider must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7
756	Desktop: Value range	The minimum and maximum value of the slider must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.7
757	Status	The status of the slider must be communicated to the Accessibility API (see Element status).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
758	Orientation	The orientation of the slider (vertical or horizontal) must be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2
759	Name	The slider must have a concise and expressive Accessible Name.	Must	EN 301 549: EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
760	Name	If the slider has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
761	Operation	It must be possible to access, operate and exit the slider with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
762	Update	Updates concerning the Accessible Name, value or status of the slider must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15

No.	Property	Description	Classification	Reference
763	Desktop: Position	The size and position of the slider and the drag point must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: sliders in Web applications

Screen reader output

- JAWS: [label] **up down slider** | **left right slider** [value] [instruction on operation with the arrow keys]
- NVDA: [label] **slider** [value]
- Windows Narrator: [label] **slider** at [value] current value [value] minimum value [minimum value] maximum value [maximum value]

HTML

The slider should be implemented with the HTML element <input type=range>. The initial value can be set with the value attribute.

The minimum and maximum values and increment are set with the min, max and step attributes. It should be noted that these values are imperceptible with many forms of assistive technology.

The label should be linked to the slider with the <label for=ID> element.

A slider can be marked as disabled (disabled), but not read-only (readonly) or as a required field (required).

It is not possible to set the orientation of the slider. Most browsers display the slider horizontally.

Further information: 4.10.5.1.13 Range state (type=range) - HTML Standard (whatwg.org)

ARIA

Please note: On mobile devices, an ARIA slider may not be operated with assistive technology, because no gesture commands for the operation of non-native sliders have been implemented.

If the slider is not implemented with the HTML element, it is also necessary to take account of the following:

- The role is communicated with role=slider.
- The current value must be specified with aria-valuenow.
- The minimum and maximum values can be specified with aria-valuemin and aria-valuemax.
- The labeling can take place with aria-label or aria-labelledby.
- The presentation of the slider should be verified in Windows High Contrast mode.
- The visible slider and the programmatically focused element should have the same position and size.

The ARIA slider differs from the HTML slider in the following ways:

- An increment cannot be specified.
- The orientation can be specified with aria-orientation. Frequently, the orientation is not output by the assistive technology, so that the operation with all the arrow keys should be possible.

- aria-valuetext can also be used to specify a value in text form which should be then output by the assistive technology instead of the value in aria-valuenow.
- Sliders with multiple drag points can be implemented for the selection of a range of values, for example.
- The slider can be marked as read-only with aria-readonly.

Further information: slider role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, Slider Pattern | APG | WAI | W3C

Scrollbars

Online betrachten

Synonyms: Scrollbar

See also: Slider, pagination, carousel, control point

A scrollbar is for scrolling through the entire page, part of a page or parts of an element (such as list entries in a drop-down list) in the visible area. The scrollbar also serves the visualization of the current position and the overall size of the page, parts of the page or elements (see DIN EN ISO 9241-161: 8.35).

A scrollbar consists of a moving bar and the scroll box. The moving bar represents the total length or width of the scrollable area. The scroll box shows the location and size of the visible section and also allows the visible section to be moved.

At the start and end of the moving bar there is generally a button with an arrow icon for incremental scrolling.

Vertical scrollbars are located on the right-hand border of the scrollable area. Horizontal scrollbars are located on the lower border of the scrollable area.

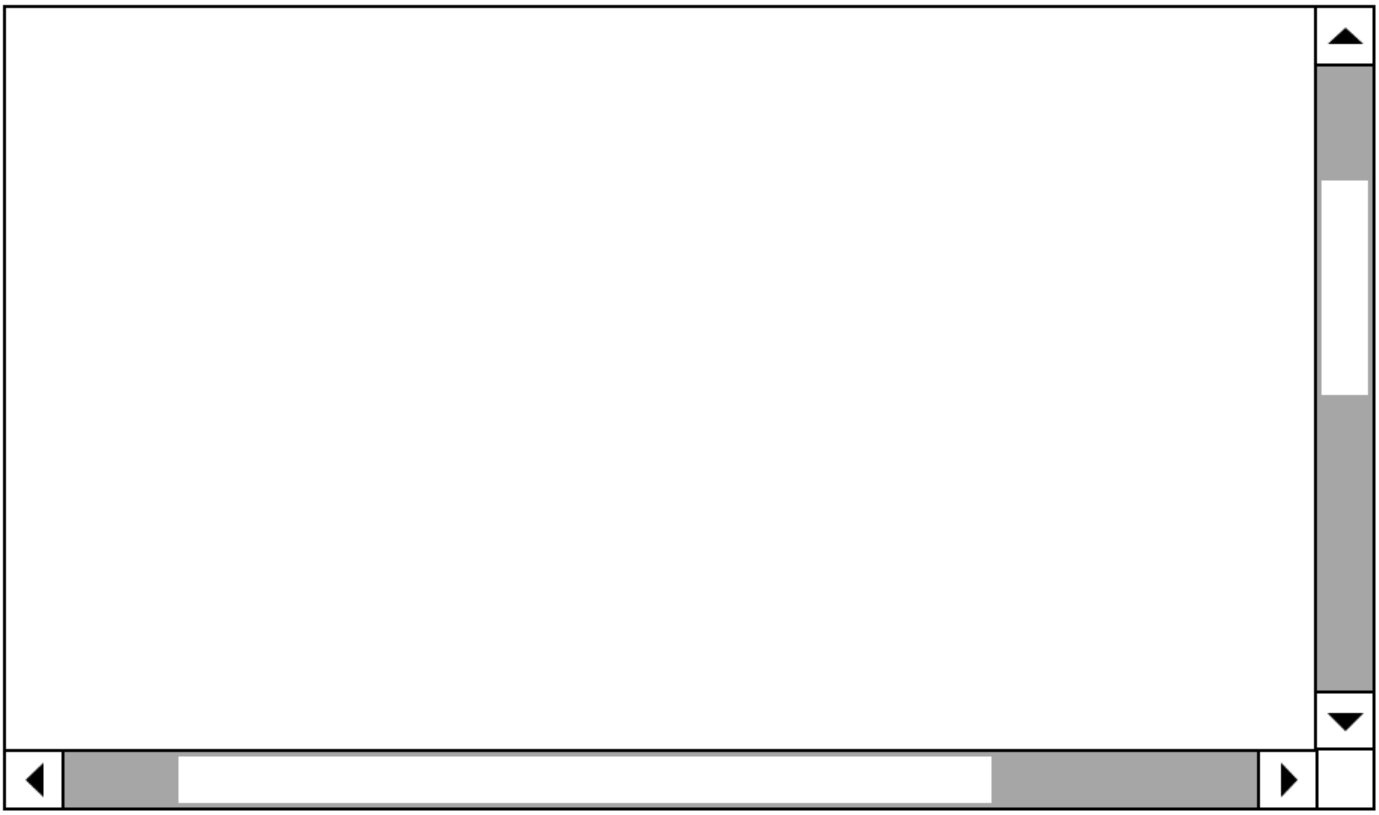

Figure 51: Vertical and horizontal scrollbar

Presentation

No.	Property	Description	Classification	Reference
764	Contrast	The icons of the buttons on the border of the scrollbar must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
765	Contrast	The scroll box must have a contrast ratio of at least 3:1 with respect to the moving bar. Note: The contrast ratio of the scroll box and moving bar can also be maintained with a corresponding border around the moving bar.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
766	Contrast	If the moving bar is only identifiable as such on the basis of its color design, the color must have a contrast ratio of at least 3:1 with respect to the neighboring colors. Note 1: A moving bar might be recognizable as an interactive element on the basis of its border or its background color, for example. Note 2: This requirement does not apply if the entire scrollbar is clearly recognizable as such, because of its position in combination with the buttons at the start and end and the scroll box, for example.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
767	Amount	The page contents must wrap in such a way that they only need to be scrolled vertically or horizontally for a display size up to a minimum of 320 x 256 px. This does not apply to required two-dimensional content (see Zoom).	Must	EN 301 549 9.1.4.10, 11.1.4.10

Operation

No.	Property	Description	Classification	Reference
768	Use of the keyboard	It must be possible to scroll in the areas with the keyboard (see Use of the keyboard table, below). Note: The scrollbars should not receive keyboard focus, but the scrollable areas and /or elements within the scrollable areas.	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
769	Use of the pointing device	The use of the pointing device for the scrollbar may not be complex. Please note: Complex use of the pointing device means • multipoint operation (e. g. swiping with several fingers), • path-based operation (where the start and end points of the use of the pointing device are not just relevant, but at least one intermediate point is).	Must	EN 301 549: 9.2.5.1, 11.2.5.1
770	Use of the keyboard	It should also be possible to operate the scrollbar without the dragging use of the pointing device. Note: This can be done enabling the scrolling by clicking on the moving bar or the buttons, for example.	Should	WCAG 2.2

No.	Property	Description	Classification	Reference
771	Updates	When scrolling, no unexpected change of context should occur.	Must	WCAG 2.1: 3.2.5 (AAA)
772	Animations	When scrolling, there should not be any other visual animation apart from the moving of the visible area.	Should	WCAG 2.1: 2.3.3 (AAA)
773	Click area	The scrollbars should be at least 24 px wide.	Should	WCAG 2.2
774	Click area	The click area of the buttons and scroll box of the scrollbar should be at least 24 x 24 px.	Should	WCAG 2.2

Use of the keyboard: scrollbar

Action	Key	Classification
Focusing of the scrollable area or elements within the scrollable area	TAB Note 1: If the scrollable area does not contain any elements that allow for scrolling with the arrow keys, then the area itself must receive the focus. Note 2: Elements that are themselves controlled by the arrow keys (such as input fields, selection lists and radio buttons) do not allow for the scrolling of the area in which they are located.	Required
Exiting the scrollable area	TAB	Required
Scrolling of a focused item to the visible area	When receiving the keyboard focus	Required
Vertical scrolling	UP/DOWN ARROW	Required
Horizontal scrolling	RIGHT/LEFT ARROW	Required
Vertical scrolling (quick navigation)	PAGE UP/PAGE DOWN (CTRL +) POS1/END	Recommended

Use of the pointing device: scrollbar

Action	Key	Classification
Incremental scrolling	Left click on the buttons on the edge of the scrollbar	Required
Scrolling (quick navigation)	Left click on the moving bar outside the scroll box	Required
Scrolling to a specific position	Dragging of the scroll box (drag and drop)	Required

Note: The scrolling capabilities of the pointing devices should also be supported (e.g., mouse scroll wheel, gesture commands for scrolling on the touch-pad).

Programming/interfaces

No.	Property	Description	Classification	Reference
775	Role	The role of the scrollbar must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
776	Role	If the scrollable area receives the keyboard focus and does not have the role of control element, the scrollable area role must be communicated to the Accessibility API. Note: If the technology used does not know the role of the scrollable area, reference should be made in the Accessible Description that the element for scrolling receives the focus.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
777	Value	The values of the scrollbar must be communicated to the Accessibility API (see Accessibility API). Note: The scrollbar has the following values: • Size of scroll box relative to the size of the moving bar (represents the size of the visible section) • Position of scroll box within the moving bar.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7
778	Desktop: Value range	The minimum and maximum value of the scrollbar must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.7
779	Status	The status of the scrollbar must be communicated to the Accessibility API (see Element status).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
780	Orientation	The orientation of the scrollbar (vertical or horizontal) must be communicated to the Accessibility API.	Must	EN 301 549: 9.4.1.2, 11.4.1.2
781	Desktop: Position	The new position of the focused element must be communicated to the Accessibility API after the scrolling of the area (see Accessibility API).	Must	EN 301 549: 11.5.2.5, 11.5.2.13, 11.5.2.15

Radio buttons

Online betrachten

Synonyms: Option fields, selection button, radio button group

See also: Selection lists, drop-down lists, check boxes

Radio buttons are used for selecting mutually exclusive options (see DIN EN ISO 9241-161: 8.33).

A radio button consists of an indicator which indicates whether the option is selected or not selected. A radio button group consists of several radio buttons with their labels and a group label. Radio buttons must be grouped.

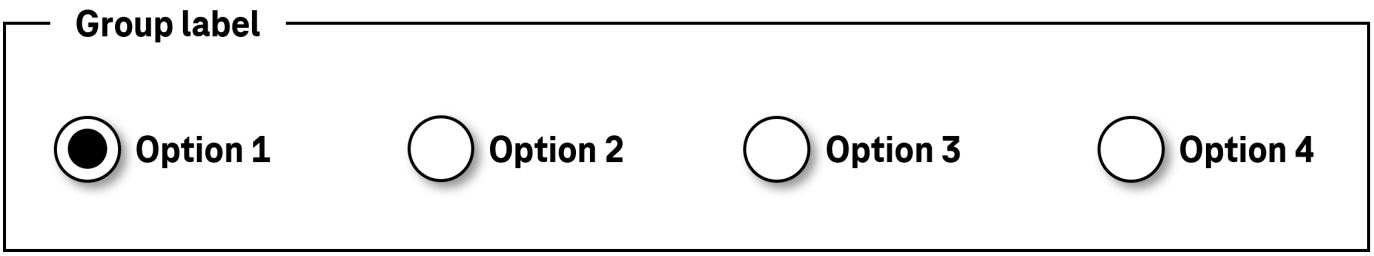

Figure 52: Radio buttons in a group

Presentation

No.	Property	Description	Classification	Reference
782	Contrast	The border of the radio button must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
783	Contrast	The symbol which relays the status (circle) must have a contrast ratio of at least 3:1 with respect to the neighboring color.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
784	Label	The radio buttons must have a visible label (see Label).	Must	EN 301 549 9.3.3.2, 11.3.3.2
785	Label	The label of the radio button should be located to the right of the radio button.	Should	DIN EN ISO 9241-125: 5.1.15
786	Label	The label of the group should be unambiguous and understandable within the context (see Group).	Should	DIN EN ISO 9241-171: 8.1.2, 8.1.3
787	Focus visibility	If the radio button receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
788	Use of the keyboard	It must be possible to access, operate and exit the radio buttons with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.25
789	Updates	When focusing and operating the radio buttons, no unexpected change of context may occur. Note: This means that no loss of focus must occur during or after the operation of the radio buttons.	Must	EN 301 549: 9.3.2.1, 11.3.2.1, 9.3.2.2, 11.3.2.2
790	Click area	The click area of the radio buttons should be at least 24 x 24 px (see Use of the pointing device). Note: It should be possible to operate the radio buttons by both clicking on the radio button and clicking on the respective label (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: radio buttons

Action	Key	Classification
Focusing of the radio button group	TAB Note: The selected radio button receives the focus. If no radio button is selected, the first radio button receives the focus.	Required
Exiting the radio button group	TAB	Required
Selecting a radio button	SPACE	Required
Operation of the radio button group (selecting a radio button)	UP/DOWN RIGHT/LEFT ARROW Note: In this context, the navigation must be limited to the radio button group.	Required
Navigation within the radio button group (without changing the selection)	CTRL + UP/DOWN RIGHT/LEFT ARROW	Recommended

Use of the pointing device: radio buttons

Action	Key	Classification
Selecting a radio button	Left click	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
791	Role	The radio button role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
792	Value	The value of the radio button (selected, not selected) must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7

No.	Property	Description	Classification	Reference
793	Desktop: Element hierarchy	The parent / child relationships of the elements within the radio button group must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9
794	Status	The status of the radio button must be communicated to the Accessibility API (see Element status).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
795	Name	The radio buttons must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
796	Name	If the radio buttons have a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
797	Name	If the radio button group has a label, this must be communicated to the Accessibility API as the Accessible Name of the group (see Group).	Must	EN 301 549: 9.1.3.1, 11.1.3.1
798	Operation	It must be possible to access, operate and exit the radio button group with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
799	Update	Updates concerning the Accessible Name, value or status of the radio buttons and radio button group must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
800	Desktop: Position	The size and position of the radio button must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: radio buttons in Web applications

Screen reader output

- JAWS: [group label] [label] **radio button** checked | not checked [position and amount] [instruction on operation with the arrow keys]
- NVDA: [group label] **grouping** [label] **radio button** checked | not checked [position and amount]
- Windows Narrator: [label] **radio button** selected | not selected [position and amount]

HTML

The radio button group should be marked with the HTML element <fieldset> and labeled with the <legend> element.

The radio buttons should be implemented with the HTML element <input type=radio> . Radio buttons that belong to the same group must have the same value in the name attribute and may not be located in different <form> elements.

The initially selected status is set with the checked attribute. An initial radio button should be selected in each radio button group, as keyboard users automatically make a selection when navigating through the radio buttons which cannot be undone. The radio button should be initially selected which is either most likely to be selected or which contains a neutral option (e.g. “not applicable”).

The label should be linked to the respective radio button using the <label for=ID> element in order to expand the click area of the radio button by its label.

A radio button and the radio button group can be marked as disabled (disabled), but not as read-only (readonly).

If a radio button is marked as a required field with required, this applies to the entire radio button group, i. e. if any radio button was selected, this is sufficient for submitting the form. To ensure that the required field label is perceptible with the assistive technology on all radio buttons, using the required attribute on all radio buttons, or alternatively, adding a required text field reference (e.g. an asterisk) in the group label (<legend>) is recommended. If a radio button is initially preselected with the checked attribute, no required field label is required.

Error messages should not be associated with each individual radio button, but with the group.

Further information: 4.10.5.1.16 Radio Button state (type=radio) - HTML Standard (whatwg.org)

ARIA

If the radio button group is not implemented with the HTML element, it is also necessary to take account of the following:

- The radio buttons are located in an element which is marked with role=radiogroup.
- The radio button group can be labeled with aria-labelledby or aria-label.
- The role of the radio buttons is communicated with role=radio.
- The status is communicated with aria-checked=true|false and must be updated during operation.
- The labeling of the radio buttons can take place using text content or aria-labelledby.
- The radio button group can be marked as read-only with aria-readonly. The radio buttons cannot be marked as read-only.

- In contrast to HTML, aria-required is not used to mark the radio buttons as a required input, but the radio button group (role=radiogroup).
- The presentation of the radio buttons should be verified in Windows High Contrast mode.
- The visible radio buttons and the programmatically focused elements should have the same position and size.

Further information: radio role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link) , radiogroup role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link) , Radio Group Pattern | APG | WAI | W3C, Checkbox Pattern | APG | WAI | W3C

Check box

Online betrachten

Synonyms: Control box, control field, selection box

See also: Toggle switch, switch, radio buttons, selection list with multiple selection

A check box serves the selection of the options “selected” or “not selected” (see DIN EN ISO 9241-161: 8.4). A check box can also relay the status of a subordinate check box group (“all selected”, “none selected” or “some selected”).

A check box consists of a square frame and an indicator (check mark) which indicates whether the check box has been selected, has not been selected, or whether some of the subordinate group have been selected.

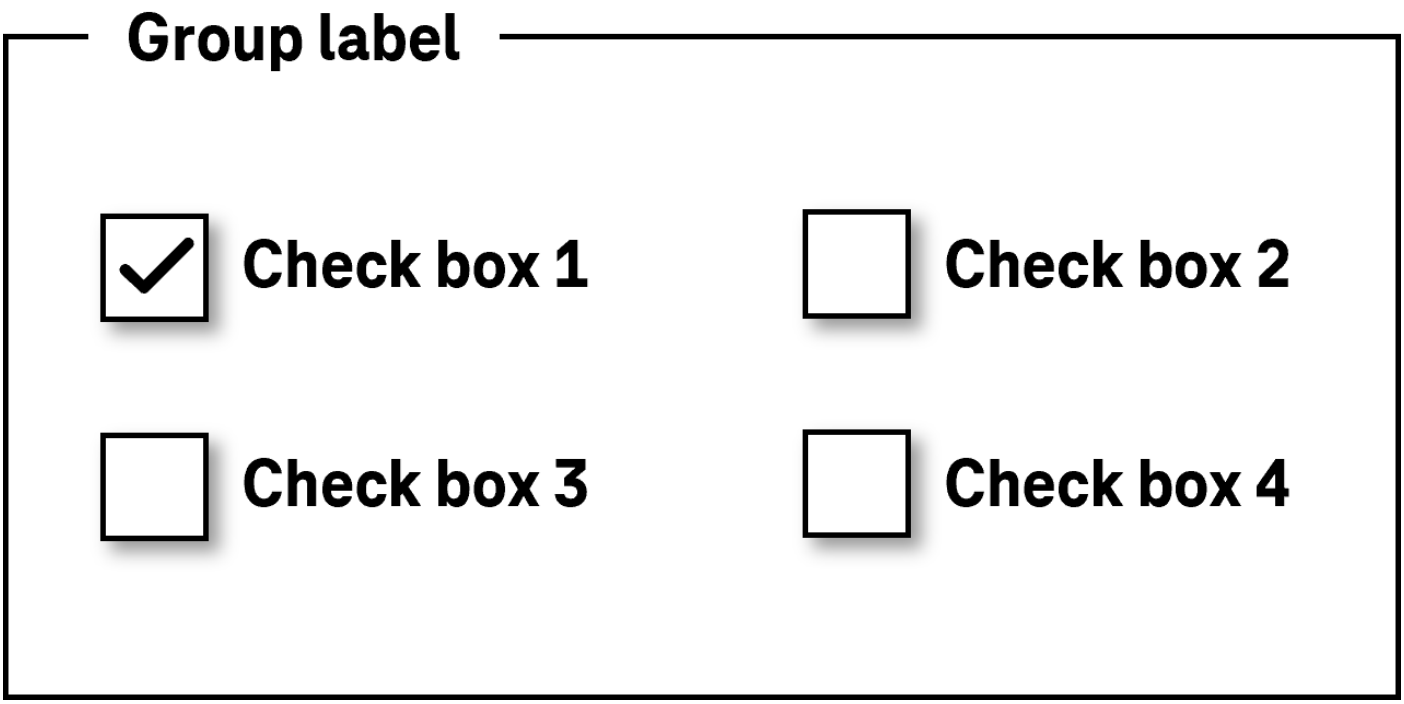

Figure 53: Check boxes in a group

Presentation

No.	Property	Description	Classification	Reference
801	Contrast	The border of the check box must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
802	Contrast	The symbol which relays the “selected” and “some selected” states (check mark) must have a contrast ratio of at least 3:1 with respect to the neighboring color.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
803	Label	The check box must have a visible label (see Label).	Must	EN 301 549 9.3.3.2, 11.3.3.2
804	Label	The label of the check box should be on the right-hand side of the check box.	Should	DIN EN ISO 9241-125: 5.1.15
805	Focus visibility	If the check box receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
806	Use of the keyboard	It must be possible to access, operate and exit the check box with the keyboard (see Use of the keyboard table).	Must	EN 301 549: 9.2.1.1, 9.2.1.2, 11.2.1.1, 11.2.1.2
807	Updates	When focusing and operating the check box, no unexpected change of context may occur. Note: This means that no loss of focus must occur during or after the operation of the check box.	Must	EN 301 549: 9.3.2.1, 9.3.2.2, 11.3.2.1, 11.3.2.2
808	Click area	The click area of the check box should be at least 24 x 24 px (see Use of the pointing device). Note: It should be possible to operate the check box by both clicking on the check box and clicking on the label (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: check box

Action	Key	Classification
Focusing of the check box	TAB	Required
Exiting the check box	TAB	Required
Operation of the check box (value change)	SPACE	Required
Desktop: Navigating within a check box group	LEFT/UP ARROW, RIGHT/DOWN ARROW Note: Navigation does not change the value.	Recommended

Use of the pointing device: check box

Action	Key	Classification
Operation of the check box (value change)	Left click	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
809	Role	The check box role must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2,11.4.1.2, 11.5.2.5
810	Value	The value of the check box (selected, partially selected, not selected) must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.7
811	Status	The status of the check box must be communicated to the Accessibility API (see Element status).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
812	Name	The check box must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
813	Name	If the check box has a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
814	Operation	It must be possible to access, operate and exit the check box with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
815	Update	Updates concerning the name, value or status of the check box must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
816	Desktop: Position	The size and position of the check box must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: check box in Web applications

Screen reader output

- JAWS: [label] **check box** checked | not checked | partially checked [instruction on operation with the space bar]
- NVDA: [label] **check box** checked | not checked | half checked
- Windows Narrator: [label] **check box** checked | unchecked | indeterminate

HTML

The check box should be implemented with the HTML element <input type=checkbox>.

The initially selected status is set with the checked attribute. The “some selected” state can only be set using JavaScript with the indeterminate=true characteristic.

The labeling of the check box should be linked to the check box using the <label for=ID> element in order to expand the click area of the check box by its label.

A check box can be marked as disabled (disabled), but not as read-only (readonly).

A check box can be marked as a required field with required. This is only useful in cases where the check box has to be submitted in the “selected” (checked) status. However, if at least one check box in a group of check boxes must be selected, the check box should not be marked with required, but the required field label should be implemented in the group. As the group cannot be marked with required, the required field label should be in text form (e.g. an asterisk within the group label).

Check boxes that belong together should be positioned within a labeled Form field group. The <fieldset> element is used for the group and the <legend> element is used for the group label.

Error messages that do not relate to a single check box but to the check box group should not be linked to each individual check box but to the group.

Further information: 4.10.5.1.15 Checkbox state (type=checkbox) - HTML Standard (whatwg.org)

ARIA

If the check box is not implemented with the HTML element, it is also necessary to take account of the following:

- The role is communicated with role=checkbox.
- The status is communicated with aria-checked=true|false|mixed and must be updated during the operation.
- The labeling can take place by text content or aria-labelledby.
- The check box can be marked as read-only with aria-readonly.
- The presentation of the check box should be verified in Windows High Contrast mode.
- The visible check box and the programmatically focused element should have the same position and size.

Further information: checkbox role - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, Checkbox Pattern | APG | WAI | W3C

Composite control elements

Online betrachten

Table of contents

- Pagination
- Composite form fields
- Accordion
- Date picker
- Color picker
- Carousel

Pagination

Online betrachten

Synonyms: Page navigation

See also: Scrollbars

Pagination is used to navigate through pages or sequential elements (for example, with tables).

The requirements for the individual control elements are described for the respective control element.

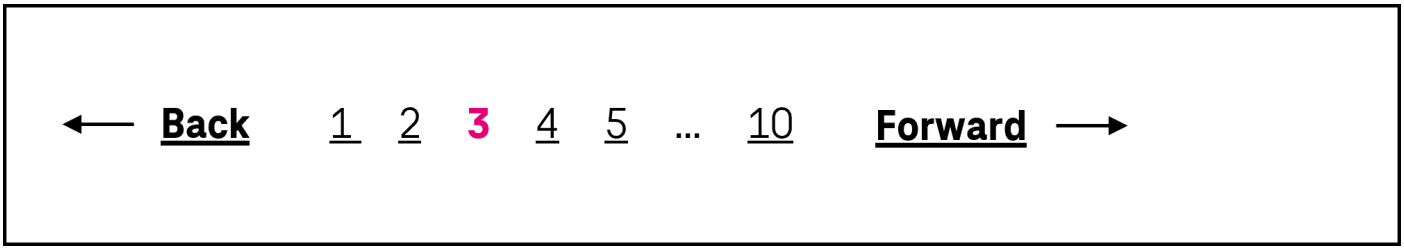

Figure 54: Pagination

Presentation

No.	Property	Description	Classification	Reference
817	Contrast	The highlighting for the current page must have a contrast ratio of at least 3:1 with respect to the background and the other pages.	Must	EN 301 549: 9.1.4.11, 11.1.4.11

Operation

No.	Property	Description	Classification	Reference
818	Use of the keyboard	It must be possible to access, operate and exit the page navigation with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
819	Updates	When focusing and using the control elements for the page navigation, no unexpected change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1
819	Updates	When operating the control elements for the pagination, no loss of focus may occur. Note: During the operation, the focus must remain on the control elements of the pagination or be placed at the start of the area controlled by the pagination.	Must	EN 301 549: 9.2.4.3, 11.2.4.3
820	Click area	The click area of the control elements for the page navigation should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: page navigation

Action	Key	Classification
Focusing of the first element of the page navigation	TAB	Required
Exiting the page navigation	TAB	Required
Navigating within the page navigation	TAB or ARROW buttons (depending on the used elements)	Required
Operating interactive elements in the page navigation	Corresponding to the respective element	Required
Navigating to the previous or next page (if the focus is in the element which is controlled by the page navigation)	PAGE UP, PAGE DOWN ARROW keys	Recommended
Navigating to the first or last page (if the focus is in the element which is controlled by the page navigation)	POS 1, END	Recommended

Programming/interfaces

No.	Property	Description	Classification	Reference
821	Name	Each control element of the pagination must have a concise and expressive Accessible Name . Note 1: If the navigation buttons are visually labeled with “1”, “2”, “3” or “<” and “>”, the Accessible Name should be called • either “page 1”, “page 2”, “page 3” and/or “previous page” and “next page”, or • “1”, “2”, “3” and/or “back” and “forward”, and the buttons should be located in a group which is labeled with “pagination”, for example.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8

No.	Property	Description	Classification	Reference
		Note 2: If the context of the pagination is visually clear but programmatically unclear (because the pagination might refer to a page or table on the page, for example), this context should be communicated the Accessibility API (with the Accessible Name, the Accessible Description, or the labeling of a group, for example).		
822	Desktop: Element hierarchy	The parent / child relationships of the elements within the pagination must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9
823	Status	The status of the control elements for the pagination must be communicated to the Accessibility API (see Element status). Note: In particular, this applies to the “current page” status and the “disabled” status (e.g. to the buttons for the first and previous page, if page 1 is the current page).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

Composite form fields

Online betrachten

Composite form fields consist of two or more form fields that have a visual label.

The requirements for the individual form fields are described for the respective form field type. Here, only the additional requirements are described which result from the fact that a label refers to several fields.

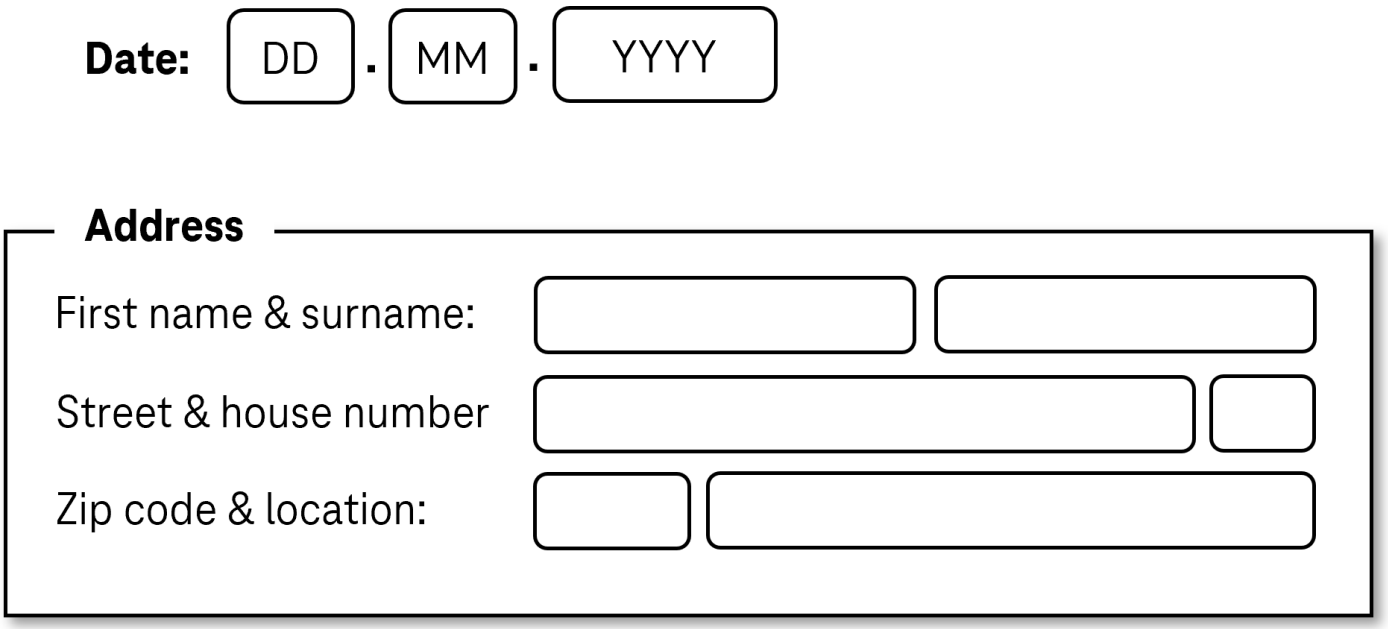

Figure 55: Compoiste input fields

Presentation

No.	Property	Description	Classification	Reference
824	Label	Each form field should have a tooltip that contains the specific label. Note: In the address form, this would be „zip code“ or „location“. In the case of form fields for the time period, this is, for example, „start“ and „end“.	Should	DIN EN ISO 9241-143: 9.6.11

Programming/interfaces

No.	Property	Description	Classification	Reference
825	Name	Each form field must have a concise and expressive Accessible Name. Note: In the address form, this would be „zip code“ or „location“. In the case of form fields for the time period, this is, for example, „period: start“ and „period: end“ .	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
826	Name	The visible label must match with or be contained in the Accessible Name (see Practical tip: composite form fields).	Must	EN 301 549: 9.2.5.3, 11.2.5.3

Practical tip: composite form fields

With composite form fields, the following problems may arise:

- The spacing between the label and the form field is often greater, which makes the purpose of the form fields more difficult to identify, when using a screen magnifier or for cognitively disabled people, for example.
- Each field requires an expressive Accessible Name which is communicated to the Accessibility API, even if the field does not visually have an explicit label.
- The visible label must match with or be contained in the Accessible Name, even if the visible label refers to several fields.
- In applications that support the virtual cursor, the reading sequence with the screen reader may not be correct, for example, because the labels of the fields are output first, followed by the form fields (without the label being output again).

For these reasons, we recommend that the use of composite form fields is avoided and each field is assigned a label visually and programmatically.

Otherwise, and at the least, the following information should be taken into account:

- If the visual label contains only one overarching label, (e.g. „date“ for three date fields in which the day, month and year are entered), these should be included in the Accessible Name together with a clear label (e.g. „date day“, „date month“, „date year“). Alternatively, a field can be used for entering the full date, for example.
- If the visual label contains the label of each field (for example, „zip code“ and „location“), the Accessible Name should match with the individual labels (for example, „zip code“ and „location“).
- Complex forms should be generally redesigned in order to meet accessibility requirements. The following screenshot, for example, contains two radio buttons with the „on“ label in an unlabeled group. From the visual context, it can be seen that the first radio button refers to a repetition of a recurring event on an explicitly defined date, while the second refers to the repetition on an implicitly defined day of the week. The group labeling of the radio buttons could be „repeat on one“,

and the radio buttons themselves could be labeled visually and programmatically with „date“ and „day of the week“, whereby the currently selected date or the currently selected day of the week can be communicated as the Accessible Description. The form fields that are located behind the two radio buttons and which allow for the selection of the date or day of the week should be explicitly labeled.

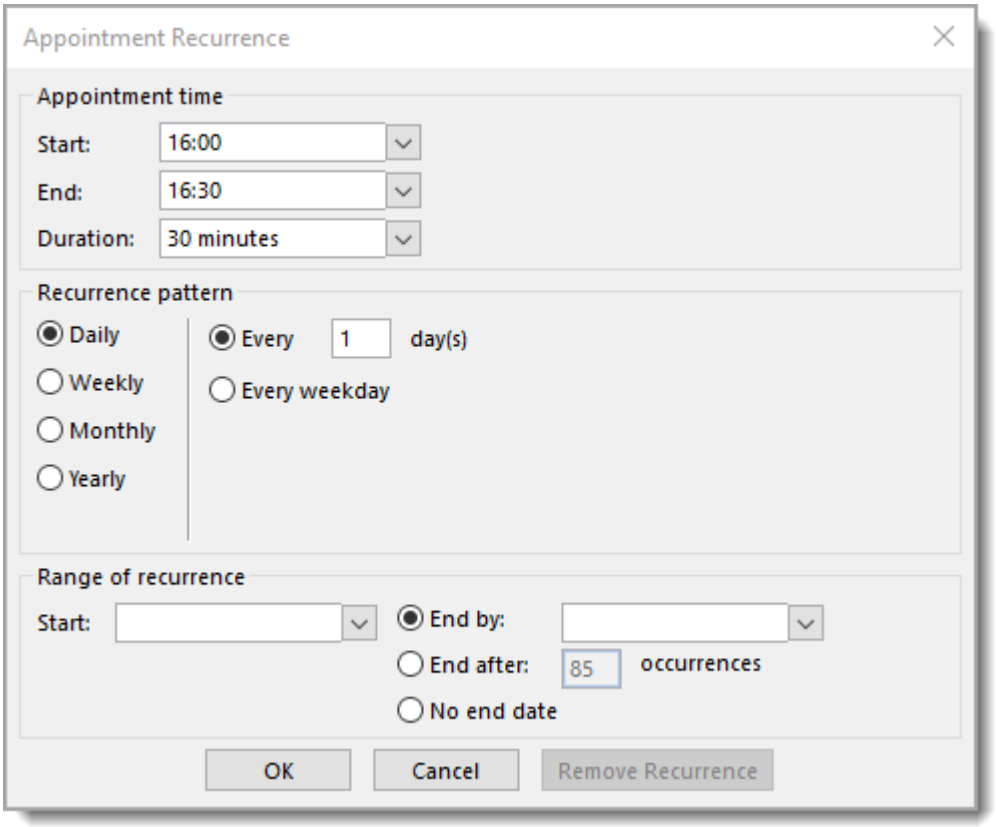

Figure 56: Complex form field (Outlook: New Items > Appointment > Make Recurring)

Accordion

Online betrachten

See also: Button, tabs, carousel

An accordion combines several areas which are arranged one beneath the other and which can be shown and hidden using buttons. In this respect, the labeling of the areas is visible at all times (see DIN EN ISO 9241-161: 8.1). The label usually has a visual indicator which refers to the status of the area (shown or hidden). For an accordion, different implementation variants are possible with respect to the minimum or maximum number of open areas, e.g.

- Initially, all the areas are closed and only one area can be opened at a time. If an area is opened, the previously opened area is automatically closed.
- At least one area is open at all times, and all areas can be opened.

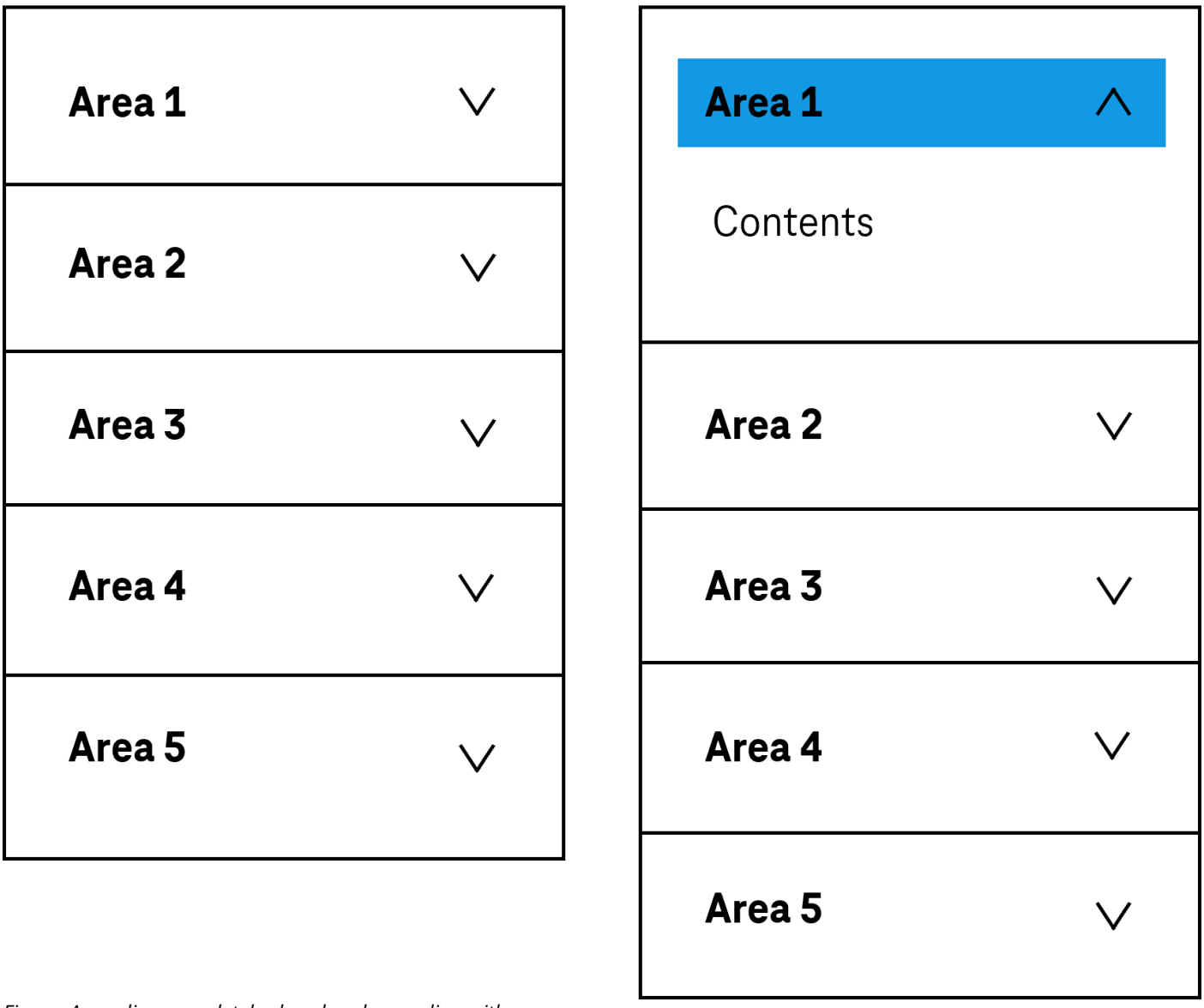

Figure: Accordion completely closed and accordion with an open area

Presentation

No.	Property	Description	Classification	Reference
827	Contrast	The labeling of the areas must have a contrast of at least 4.5:1.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
828	Contrast	The visual indicator in the respective labeling of an accordion area which refers to the status of the area (open or closed) must have a contrast ratio of at least 3:1 with respect to the background.	Must	EN 301 549: 11.1.4.1; EN 301 549: 11.1.4.11
829	Contrast	If the areas and the area labels are only identifiable as such on the basis of their color design, these colors must have a contrast ratio of at least 3:1 with respect to the neighboring colors. Note 1: The areas and area labels can be identifiable as such, for example, due to their borders or background colors. Note 2: The requirement does not apply if the areas and area labels are clearly identifiable as such, due to their position and spacing, for example.	Should	EN 301 549: 9.1.4.11, 11.1.4.11
830	Label	The label of the areas must be expressive (see Label).	Must	EN 301 549 9.2.4.6, 11.2.4.6
831	Focus visibility	If the area label receives the keyboard focus, the focus indicator must be visible (see Focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
832	Use of the keyboard	It must be possible to access, operate and exit the accordion with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
833	Use of the keyboard	The hidden areas and their contents must not receive keyboard focus.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
834	Updates	When focusing and operating the accordion, no unexpected change of context may occure.	Must	EN 301 549: 9.3.2.1, 11.3.2.1
835	Updates	When operating the buttons for showing and hiding the areas, no loss of focus loss may occur. Note: The focus must remain on the button or be placed at the beginning of the shown area.	Must	EN 301 549: 9.2.4.3, 11.2.4.3
836	Focus order	The focus order in the accordion must correspond to the visual presentation, i.e. the opened areas receive the keyboard focus immediately after the area label.	Must	EN 301 549: 9.2.4.3, 11.2.4.3
837	Click area	The click area of the area labels should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: accordion

Action	Key	Classification
Focusing of the area labels	TAB	Required
Exiting the area labels	TAB Note: If the corresponding area is closed, the next area label is focused using the TAB key. Otherwise, the focus is placed in the area.	Required
Operation of the area labels (opening and/or closing the area)	ENTER, SPACE	Required
Navigating between the area labels	UP/DOWN ARROW	Recommended
Quick navigation between the area labels	POS1, END	Recommended

Use of the pointing device: accordion

Action	Key	Classification
Operation of the area labels (opening and/or closing the area)	Left click	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
838	Role	For the area labeling, the button role must be communicated to the Accessibility API (siehe Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
839	Desktop: Element hierarchie	The parent / child relationships of the elements within the accordion must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9
840	Status	The status of the area labels must be communicated to the Accessibility API (see Element status). Note: This also applies to the „open“ or „closed“ status.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
841	Name	The buttons with the area labels must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5
842	Name	If the buttons with the area labels have a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
843	Operation	It must be possible to access, operate and exit the buttons with the area labels with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
844	Update	Updates concerning the Accessible Name, value or status of the area labels must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15
845	Desktop: Position	The size and position of the area labels and areas must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: accordion in Web applications

Screen reader output for the accordion buttons for showing and hiding the areas

- JAWS: [Label] **button** [collapsed | expanded] [note on operation with the Enter key]
- NVDA: [Label] **button** [collapsed | expanded]
- Windows narrator: [Label] **button** [collapsed | expanded]

Note: When using the <details>-element without a label, the beginning and end of the page area are not perceptible with assistive technology. If the <details>-element is explicitly labeled (by)aria-label or aria-labelledby) the area is output as „group“ (JAWS) or „grouping“ (NVDA). Die Windows Narrator does not output labeled groups either.

HTML

There is no element for accordions in HTML. Instead, several areas can be used which are shown and hidden using buttons. The areas are marked with <details> and the buttons with <summary>. The <summary> element is the first child element within <details>. The labels of the buttons are derived from the text content in the <summary> element. The initial state of the area (opened or closed) is set with the open attribute.

To make the coherence of the areas and buttons perceptible with assistive technology, they can be nested in a labeled group.

According to the HTML specification, the <summary> element may contain links, headings, input fields and many other elements – it is important to remember, however, that all the elements which are located within <summary> are imperceptible and cannot be operated with the screen reader, as the <summary> element is communicated to the Accessibility API as a button. Therefore, the <summary> element should only contain a concise and expressive label in text form.

Further information: 4.11.1 The details element - HTML Standard (whatwg.org) (External Link)
, 4.11.2 The summary element - HTML Standard (whatwg.org) (External Link)

ARIA

There is no role for accordions in ARIA. Instead, several areas can be used which are shown and hidden using Button. In this respect, the following should be taken into account:

- The buttons that have the purpose of showing and hiding areas should be marked with the aria-expanded attribute. aria-controls can be used to make reference to the ID of the area that is shown or hidden.
- The labeling of the buttons should take place using text content or aria-labelledby.
- To make the coherence of the areas and buttons perceptible with assistive technology, they can be nested in a labeled group.
- The presentation of the accordion should be verified in Windows High Contrast mode. The areas should therefore have a border.
- The visible button and the programmatically focused element should have the same position and size.

Alternatively, a tab group can be used instead of an accordion.

Further information: aria-expanded state - Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
, Accordion Pattern (Sections With Show/Hide Functionality) | APG | WAI | W3C

Date picker

Online betrachten

Synonyms: Calendar date picker, calendar, calendar element, calendar control, time selector, time picker, clock, date and time selector, date-time picker

See also: Combined input field, color picker

Note: All the following statements apply equally to the time picker or to control elements for the selection of weeks, months or years.

A date picker is used for the supported entry of a date (see DIN EN ISO 9241-161: 8.9 and 8.46).

Date pickers can be implemented in different ways, e.g. as

- input field for a date combined with a button, with which it is possible to open a dialog for the selection of a date,
- read-only input field for a date combined with a button with which it is possible to open a dialog for the selection of a date,
- combined input field for a date with which it is possible to open a dialog for the selection of a date,
- table for selecting a date within a specific time period and buttons for changing the time period,
- separate input fields for entering the day, month and year, combined with a button with which it is possible to open a dialog for the selection of a date,
- separate spin buttons for entering and selecting the day, month and year,
- seperate selection lists for selecting the day, month and year,
- seperate drop-down lists for selecting the day, month and year.

The dialog for selecting a date can contain the following elements:

- Tables, buttons, radio buttons, spin buttons, selection lists or drop-down lists for selecting a date within a specific period of time (e.g. days within a month, months within a year, years within a decade),
- Buttons for selecting another period,
- visual highlighting of the days and periods (e.g. „today“, „chosen day“, „holiday day“, „public holiday“, „weekend“, „day with free appointments“). The requirements for the individual control elements within the date picker are described for the respective control element. Only the additional requirements for the whole of the element are described here.

Examples:

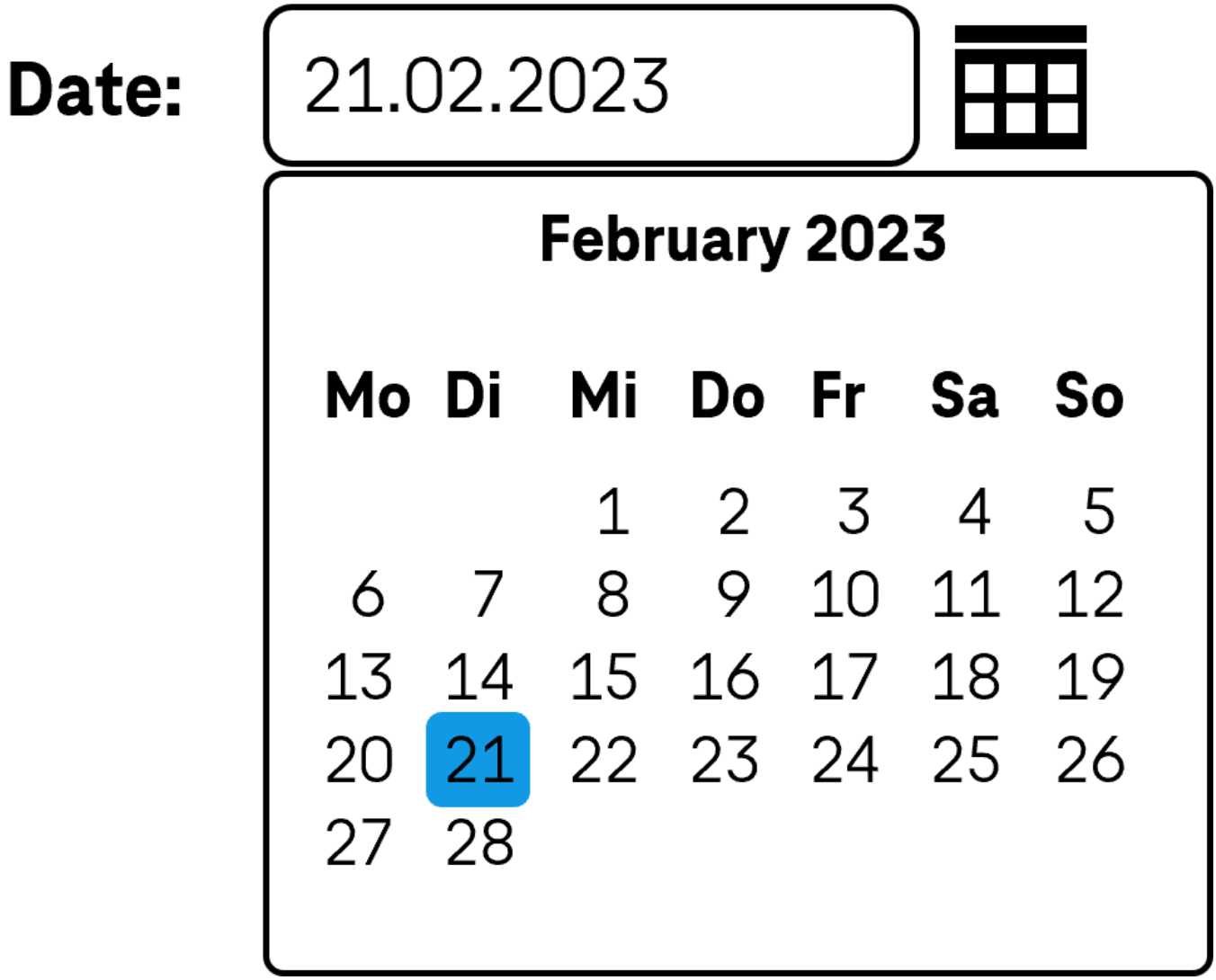

Figure: Date picker

Presentation

No.	Property	Description	Classification	Reference
846	Contrast	The visual highlighting of the days and time periods must have a contrast ratio of at least 3:1 between each other. Note: If this is not possible, an additional visual method must be used for the differentiation.	Must	EN 301 549: 11.1.4.1
847	Description	A legend or tooltip should explain the meaning of visual highlighting, unless it is self-explanatory. Note: The visual highlighting of weekends and public holidays is self-explanatory, for example.	Should	DIN EN ISO 9241-143: 9.6.11

Operation

No.	Property	Description	Classification	Reference
848	Use of the keyboard	It must be possible to access, operate and exit the date picker and the elements that it contains with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
849	Updates	When focusing and operating the date picker, no unexpected change of context may occure .	Must	EN 301 549: 9.3.2.1, 11.3.2.1, 9.3.2.2, 11.3.2.2
850	Error message	An entered date may not be corrected automatically without an error message which must be provided in text form. Note: This does not include changes to the input, unless these change the date (such as the automatic insertion of leading zeros)	Must	EN 301 549: 9.3.3.1, 11.3.3.1
851	focus order	The focus order in the date picker should correspond to the visual presentation.	Should	DIN EN ISO 9241-171: 9.3.18
852	Focus order	If the date picker outside a dialog contains several control elements that receive the focus with TAB, it should be possible to skip the date picker with the keyboard (see navigtion sequence)	Should	DIN EN ISO 9241-171: 9.3.17
853	Click area	The click area of the control elements for the date picker should be at least 24 x 24 px (see Use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: date picker

Action	Key	Classification
Focusing of the date picker	TAB	Required
Exiting the date picker	TAB Note: There can be several control elements within the date picker that have been previously focused with TAB.	Required
Navigating within the date picker	TAB or LEFT/RIGHT/UP/DOWN ARROW (depending on the used control element)	Required
Quick navigation within the selection elements for a day, a month or a year	POS1, END, PAGE UP/DOWN	Recommended
Activation of the control elements in the date picker	ENTER or SPACE (depending on the used control element)	Required

Use of the pointing device: date picker

Action	Key	Classification
Activation of the control elements in the date picker	left click	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
854	Role	The role of the date picker and its control elements must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
854	Desktop: Element hierarchy	The parent / child relationships of the elements within the date picker must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9

No.	Property	Description	Classification	Reference
855	Status	The status of the date picker and its control elements must be communicated to the Accessibility API (see Element status). Note 1: This refers, for example, to the „open“ or „closed“ status with respect to the dialog, and to the „selected“ status with respect to the selected date within one of the selection elements. Note 2: Insofar as the purpose of the visual highlighting (such as public holiday, holiday) cannot be communicated as the status, this information should be communicated as part of the description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
856	Name	The date picker and its control elements must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
857	Name	If the date picker or its control elements have a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
858	Operation	It must be possible to access, operate and exit the date picker and its control elements with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
859	Update	Updates concerning the Accessible Name or status of the date picker and its control elements must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15

No.	Property	Description	Classification	Reference
860	Position	The size and position of the date picker and its control elements must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.5, 11.5.2.15

Color picker

Online betrachten

Synonyms: Color picker

See also: Slider, Date picker The color picker serves the selection of a color (see DIN EN ISO 9241-161: 8.6). Color pickers can be implemented in different ways, e.g. as

- Table, Drop down list or Selection list for selecting a color,
- single input field for entering a color value (e.g. HEX value),
- several input fields or spin buttons for entering single color channels (e.g. RGB, RGBA, CMYK or HSB),
- two-dimensional gradient field for selecting a value,
- individual slider for selecting a color,
- several sliders for selecting individual color channels (e.g. RGB, RGBA, CMYK or HSB),
- button, with which a dialog can be opened for selecting a color (the dialog contains one of the above-mentioned elements),
- a combination of the above-mentioned elements.

The requirements for the individual control elements within the color picker are described for the respective control element. Only the additional requirements for the whole of the element are described here.

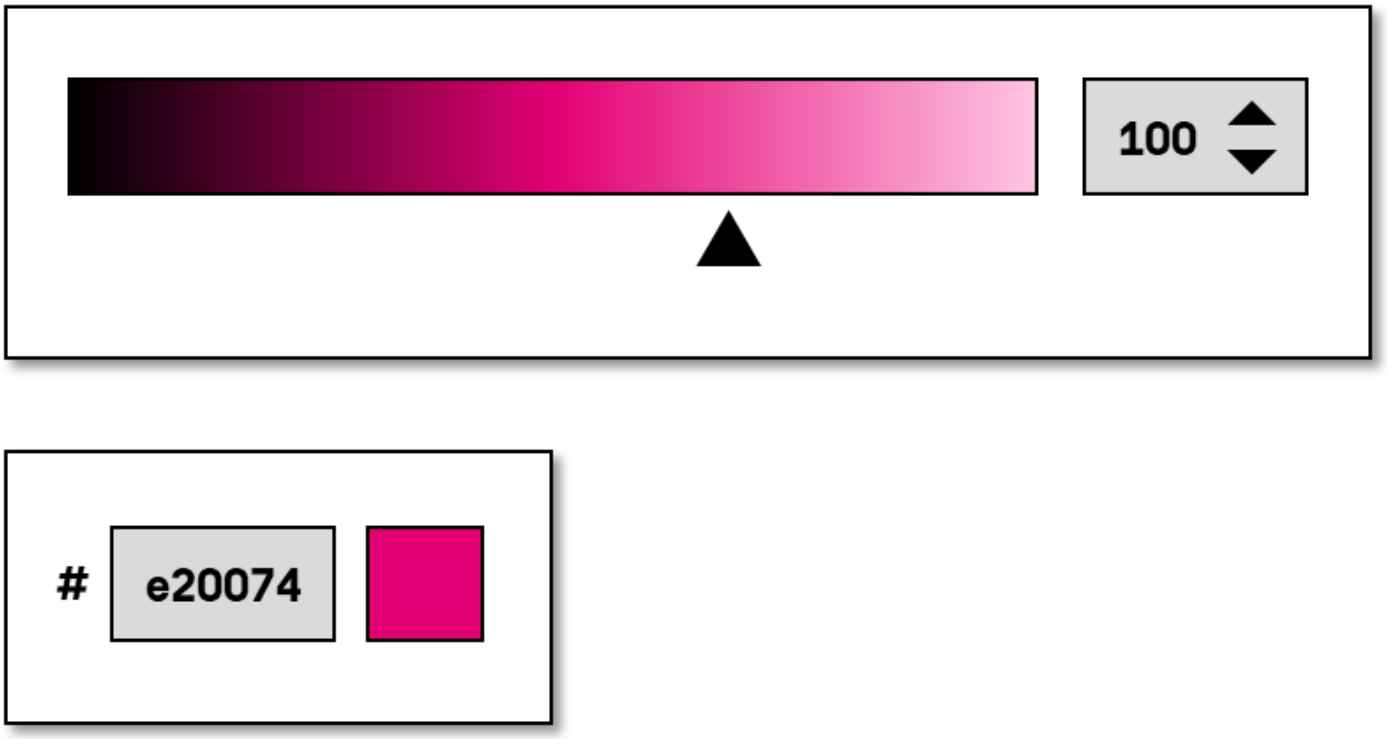

Figure: Color picker

Presentation

No.	Property	Description	Classification	Reference
861	Color coding	The color picker must contain at least one option for selecting a color by its color name, color code (such as HEX), or values of the color channels (such as RGB).	Must	EN 301 549: 9.1.4.1, 11.1.4.1
862	Color coding	If the selectable values or the current value are visually displayed in color, the elements should have a tooltip with the color name or the color value in text form. Note: The output of the color name should preferably be used.	Should	DIN EN ISO 9241-143: 9.6.11

Operation

No.	Property	Description	Classification	Reference
863	Use of the keyboard	It must be possible to access, operate and exit the color picker and the elements that it contains with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
864	Update	When focusing and operating the color picker, no unexpected change of context may occure.	Must	EN 301 549: 9.3.2.1, 11.3.2.1 und 9.3.2.2, 11.3.2.2
865	Focus order	The focus order in the color picker should correspond to the visual presentation.	Should	DIN EN ISO 9241-171: 9.3.18
866	Focus order	If the color picker outside a dialog contains several control elements that receive the focus with TAB, it should be possible to skip the color picker with the keyboard (see navigation sequence)	Should	DIN EN ISO 9241-171: 9.3.17
867	Click area	The click area of the control elements for the color picker should be at least 24 x 24 px (see use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: color picker

Action	Key	Classification
Focusing of the color picker	TAB	Required
Exiting the color picker	TAB Note: There can be several control elements within the color picker that have been previously focused with TAB.	Required
Navigating within the color picker	TAB or LEFT/RIGHT/UP/DOWN ARROW (depending on the used control element)	Required
Activation of the control elements in the color picker	ENTER or SPACE (depending on the used control element)	Required

Use of the pointing device: color picker

Action	Key	Classification
Activation of the control elements in the color picker	left click	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
868	Role	The role of the color picker and its control elements must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
869	Desktop: Element hierarchy	The parent / child relationships of the elements within the color picker must be communicated to the Accessibility API.	Must	EN 301 549: 11.5.2.9

No.	Property	Description	Classification	Reference
870	Status, value	The value and status of the color picker and its control elements must be communicated to the Accessibility API (see Element status). Note 1: Status refers, for example, to the „open“ or „closed “ status with respect to the dialog, and to the „selected“ status with respect to the selected color within one of the selection elements. Note 2: Value refers, for example, to the selected color or the value of a color channel. Color names are to be preferred over color values.	Must	EN 301 549: 11.4.1.2, 11.5.2.5
871	Name	The color picker and its control elements must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
872	Name	If the color picker or its control elements have a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
873	Operation	It must be possible to access, operate and exit the color picker and its control elements with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
874	Update	Updates concerning the Accessible Name or status of the color picker and its control elements must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15

No.	Property	Description	Classification	Reference
875	Desktop: Position	The size and position of the color picker and its control elements must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 11.5.2.5, 11.5.2.13

Practical tip: color picker in Web applications

HTML

The color picker can be implemented with the HTML element <input type=color>.

The initial value is communicated through the value attribute. Only the hexadecimal color values starting with a diamond and followed by 6 characters are permitted as values.

The label of the color picker should be linked to the color picker with the <label for=ID> element.

The color picker can be marked as disabled, but not as a required field (required) or as read-only (readonly).

Please note: Depending on the browser and screen reader used, the HTML color picker is either not perceptible at all (e.g. JAWS with Chrome and Edge) or only partially perceptible and operable (e.g. NVDA with Chrome, Edge and Firefox). Using the HTML color picker is not recommended, unless the application is only intended to work with a browser and specific assistive technology, and accessibility can be guaranteed in this environment. Otherwise, depending on the requirement, for instance, a button, which opens a modal dialog for the color selection, one or several input fields, a drop-down list, slider or a combination of these elements can be used.

Further information: 4.10.5.1.14 Color state (type=color) - HTML Standard (whatwg.org)

ARIA

There is no role for color pickers in ARIA. Instead, depending on the requirement, for instance, a button which opens a modal dialog for the color selection, one or several input fields, a drop-down list, slider or a combination of these elements can be used.

Carousel

Online betrachten

Synonyms: Slide show, diashow, image rotator, cover flow, element flow, slider

See also: Accordion, Tabs

A carousel is used for the grouping of blocks of content that can be scrolled through (see DIN EN ISO 9241-161: 8.3).

Several implementation variants are possible, including:

- Only one content block is visible at a time, several content blocks are visible, or all content blocks are visible.
- The visible content blocks are completely visible, conceal each other or are truncated at the edge of the screen.
- The content blocks are arranged vertically or horizontally side by side, in a circle, or one above the other.
- All visible content blocks are displayed in the same way or a central content block is highlighted, while the content blocks are displayed adjacently in smaller form or grayed out.
- It is possible to scroll through the content blocks automatically or this can be done by the user.
- The scrolling through the content blocks takes place using a scrollbar, button for scrolling back and forth, or with a pagination.
- The content blocks may contain graphical elements, text, and control elements.

Examples:

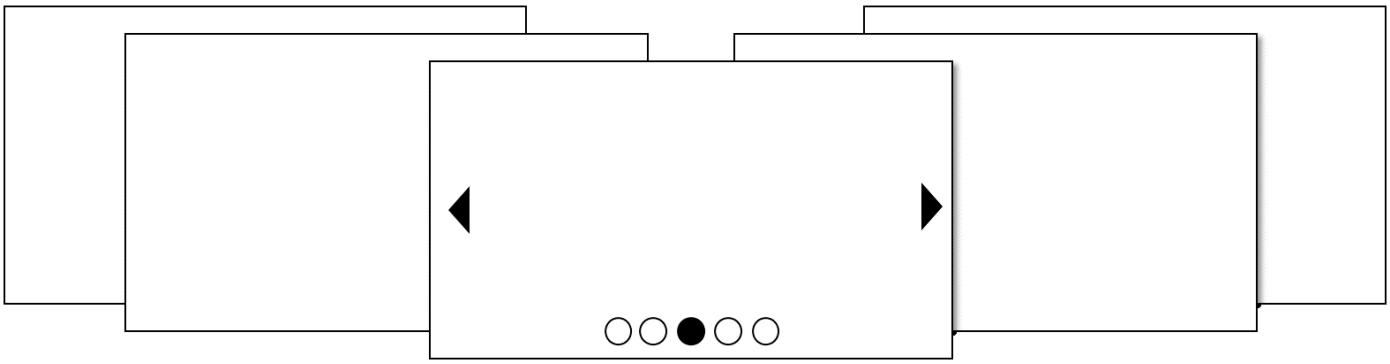

Figure: Carousel

Presentation

No.	Property	Description	Classification	Reference
876	Contrast	If the control elements of the carousel are labeled with text, they must have a contrast ratio of at least 4.5:1.	Must	EN 301 549: 9.1.4.3, 11.1.4.3
877	Contrast	If the control elements of the carousel are labeled with graphics, they must have a contrast ratio of at least 3:1.	Must	EN 301 549: 9.1.4.11, 11.1.4.11
878	Contrast	If, in terms of the pagination, the current page only differs from the rest of the pages due to its color, the colors must have a contrast ratio of at least 3:1.	Must	EN 301 549: 9.1.4.1, 11.1.4.1
879	Animation	If the content blocks in the carousel are scrolled through automatically, a mechanism must be implemented to stop the automatic scrolling and it must then be possible to scroll through the content blocks manually.	Must	EN 301 549: 9.2.2.1, 11.2.2.1, 9.2.2.2, 11.2.2.2
880	Animation	If moving animations are displayed during the operation of the carousel, a mechanism should be available to disable them(see) animations). Note: This applies to scrolling through the content blocks, for example..	Should	WCAG 2.1.: 2.3.3 (AAA)
881	Focus visibility	If an element in the carousel receives the keyboard focus, the focus indicator must be visible(see focus indicator).	Must	EN 301 549: 9.2.4.7, 11.2.4.7

Operation

No.	Property	Description	Classification	Reference
882	Use of the keyboard	It must be possible to access, operate and exit the carousel and the elements that it contains with the keyboard (see Use of the keyboard table, below).	Must	EN 301 549: 9.2.1.1, 11.2.1.1, 9.2.1.2, 11.2.1.2
883	Use of the keyboard	The hidden content blocks and their contents must not receive keyboard focus.	Must	EN 301 549: 9.1.3.1, 11.1.3.1
884	Updates	During the focusing of the control elements of the carousel, no change of context may occur.	Must	EN 301 549: 9.3.2.1, 11.3.2.1
884	Updates	When operating the carousel, no loss of focus may occur. Note: The focus must remain on the respective control element or be placed at the beginning of the shown area.	Must	EN 301 549: 9.2.4.3, 11.2.4.3
885	Focus order	The focus order in the carousel should be appropriate to the work task.	Should	DIN EN ISO 9241-171: 9.3.18
886	Focus order	If the carousel contains several control elements that receive the focus with TAB, it should be possible to skip the carousel with the keyboard (see navigation sequence)	Should	DIN EN ISO 9241-171: 9.3.17
887	Click area	The click area of the control elements of the carousel should be at least 24 x 24 px (see use of the pointing device).	Should	WCAG 2.2

Use of the keyboard: carousel

Action	Key	Classification
Focusing of the first element in the carousel	TAB	Required
Exiting the carousel	TAB Note: There can be several control elements within the carousel that have been previously focused with TAB.	Required
Navigating within the carousel	TAB or LEFT/RIGHT/UP/DOWN ARROW (depending on the used control element)	Required
Quick navigation between the content blocks	POS1, END, PAGE UP/DOWN	Recommended
Activating of the control elements in the carousel	ENTER or SPACE (depending on the used control element)	Required

Use of the pointing device: carousel

Action	Key	Classification
Enabling of the control elements in the carousel	left click	Required

Programming/interfaces

No.	Property	Description	Classification	Reference
888	Role	The role of the control elements in the carousel must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5

No.	Property	Description	Classification	Reference
		Note: If the carousel role is not known in the programming language used, the entire carousel should be located in a labeled group. The label or description of the group should contain a note about the element type of the carousel.		
889	Status	The status of the content blocks and control elements in the carousel must be communicated to the Accessibility API (see Element status). Note: This also applies to the “current content block” status in the carousel or the pagination.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
890	Name	The control elements in the carousel must have a concise and expressive Accessible Name.	Must	EN 301 549: 9.2.4.6, 11.2.4.6, 9.4.1.2, 11.4.1.2, 11.5.2.5, 11.5.2.8
891	Name	If the control elements in the carousel have a description, it must be communicated as an Accessible Description.	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.5
891	Operation	It must be possible to access, operate and exit the carousel and its control elements with assistive technology (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.12, 11.5.2.17
892	Update	Updates concerning the Accessible Name or status of the content blocks and the control elements in the carousel must be communicated to the Accessibility API (see Accessibility API).	Must	EN 301 549: 9.4.1.2, 11.4.1.2, 11.5.2.15

No.	Property	Description	Classification	Reference
893	Desktop: Position	The size and position of the content blocks and the control elements in the carousel must be communicated to the Accessibility API (see Focus indicator).	Must	EN 301 549: 11.5.2.5, 11.2.5.13

Annex

Online betrachten

Table of contents

- References and sources
- Glossary
- Index

References and sources

Online betrachten

- Accessibility - Windows apps | Microsoft Learn (External Link)
- Accessible Name and Description Computation 1.2 (w3.org) (External Link)
- Accessible Rich Internet Applications (WAI-ARIA) 1.2 (w3.org) (External Link)
- ARIA Authoring Practices Guide | APG | WAI | W3C (External Link)
- Authoring Tool Accessibility Guidelines (ATAG) 2.0 (w3.org) (External Link)
- BITV 2.0 - Verordnung zur Schaffung barrierefreier Informationstechnik nach dem Behindertengleichstellungsgesetz (External Link)
- Core Accessibility API Mappings 1.2 (w3.org) (External Link)
- DIN EN ISO 9241-125:2018 - Empfehlungen zur visuellen Informationsdarstellung
- DIN EN ISO 9241-171:2008 - Leitlinien für die Zugänglichkeit von Software
- DIN EN ISO 9241-143:2012 - Formulardialoge
- DIN EN ISO 9241-161:2016 - Leitfaden zu visuellen User-Interface-Elementen
- ETSI EN 301 549 - V3.2.1 - Accessibility requirements for ICT products and services (öffnet PDF) (External Link)
- Guidance on Applying WCAG 2.0 to Non-Web Information and Communications Technologies (WCAG2ICT) (w3.org) (External Link)
- IUIAutomation (uiautomationclient.h) - Win32 apps | Microsoft Learn (External Link)
- Object Roles (Oleacc.h) - Win32 apps | Microsoft Learn (External Link)
- Object State Constants (Oleacc.h) - Win32 apps | Microsoft Learn (External Link)
- Revised 508 Standards and 255 Guidelines (access-board.gov) (External Link)
- User Agent Accessibility Guidelines (UAAG) 2.0 (w3.org) (External Link)
- Web Content Accessibility Guidelines (WCAG) 2.1 (w3.org) (External Link)
- Web Content Accessibility Guidelines (WCAG) 2.2 (w3.org) (External Link)

Glossary

Online betrachten

Accessibility

The extent to which products, systems, services, environments and facilities can be used by people from a population with the widest scope of use-related needs, characteristics and skills to be able to achieve identified objectives in identified use contexts (EN 301 549 v3.2.1:2021)

Accessible methods of access include construction and other facilities, methods of transport, technical objects, information processing systems, acoustic and visual information sources and communication devices and other designed areas of life, if they can be found, accessed and used by persons with disabilities in the usual way, without any particular difficulties and in general without assistance from others. In this respect, the use of necessary auxiliaries due to disability is permitted (Section 4 of the BGG).

Accessible Description

Programmatically interpretable characteristic of a UI element that has the purpose of the more detailed description of the UI element for assistive technology

Accessible Name

Name; Programmatically interpretable characteristic of a UI element that has the purpose of the labeling of the UI element for assistive technology

Area navigation

The ability to move from one UI element group to another with the use of the keyboard

Assistive Technology

AT; Product, system, hardware or software which is used to enhance, maintain or improve the functional skills of people (EN 301 549 v3.2.1:2021)

Examples: Screen magnifier, screen reader, text-to-speech software, speech recognition software, alternative keyboards and pointing devices (assistive technology - Web Content Accessibility Guidelines (WCAG) 2.2 (w3.org) (External Link)
)(w3.org))

Authoring Tool

Software which can be used to create or edit content (EN 301 549 v3.2.1: 2021)

Captcha

„“Completely Automated Public Turing test to tell Computers and Humans Apart” Program that protects websites from bots by generating and evaluating tests that people are able to pass but current computer programs are not (Official CAPTCHA page (External Link)
)

Click area

The area of the user interface which is activated by a pointer device (e.g. a mouse) (DIN EN ISO 9241-161: 2016)

Closed functionality

Functionality which is restricted by specific features that prevent the connection, installation or use of assistive technology (EN 301 549 v3.2.1:2021)

Color scheme

Series of color assignments for the presentation of UI elements (DIN EN ISO 9241-171)

Context-sensitive Help

Help text that contains information about the function which is currently being run

Note: Unique labels can be used as context-sensitive help.

Examples:

- Link to the Help option
- Digital input assistant
- Spell check
- Word suggestions during text input
- Suggestion of alternative words in the case of incorrect entries
- Tooltips context-sensitive help - Web Content Accessibility Guidelines (WCAG) 2.1 (w3.org) (External Link)

Continuous text

Also scrolling text or text block; continuous text of an article without a heading, table or equivalent; Text with more than one sentence (WCAG 2.1, Understanding SC 1.4.8)

Contrast ratio

Evaluation of the difference between two immediately adjacent or sequential impressions (luminance contrast, brightness contrast, color contrast, etc.)

Here: Measurement value for the presentation of the maximum relative brightness differences between two colors

Control element

Control, operating element;

UI element with which users are able to interact, such as the keyboard or a pointing device

Examples: Links, buttons, form fields

Custom element

UI element which, in contrast to defined UI elements of the programming language (standard elements), can be created by development teams with full functionality

Description

The detailed labeling for a UI element (input field, display field, a table, a control element or an object)

ENTER key

This refers to both the RETURN and ENTER keys on the numeric keypad

Explicit identifier

Code or abbreviation for a menu item or the label of a control element (usually on the left) which is highlighted next to the name and entered when selected (DIN EN ISO 9241-171:2008)

Flash

A series of two opposing changes in relative luminance. If the changes are big enough and occur at the correct frequency, they pose the risk of triggering seizures in some people. (general flash and red flash thresholds - Web Content Accessibility Guidelines (WCAG) 2.1 (w3.org) (External Link)
)

Focus indicator

Position cursor;

Display that shows which UI element has the keyboard focus (DIN EN ISO 9241-171:2008) For example: Keyboard focus indicator: visual display of where the user interaction with the keyboard (or keyboard emulator) will take place (DIN EN ISO 9241-171:2008)

Form

structured presentation of fields and other UI elements that users read, fill out, for which they select or modify entries (DIN EN ISO 9241-161:2016)

Form element

UI element which is used for entering or selecting values in form dialogs (also buttons)

Hard line break

fixed end-of-line marking which is interpreted as a paragraph (pilcrow, symbol: ¶) (automatic line break – Wikipedia)

Hover area

The area of the user interface which responds to an overlying pointer (DIN EN ISO 9241-161:2016)

Hybrid application

Desktop application that uses Web technologies for the presentation of the user interface

Image of text

Text which is rendered in a non-text form (e.g., an image) to achieve a particular visual effect (image of text - Web Content Accessibility Guidelines (WCAG) 2.1 (w3.org) (External Link)
)

Implicit identifier

The part of an option name or the label of a control element which is used for a keyboard selection (DIN EN ISO 9241-171:2008)

For example „Print“

Keyboard

here: Keyboard interface;

Interface which is used by the software for the receiving of keystrokes (keyboard interface - Web Content Accessibility Guidelines (WCAG) 2.1 (w3.org) (External Link)
)

Keyboard focus

current assignment of the input made on a keyboard or keyboard equivalent to a UI element (DIN EN ISO 9241-171:2008)

Keyboard navigation

The ability to move from one UI element to another within an interactive system with the use of the

Keyboard shortcut

Key shortcut, key combination, keyboard combination, keyboard command, keyboard equivalents, key sequence, access key, hot key, shortcut

Keys and key combinations which provide access to functions that are usually activated by means of a pointing device, voice input, or through other input or control mechanisms

Label

A short designation or heading for a UI element (for example, an input field or display field, a table, a control element or an object) (see also DIN EN ISO 9241-171: 2008)

Navigation

The ability to move from one UI element to another within a single user interface and to move within an interactive system (DIN EN ISO 9241-161:2016)

Non-text content

Content which is not a sequence of letters determinable by software, or in the case of which the sequence does not express anything in human language (EN 301 549 v3.2.1:2021)

Examples:

- Captcha
- Sound
- Graphic
- Vibration
- ASCII-Art

Open functionality

Functionality which supports the access through assistive technology (EN 301 549 v3.2.1:2021)

Platform software

A collection of software components which is run on an underlying software or hardware layer, and which provides a set of software services to other software components through which these applications can be isolated from the underlying software or hardware layer (EN 301 549 v3.2.1)

Examples: An operating system, device drivers, window systems and software toolkits (DIN EN ISO 9241-161)

Note: A browser can function both as an application and as platform software. (DIN EN ISO 9241-161)

Pointer

Graphical symbol whose position on the screen is changed according to the movement of a pointing device, and whose shape can be adjusted depending on the control element which is situated beneath it (DIN EN ISO 9241-171:2008)

Pointing device

Pointing instrument;

An auxiliary which converts an operational step of the user into a step that is displayed on the screen

Note: Depending on the technology used, both mechanical auxiliary products and parts of the human body (e.g. fingers, arms) can be used as pointing devices. (DIN EN ISO 9241-171:2008)

Quick navigation

Keyboard navigation in which navigation steps are skipped to enable the efficient operation

Real-time event

An event that a) occurs at the same time as it is viewed, and b) is not entirely generated by the content

For example: “Web cast” of a live performance, online auction (real-time event (Web Content Accessibility Guidelines (WCAG) 2.0) (w3.org))

Role

UI element type for user interfaces;

Characteristic which serves as a known identifier, indicating the type of UI element;

Client applications, especially assistive technology, use the characteristic in order to identify the functions of a control element and to determine how to interact with it

Screen magnifier

Assistive technology for the individualizing of the visual display

Frequent functions are:

- Zoom
- Color adjustment
- Highlighting of the focus indicator, text cursor and mouse pointer
- Reading out of text
- Linearized presentation of text with freely selectable font styles/sizes

Screen reader

Assistive technology which enables users to use software without having to see the visual display (DIN EN ISO 9241-171:2008)

The UI elements are output acoustically by speakers or headphones and on a tactile basis with the use of a Braille display.

The input and control takes place with the keyboard or the Braille display. The inputs are initially processed by the screen reader before being forwarded to the user interface.

Shortcut key

Mnemonic, menu accelerator, accelerator key, shortcut keys

Keys and combinations of keys for initializing a menu option without displaying the corresponding menu with the option or intermediate menus on the screen

For example: Save (ALT + S)

- S = shortcut key – only works when the menu is open and has the focus
- ALT + S = keyboard shortcut – always works (unless the function is temporarily disabled)

Skip link

An internal page link to skip areas of content during the keyboard navigation

Speech recognition software

Assistive technology for the input of text (dictation) or control commands for the running of control elements

An alternative method of input for the mouse or keyboard interface

Standard element

UI elements that are defined by the programming language by default

Status

State;

dynamic characteristic which expresses the characteristics of a UI element which is able to change in response to actions by the user or automated processes

The status does not affect the nature of the UI element, but represents data which is assigned to the component or user interaction capabilities

Examples: focused, selected, pressed, highlighted, operable/disabled, correct/incorrect and open/closed, read-only

Status message

Change in content which is not a change in context, and provides users with information about the success or outcome of an action, the waiting state of an application, the progress of a process, or the presence of errors (status message - Web Content Accessibility Guidelines (WCAG) 2.2 (w3.org) (External Link)
)

Support services

Support services include, but are not limited to: Help desks, call centers, technical support, mediation services and training (see also EN 301 549: 12.2.1)

Text cursor

Text indicator;

visual display of the current cursor for the text input (DIN EN ISO 9241-171:2008)

UI

All components of an interactive system (software or hardware) that provide users with information and control elements in order to be able to perform specific work tasks with the interactive system (DIN EN ISO 9241-171:2008)

UI element

User interface element

User interface element; elementary component of the user interface which is displayed or otherwise presented to the users by the software (DIN EN ISO 9241-171:2008)

User agent

Software which retrieves and displays content for users (EN 301 549 v3.2.1:2021)

User preferences

Individualization, customization;

The modification of the interaction and the presentation of information in order to respond to the individual abilities and requirements of the user (DIN EN ISO 9241-171:2008)

Value

Form elements have a value which is communicated when the form is submitted. In an input field, the value is the entered text. In a selection list, the value is the selected option.

Virtual cursor

The virtual cursor is a mode of the screen reader. It is used, for example, in order to read Web pages in a Web browser (unless they are marked with role=application), PDF documents in a PDF reader, or content in hybrid desktop software. Although it is not visible like the mouse cursor, the virtual cursor simulates an insertion point, and offers the same functionality as when reading a text-based document.

Index

Online betrachten

A

- Abbreviations
- Access key
- Accessibility API
- Accessibility interface
- Accessible Description
- Accessible Name
- Accordion
- Adjustments
- Alternative version
- Analog form element
- Animations
- Application language
- Application window
- Authentication
- Auto-suggest box
- Autocomplete

B

- Background graphic
- Biometric data
- Button, see also:
  - Accordion
  - Split button
  - Spin button
  - Menu button
  - Toggle switch
  - Switch

C

- Calendar control
- Calendar element
- Captcha
- Carousel
- Change of context
- Change of language
- Changes of context
- Characters
- Check box
- Click area
- Closed (status)
- Color
- Color coding

- Colorpicker
- Colors
- Combo box
- Combobox, see also:
    - Drop-down list
    - Combined input field
- Command bar
- Command button
- Compatibility with assistive technology
- Conforming alternate version
- Conformity with expectations
- Consistency
- Context menu
- Context-specific help
- Continuous text
- Contrast
- Contrast adjustment
- Contrasts
- Control panel
- Control point
- Cover flow
- Cross reference
- Cursor

D

- Data entry
- Date picker
- Decorative graphic
- Description
- Designation
- Dialog
- Dialog box
- Dialog window
- Diashow
- Disabled (Status)
- Display (status)
- Display-only (Status)
- Documentation
- Drag & drop, see:
    - Use of the keyboard
    - Use of the pointing device
- Drag point
- drop-down list
- Drop-down list field

E

- Edge smoothing
- Edit

- Efficient navigation
- Element flow
- Element hierarchy
- Element status
- Element type
- Error correction
- Error message
- Error messages
- Error prevention

F

- Facial recognition
- Figure
- Flash
- Flashing
- Focus
- Focus appearance
- Focus frame
- Focus indicator
- Focus order
- Folding list
- Font
- Font size adjustment
- Font style
- Footer
- Foreign-language content
- Form, see also:
    - Control elements
    - Error prevention
    - Required field label
    - Composite form fieldsr
    - Form field group
- Form field label

G

- Graphic
- Grid
- Grid table
- Group
- Grouping

H

- Handle
- Heading
- Help, see also:
    - Description
    - Error prevention
- Hierarchical list

- Hierarchical table
- High Contrast Mode
- Hotkey
- Hyperlink
- Hyphenation

I

- Icon
- Image
- Image Rotator
- Inactive (status)
- Individualization
- Input field
- Input field, see:
  - Input field (one-line)
  - Input field (multi-line)
  - Input field with autocomplete function
  - Spin button
  - Password input field
  - Combined input field
- Input instructions, see:
  - Description
  - Error prevention
- Instruction
- Interoperability with assistive technology

K

- Key combination
- Keyboard combination
- Keyboard command
- Keyboard operation
- Keyboard shortcut
- Keyboard trap

L

- Label
- Language
- Layout graphic
- Letters
- Link
- List
- List box
- List button
- List field
- Login
- Logout

M

- Main heading
- Manual
- Menu, see also:
  - Context menu
  - Menu button
- Menu accelerator
- Menu bar
- Menu button
- Mnemonic
- Modal dialog
- Modifier key
- Multiple selection list

N

- Name
- Navigation, see:
  - Link
  - Menu
  - Navigation sequence
  - Pagination
- Note
- Numbers

O

- Opened (status)
- Operating instruction
- Operating system settings
- Option fields
- Overlay tool

P

- Page navigation
- Pagination
- Parent-child relationships
- Password
- Password field
- Password input field
- Pen operation
- Pixel graphic
- Platform settings
- Platform support of accessibility services for assistive technology
- Pointer operation
- Pointing device operation
- Pop-up
- Pop-up menu
- Pressed (status)

- Programming interface for assistive technology
- Progress bar
- Progress display
- Push button

Q

- Quick key

R

- Radio buttons
- Range control
- Range slider
- Read mode (status)
- Readonly (status)
- Reference
- Required (status)
- Required field label
- Resizing
- Role

S

- Scaling
- Scroll bar
- Scroll navigation
- Section heading
- Section separator
- Selected (status)
- Selection (status)
- Selection list, see also:
  - Drop-down list
  - Autocomplete
  - tree structure
  - Combined input field
  - multiple selection list
- selection switch
- Separator
- Shortcut
- Shortcut keys
- Slide show
- Slider
- Slider, see:
  - Carousel
  - Slider
- Special character
- Spin button
- Spinner
- Split button
- Splitter

- Status, see:
  - Element status
  - Error status
  - Graphically communicated status
  - Required field label
  - Status regarding operability
- Status bar
- Status information
- Structure, see:
  - Group
  - Structural elements
  - Heading
- Structure view
- Suggestion list
- Support
- Switch
- Switch button
- Symbols bar

T

- Tab
- Tab group
- Tab order
- Tab panel
- Table
- Table with tree structure
- Target size
- Text, see also:
  - Font
  - Text elements
- Text area
- Text box
- Text cursor
- Text field
- Text formatting
- Text spacing
- Time limit
- Time limits
- Time picker
- Timeout
- Title
- Title bar
- Toggle button
- Toggle switch
- Toggle switch
- Toolbar
- Toolbox
- Tooltip, see also:
  - Description

    - Use of the pointing device
- Touch operation
- Tree
- Tree grid
- Tree structure
- Tree table
- Tree view

U

- Unicode
- Updates, see:
    - Animations
    - Changes of context
- Use of the keyboard, see also:
    - Navigation sequence
    - Keyboard shortcut
    - Focus indicator
- Use of the mouse
- Use of the pointing device
- User preferences
- User settings

W

- Window
- Window splitter
- Windows contrast adjustment
- Write-protected (status)

V

- Vector graphic

Z

- Zoom

Informationen zu diesem Dokument

Diese Handreichung hat die Version 1.0 und wurde am 23.11.2026 erstellt.

Allgemeine Informationspflichten gemäß § 5 Telemediengesetz und § 55 Rundfunkstaatsvertrag

Die Deutsche Rentenversicherung Knappschaft-Bahn-See ist eine rechtsfähige Körperschaft des öffentlichen Rechts mit Selbstverwaltung und besitzt Dienstherrnfähigkeit (§ 29 SGB IV in Verbindung mit § 143 Abs. 1 SGB VI).

Dieses Impressum gilt für dieses Dokument der Arbeitsgruppen des Ausschusses für barrierefreie Informationstechnik nach § 5 BITV 2.0. Die Arbeitsgruppen werden von der Überwachungsstelle des Bundes für Barrierefreiheit von Informationstechnik organisiert.

Herausgeber

Deutsche Rentenversicherung Knappschaft-Bahn-See
Pieperstraße 14 - 28
44789 Bochum
Tel. 0234 304 - 0
Fax 0234 304 - 66050
E-Mail an die Zentrale der KBS: zentrale@kbs.de

Umsatzsteuer-Identifikationsnummer: DE 124089627

Dieses Dokument wird herausgegeben von der Deutschen Rentenversicherung Knappschaft-Bahn-See, vertreten durch die Geschäftsführung, Dr. Rainer Wilhelm.

Zuständige Fachaufsichtsbehörde für die Überwachungsstelle des Bundes für Barrierefreiheit von Informationstechnik

Bundesministerium für Arbeit und Soziales
Wilhelmstraße 49
10117 Berlin

Nutzungsbedingungen

Die Inhalte dieser Handreichung werden mit größtmöglicher Sorgfalt verfasst. Unser Anspruch ist es, richtige, vollständige und aktuelle Inhalte bereitzustellen. Wir übernehmen dennoch keine Gewähr für versehentlich gemachte falsche Angaben.

Diese Handreichung enthält Verknüpfungen zu Webauftritten Dritter (“externe Links”). Wir haben bei der erstmaligen Verknüpfung zu externen Links die fremden Inhalte daraufhin überprüft, ob etwaige Rechtsverstöße bestehen. Zu diesem Zeitpunkt haben wir keine Rechtsverstöße vorgefunden. Wir haben jedoch weder Einfluss auf die aktuelle und zukünftige Gestaltung der verknüpften Seiten noch auf deren Inhalte oder Angebote. Sollten uns Rechtsverstöße bekannt werden, löschen wir die betreffenden externen Links unverzüglich. Bitte weisen Sie uns gegebenenfalls darauf hin.