

# What's New in WCAG 2.2

Paul J. Adam  
Web & Mobile Accessibility Consultant  
[www.pauljadam.com](http://www.pauljadam.com)

[https://www.icloud.com/keynote/010q8hqzuHot0VFBYfqQVBxQw#What's New in WCAG 2](https://www.icloud.com/keynote/010q8hqzuHot0VFBYfqQVBxQw#What's%20New%20in%20WCAG%202.2)



# Agenda

- WCAG 2.2 Docs
- What's Changed
- What's New
- What's Removed
- What's Watered Down
- What's Next

The screenshot shows the W3C Web Accessibility Initiative (WAI) website. The header includes the W3C logo, navigation links for Accessibility Fundamentals, Planning & Policies, Design & Develop, Test & Evaluate, Teach & Advocate, and Standards/Guidelines, and links for Skip to Content, Change Text Size or Colors, and All Translations.

The main content area is titled "What's New in WCAG 2.2 Draft". It features a "Summary" section stating that the page lists proposed new success criteria for Web Content Accessibility Guidelines (WCAG) 2.2. It includes quotes from personas (fictional people) to help understand the success criteria. Below this is a "Page Contents" section with a bulleted list of topics, many of which are linked to specific sections of the WCAG 2.2 draft document.

**Standards/ Guidelines**

- Web Content - WCAG 2
  - How to Meet WCAG 2 (Quick Reference)
  - At a Glance
  - The Documents
  - Applying to Non-Web ICT
- New in 2.2 Draft
  - New in 2.1
  - Translations
  - Commenting
  - Conformance Logos
  - FAQ
- WCAG 3 Draft
- Authoring Tools - ATAG
- User Agents - UAAG
- WAI-ARIA
- Evaluation - ACT & EARL
- WAI-Adapt
- Pronunciation
- Standards Harmonization is Essential
- W3C Process for Developing Standards
- Referencing and Linking to Standards

**What's New in WCAG 2.2 Draft**

**Summary**

This page lists the **proposed** new success criteria for Web Content Accessibility Guidelines (WCAG) 2.2. It includes quotes from personas (fictional people) to help you understand some aspects of the success criteria. It also includes links to Understanding documents that explain the success criteria in detail and provide more examples.

**Page Contents**

- [Introduction, Timeline, Changes](#)
  - [Implementations and Comments](#)
  - [Changes from WCAG 2.1 to WCAG 2.2](#)
  - [Changes to the 2.2 Draft](#)
- [Guideline 2.4 Navigable](#)
  - [2.4.11 Focus Appearance \(AA\)](#)
  - [2.4.12 Focus Not Obscured \(Minimum\) \(AA\)](#)
  - [2.4.13 Focus Not Obscured \(Enhanced\) \(AAA\)](#)
- [Guideline 2.5 Input Modalities](#)
  - [2.5.7 Dragging Movements \(AA\)](#)
  - [2.5.8 Target Size \(Minimum\) \(AA\)](#)
- [Guideline 3.2 Predictable](#)
  - [3.2.6 Consistent Help \(A\)](#)
- [Guideline 3.3 Input Assistance](#)
  - [3.3.7 Accessible Authentication \(AA\)](#)
  - [3.3.8 Accessible Authentication \(No Exception\) \(AAA\)](#)
  - [3.3.9 Redundant Entry \(A\)](#)
- [About the Personas Quotes](#)

**Introduction, Timeline, Changes**

For an introduction to Web Content Accessibility Guidelines (WCAG) and more about versions 2.0, 2.1, and 2.2, see the [WCAG Overview](#).

WCAG 2.2 is scheduled to be completed and published by December 2022. Current versions:

- [WCAG 2.2 W3C Candidate Recommendation Snapshot](#) was approved for publication in September 2022
- [Editors' Draft of WCAG 2.2](#) may include more recent proposed changes that are not yet all approved

The main purpose of "Candidate Recommendation" is to ensure that the standard can be implemented. It is

# WCAG 2.2 Docs

- Latest published version:
  - <https://www.w3.org/TR/WCAG22/>
- Latest editor's draft:
  - <https://w3c.github.io/wcag/guidelines/22/>
- What's New in WCAG 2.2 Draft:
  - <https://www.w3.org/WAI/standards-guidelines/wcag/new-in-22/>

## Web Content Accessibility Guidelines (WCAG) 2.2

[W3C Editor's Draft 20 September 2022](#)



### ▼ More details about this document

#### This version:

<https://w3c.github.io/wcag/guidelines/22/>

#### Latest published version:

<https://www.w3.org/TR/WCAG22/>

#### Latest editor's draft:

<https://w3c.github.io/wcag/guidelines/22/>

#### History:

<https://www.w3.org/standards/history/WCAG22>

[Commit history](#)

#### Latest Recommendation:

<https://www.w3.org/TR/WCAG/>

#### Editors:

[Chuck Adams](#) (Oracle)

[Alastair Campbell](#) (Nomensa)

[Rachael Montgomery](#) (Invited Expert)

[Michael Cooper](#) (W3C)

[Andrew Kirkpatrick](#) (Adobe)

#### Feedback:

[GitHub w3c/wcag](#) ([pull requests](#), [new issue](#), [open issues](#))

Copyright © 2020-2022 W3C® ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)). W3C [liability](#), [trademark](#) and [document use](#) rules apply.

# What's Changed from 2.1 to 2.2

- Focus Visible changed from Level AA to Level A
  - 2.4.7 Focus Visible (Level A): Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible.
  - <https://w3c.github.io/wcag/understanding/focus-visible.html>
- 2.5.5 Target Size (Level AAA) renamed to 2.5.5 Target Size (Enhanced) (Level AAA)
  - <https://w3c.github.io/wcag/understanding/target-size-enhanced.html>

Contents GL: Navigable Previous SC: Headings and Labels Next SC: Location

## Understanding Success Criterion 2.4.7: Focus Visible

Success Criterion [2.4.7 Focus Visible](#) (Level A): Any keyboard operable user interface has a mode of operation where the keyboard focus indicator is visible.

**On this page:**

- Intent
- Benefits
- Examples
- Related Resources
- Techniques

### Intent

The purpose of this success criterion is to help a person know which element has the keyboard focus.

“Mode of operation” accounts for user agents which may not always show a focus indicator, or only show the focus indicator when the keyboard is used. User agents may optimise when the focus indicator is shown, such as only showing it when a keyboard is used. Authors are responsible for providing at least one mode of operation where the focus is visible. In most cases there is only one mode of operation so this success criterion applies. The focus indicator must not be time limited, when the keyboard focus is shown it must remain.

Note that a keyboard focus indicator can take different forms. New in WCAG 22: This criterion does not specify what the form is, but [Focus Appearance \(Minimum\)](#) does define how visible the indicator should be. Passing [Focus Appearance \(Minimum\)](#) would pass this success criterion.

### Benefits

- This Success Criterion helps anyone who relies on the keyboard to operate the page, by letting them visually determine the component on which keyboard operations will interact at any point in time.
- People with attention limitations, short term memory limitations, or limitations in executive processes benefit by being able to discover where the focus is located.

# What's New

- 9 New Success Criteria
- Guideline 2.4 Navigable
  - 2.4.11 Focus Appearance (AA)
  - 2.4.12 Focus Not Obscured (Minimum) (AA)
  - 2.4.13 Focus Not Obscured (Enhanced) (AAA)
- Guideline 2.5 Input Modalities
  - 2.5.7 Dragging Movements (AA)
  - 2.5.8 Target Size (Minimum) (AA)
- Guideline 3.2 Predictable
  - 3.2.6 Consistent Help (A)
- Guideline 3.3 Input Assistance
  - 3.3.7 Accessible Authentication (AA)
  - 3.3.8 Accessible Authentication (No Exception) (AAA)
  - 3.3.9 Redundant Entry (A)

## What's New in WCAG 2.2 Draft

### Summary

This page lists the **proposed** new success criteria for Web Content Accessibility Guidelines (WCAG) 2.2.

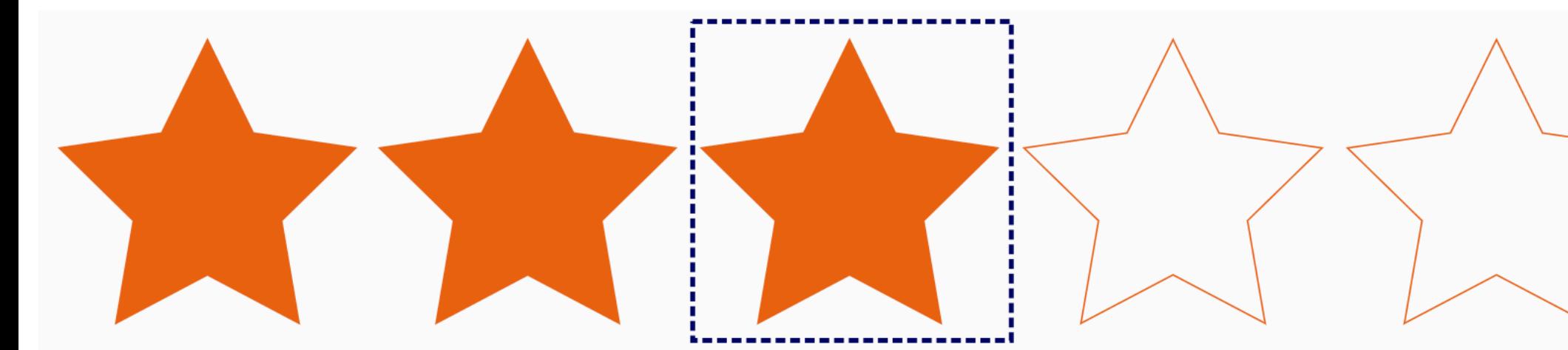
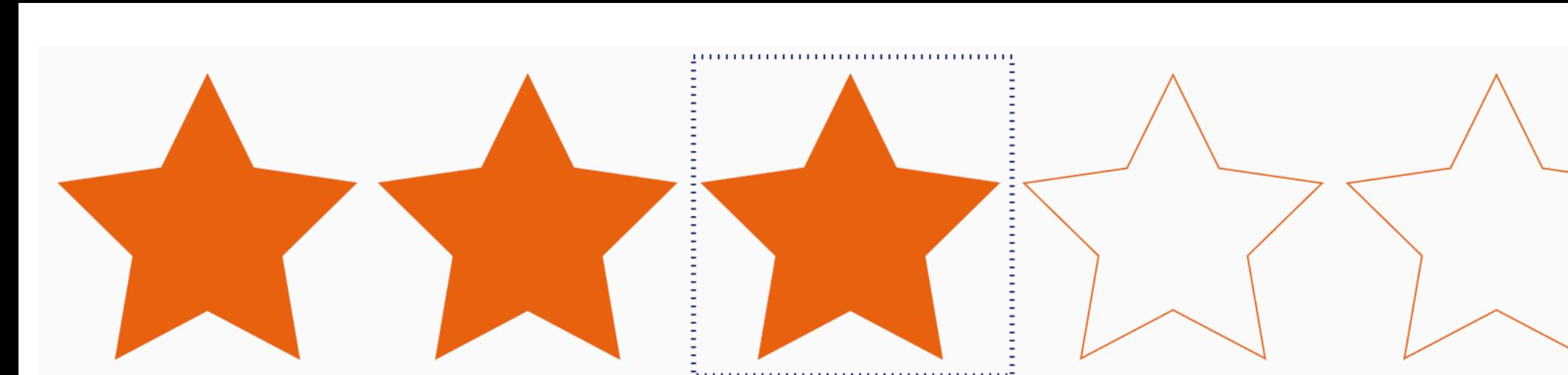
**It includes quotes from personas (fictional people)** to help you understand some aspects of the success criteria. It also includes links to Understanding documents that explain the success criteria in detail and provide more examples.

### Page Contents

- [Introduction, Timeline, Changes](#)
  - [Implementations and Comments](#)
  - [Changes from WCAG 2.1 to WCAG 2.2](#)
  - [Changes to the 2.2 Draft](#)
- [Guideline 2.4 Navigable](#)
  - [2.4.11 Focus Appearance \(AA\)](#)
  - [2.4.12 Focus Not Obscured \(Minimum\) \(AA\)](#)
  - [2.4.13 Focus Not Obscured \(Enhanced\) \(AAA\)](#)
- [Guideline 2.5 Input Modalities](#)
  - [2.5.7 Dragging Movements \(AA\)](#)
  - [2.5.8 Target Size \(Minimum\) \(AA\)](#)
- [Guideline 3.2 Predictable](#)
  - [3.2.6 Consistent Help \(A\)](#)
- [Guideline 3.3 Input Assistance](#)
  - [3.3.7 Accessible Authentication \(AA\)](#)
  - [3.3.8 Accessible Authentication \(No Exception\) \(AAA\)](#)
  - [3.3.9 Redundant Entry \(A\)](#)
- [About the Personas Quotes](#)

# 2.4.11 Focus Appearance (AA)

- <https://w3c.github.io/wcag/understanding/focus-appearance.html>



## 2.4.11 Focus Appearance (AA)

Reporter with repetitive stress injury who doesn't use a mouse:  
and Retiree with low contrast sensitivity:

**Problem:** "I can't tell where the keyboard focus is as I move around a web page or app."

**Works well:** "I can see where the keyboard focus is as I move around a web page or app."

WCAG:

When the keyboard focus indicator is visible, one or both of the following are true:

1. The entire focus indicator meets all the following:
  - encloses the user interface component or sub-component that is focused, and
  - has a contrast ratio of at least 3:1 between the same pixels in the focused and unfocused states, and
  - has a contrast ratio of at least 3:1 against adjacent non-focus-indicator colors.
2. An area of the focus indicator meets all the following:
  - is at least as large as the area of a 1 CSS pixel thick perimeter of the unfocused component or sub-component, or is at least as large as a 4 CSS pixel thick line along the shortest side of the minimum bounding box of the unfocused component or sub-component, and
  - has a contrast ratio of at least 3:1 between the same pixels in the focused and unfocused states, and
  - has a contrast ratio of at least 3:1 against adjacent non-focus-indicator colors, or is no thinner than 2 CSS pixels.

Exceptions:

- The focus indicator is determined by the user agent and cannot be adjusted by the author, or
- The focus indicator and the indicator's background color are not modified by the author.

Note: What is perceived as the user interface component or sub-component (to determine enclosure or size) depends on its visual presentation. The visual presentation includes the component's visible content, border, and component-specific background. It does not include shadow and glow effects outside the component's content, background, or border.

Note: Examples of sub-components that may receive a focus indicator are menu items in an opened drop-down menu, or focusable cells in a grid.

Note: Contrast calculations can be based on colors defined within the technology (such as HTML, CSS and SVG). Pixels modified by user agent resolution enhancements and anti-aliasing can be ignored.

Editor's note: This Success Criterion is at risk.

# 2.4.12 Focus Not Obscured (Minimum) (AA)

## 2.4.12 Focus Not Obscured (Minimum) (AA)

Reporter with repetitive stress injury who uses speech recognition software:

**Problem:** “This page has a big banner that's always across the bottom. (*a sticky footer*) When I move focus to items, some are hidden behind the banner and I can't see them.”

**Works well:** “When I move focus to items, I can see them all.”

WCAG:

When a user interface component receives keyboard focus, the component is not entirely hidden due to author-created content.

- [https://www.w3.org/WAI/WCAG22/  
Understanding/focus-not-obsured-  
minimum](https://www.w3.org/WAI/WCAG22/Understanding/focus-not-obsured-minimum)

### Sufficient Techniques

1. CSS: Using scroll-padding to ensure a sticky header does not obscure the focused item (Potential future technique).

# 2.4.13 Focus Not Obscured (Enhanced) (AAA)

- <https://w3c.github.io/wcag/understanding/focus-not-obsured-enhanced.html>
- This criterion is closely related to Focus Not Obscured (Minimum) but requires that the whole of the component is visible.

## 2.4.13 Focus Not Obscured (Enhanced) (AAA)

Reporter with repetitive stress injury who uses speech recognition software:

**Problem:** “This page has a big banner that's always across the bottom. When I move focus to items, some are hidden behind the banner and I can't see them.”

**Works well:** “When I move focus to items, I can see them all.”

WCAG:

When a user interface component receives keyboard focus, no part of the focus indicator is hidden by author-created content.

## Benefits

- This Success Criterion helps anyone who relies on the keyboard to operate the page by letting them visually determine the component on which keyboard operations will interact at any point in time.
- People with attention limitations, short term memory limitations, or limitations in executive processes benefit by being able to discover where the focus is located.

# 2.5.7 Dragging Movements (AA)

- <https://www.w3.org/WAI/WCAG22/Understanding/dragging-movements>
- G219: Ensuring that a single pointer alternative is available for dragging movements that operate on content
- <https://www.w3.org/WAI/WCAG22/Techniques/general/G219>

## 2.5.7 Dragging Movements (AA)

Retiree with hand tremor:

**Problem:** "I cannot hold down the mouse button and drag it accurately enough to move the items in this list."

**Works well:** "When I click on an item in the list, I get up and down arrows and I can click those to change the order."

WCAG:

All functionality that uses a [dragging movement](#) for operation can be achieved by a [single pointer](#) without dragging, unless dragging is [essential](#) or the functionality is determined by the user agent and not modified by the author.

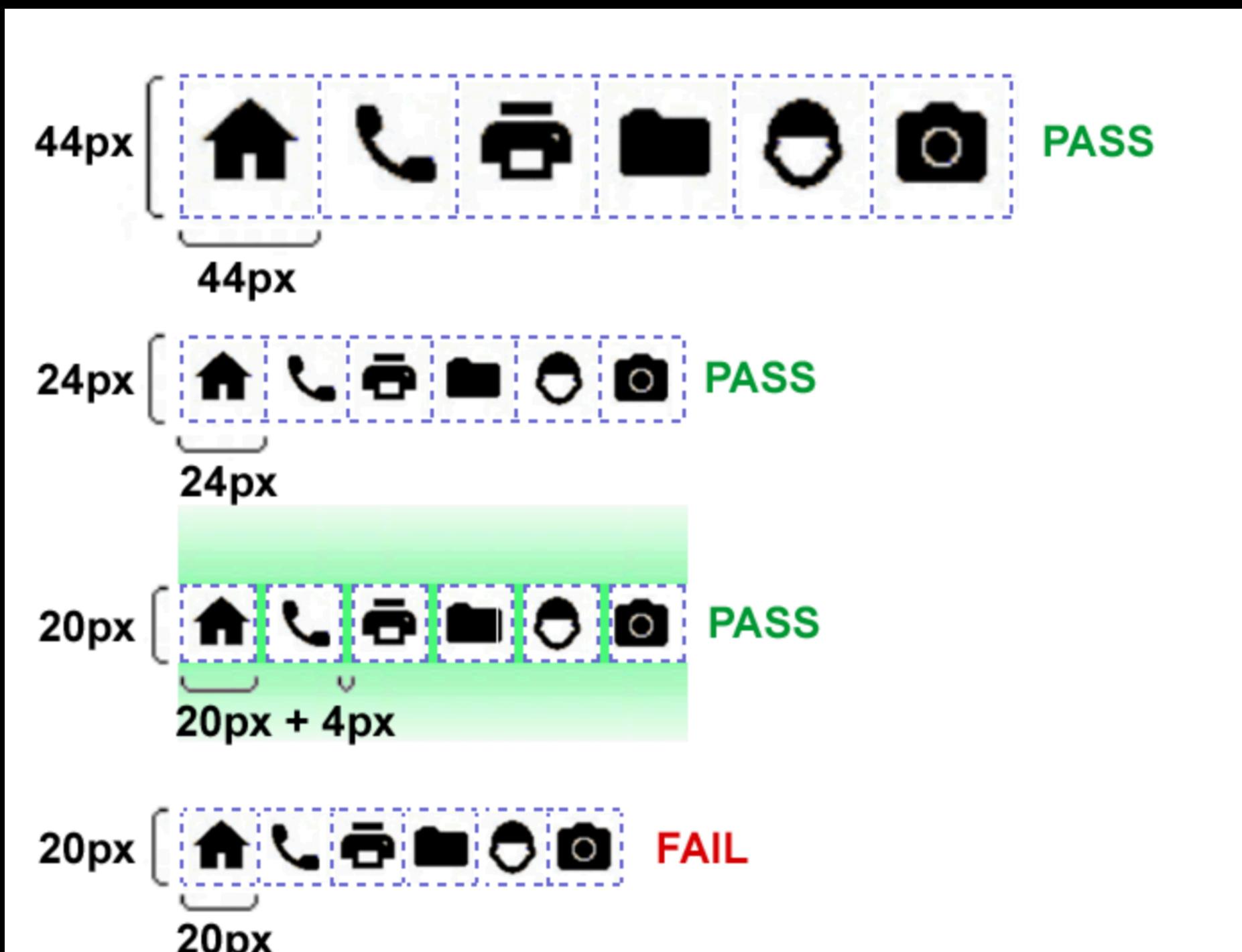
Note: This requirement applies to web content that interprets pointer actions (i.e. this does not apply to actions that are required to operate the user agent or assistive technology).

### Example 1

- A list of items can be re-ordered by picking up an item and dragging it upwards or downwards. Other elements move dynamically to open a gap where the picked-up target can be dropped. After a single pointer activation, the list items display up and down arrows which allow a step-wise re-ordering of the list via [single pointer](#) inputs (taps or clicks at the up or down arrow).
- A vertical priority list indicates the priority of items listed. Each item can be 'picked up' with a pointer and dragged up or down to another position. The other list items rearrange dynamically. To the left of each list item, a number in a text field shows the current priority position. For any of the items, users can put in another number. This leads to a dynamic reordering and renumbering of the priority list.
- In a Kanban implementation for process management, tasks can be dragged horizontally from one 'swimming lane' to another in order to change the status of tasks (for example, to change the status of a task from "in process" to "completed"). One or several items in a lane can be selected by a single tap or click. A single pointer activation of a drop-down menu labelled "Move selected items to" offers a selection of drop targets (other lanes). A further single pointer activation over the desired menu item moves targets to the specified lane.
- In a radial control widget, the visual indicator of the current value of the control can be dragged to a different position. Users can also click or tap on another position of the radial control to change the value.

# 2.5.8 Target Size (Minimum) (AA)

- <https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum.html>



## 2.5.8 Target Size (Minimum) (AA)

Retiree with hand tremor:

**Problem:** “The buttons are so close together, I hit “Cancel” when going for “Submit”. Then I have to start all over again.”

**Works well:** “There is more space between the buttons so I don’t hit the wrong button even when I’m riding on the bumpy bus.”

WCAG:

The size of the target for pointer inputs is at least 24 by 24 CSS pixels, except where:

- **Spacing:** The target offset is at least 24 CSS pixels to every adjacent target;
- **Equivalent:** The function can be achieved through a different control on the same page that has an area of at least 24 by 24 CSS pixels;
- **Inline:** The target is in a sentence or block of text;
- **User agent control:** The size of the target is determined by the user agent and is not modified by the author;
- **Essential:** A particular presentation of the target is essential or is legally required for the information being conveyed.

Note: Targets that allow for values to be selected spatially based on position within the target are considered one target for the purpose of the success criterion. Examples include sliders with granular values, color pickers displaying a gradient of colors, or editable areas where you position the cursor.

### Sufficient Techniques

1. [C42: Using min-height and min-width to ensure sufficient target spacing](#)

Figure 1 Four rows of targets, illustrating three ways of meeting this Success Criterion and one way of failing it.

# 3.2.6 Consistent Help (A)

- [https://www.w3.org/WAI/WCAG22/  
Understanding/consistent-help](https://www.w3.org/WAI/WCAG22/Understanding/consistent-help)

## Benefits

- This Success Criterion helps people who may have difficulty locating help that is not part of the page they are using. Ensuring they can connect with help allows them to complete the task.
- When a user can easily find help for their specific questions, they are more successful completing tasks.
- Users that experience cognitive fatigue or cognitive shut down will be able to reserve their energy for the task, instead of using it to find support.
- Enabling users (especially those with cognitive disabilities) to find solutions while expressing their question using their own words increases their chances of success for completing a task.

## Sufficient Techniques

1. [G220: Provide a contact-us link in a consistent location](#)

## Failures

The following are common mistakes that are considered failures of this Success Criterion by the WCAG Working Group.

- Inconsistent Help Location

## 3.2.6 Consistent Help (A)

Supermarket assistant with cognitive disabilities:

**Problem:** "Whenever I use the online app to schedule my medical appointments, I can't remember what to do at each step. I've seen a Chat option in some places, but can't find it now."

**Works well:** "When I need help, I can easily find the Chat option that's always in the lower right corner of the page."

WCAG:

If a web page contains any of the following help mechanisms, and those mechanisms are repeated on multiple web pages within a set of web pages, they occur in the same relative order to other page content, unless a change is initiated by the user:

- Human contact details;
- Human contact mechanism;
- Self-help option;
- A fully automated contact mechanism.

Note: Access to help mechanisms may be provided directly on the page, or may be provided via a direct link to a different page containing the information.

Note: For this Success Criterion, the same relative order can be thought of as how the content is ordered when the page is serialized. The visual position of a help mechanism is likely to be consistent across pages for the same page variation (e.g., CSS break-point). The user can initiate a change, such as changing the page's zoom or orientation, which may trigger a different page variation. This criterion is concerned with relative order across pages displayed in the same page variation (e.g., same zoom level and orientation).

# 3.3.7 Accessible Authentication (AA)

- [https://www.w3.org/WAI/WCAG22/  
Understanding/accessible-authentication](https://www.w3.org/WAI/WCAG22/Understanding/accessible-authentication)

## Benefits

People with cognitive issues relating to memory, reading (for example, dyslexia), numbers (for example, dyscalculia), or perception-processing limitations will be able to authenticate irrespective of the level of their cognitive abilities.

## Examples

- A web site uses a properly marked up username (or email) and password fields as the login authentication (meeting [Success Criterion 1.3.5 Input Purpose](#) and [Success Criterion 4.1.2: Name, Role, Value](#)). The user's browser or integrated third-party password manager extension can identify the purpose of the inputs and automatically fill in the username and password.
- A web site does not block paste functionality. The user is able to use a third-party password manager to store credentials, copy them, and paste them directly into a login form.
- A web site uses WebAuthn so the user can authenticate with their device instead of username/password. The user's device could use any available modality. Common methods on laptops and phones are facial-scan, fingerprint, and PIN (Personal Identification Number). The web site is not enforcing any particular use, it is assumed a user will setup a method that suits them.
- A web site offers the ability to login with a third-party provider using the OAuth method.
- A web site that requires two-factor authentication allows for multiple options for the 2nd factor, including a USB-based method where the user simply presses a button to enter a time-based token.
- A web site that requires two-factor authentication displays a QR code which can be scanned by an app on a user's device to confirm identity.
- A web site that requires two-factor authentication sends a notification to a user's device. The user must use their device's authentication mechanism (for example, user-defined PIN, fingerprint, facial recognition) to confirm identity.

## 3.3.7 Accessible Authentication (AA)

[Supermarket assistant with cognitive disabilities:](#)

**Problem:** "I can never remember my password, it's really hard to get into this app."

**Works well:** "To get into this app, I can put my e-mail address. Then I get an e-mail message, and I can click a link in the e-mail to get into the app."

WCAG:

A [cognitive function test](#) (such as remembering a password or solving a puzzle) is not required for any step in an authentication process unless that step provides at least one of the following:

- **Alternative:** Another authentication method that does not rely on a cognitive function test.
- **Mechanism:** A mechanism is available to assist the user in completing the cognitive function test.
- **Object Recognition:** The cognitive function test is to recognize objects.
- **Personal Content:** The cognitive function test is to identify non-text content the user provided to the website.

Note: Objects to recognize and user provided content may be represented by images, video, or audio.

Note: Examples of mechanisms that satisfy this criterion include:

1. support for password entry by password managers to reduce memory need, and
2. copy and paste to reduce the cognitive burden of re-typing.

[Understanding Accessible Authentication](#)

# 3.3.8 Accessible Authentication (No Exception) (AAA)

- [https://www.w3.org/WAI/WCAG22/  
Understanding/accessible-authentication-  
no-exception](https://www.w3.org/WAI/WCAG22/Understanding/accessible-authentication-no-exception)
- This criterion is the same as Accessible Authentication but without the exceptions for objects and user-provided content.

## Sufficient Techniques

1. [G218: Email link authentication](#)
2. Providing a properly marked up email and password inputs (Potential future technique)
3. Providing WebAuthn as an alternative to username/password (Potential future technique)
4. Providing a 3rd party login using oAuth (Potential future technique)
5. Using two techniques to provide 2 factor authentication (Potential future technique)

## Key Terms

### cognitive function test

A task that requires the user to remember, manipulate, or transcribe information. Examples include, but are not limited to:

- memorization, such as remembering a username, password, set of characters, images, or patterns. The common identifiers name, e-mail, and phone number are not considered cognitive function tests as they are personal to the user and consistent across websites;
- transcription, such as typing in characters;
- use of correct spelling;
- performance of calculations;
- solving of puzzles.

## 3.3.8 Accessible Authentication (No Exception) (AAA)

Supermarket assistant with cognitive disabilities:

**Problem:** “I can never remember my password, it’s really hard to get into this app.”

**Works well:** “To get into this app, I can put my e-mail address. Then I get an e-mail message, and I can click a link in the e-mail to get into the app.”

WCAG:

A cognitive function test (such as remembering a password or solving a puzzle) is not required for any step in an authentication process unless that step provides at least one of the following:

- **Alternative:** Another authentication method that does not rely on a cognitive function test.
- **Mechanism:** A mechanism is available to assist the user in completing the cognitive function test.

Understanding Accessible Authentication (No Exception)

# 3.3.9 Redundant Entry (A)

- [https://www.w3.org/WAI/WCAG22/  
Understanding/redundant-entry](https://www.w3.org/WAI/WCAG22/Understanding/redundant-entry)

## Examples

- A form requests the user's corporate identification number (ID) in the first step of a process to purchase a new computer. In the 3rd step the user is asked to confirm that the computer will belong to the user (rather than a colleague), and re-shows the ID. It allows the user to change the ID, but defaults to the previously entered one.
- A form on an e-commerce website allows the user to confirm that the billing address and delivery address are the same address.
- A search results page pre-fills the search input with the previously entered search term in the same process.

## 3.3.9 Redundant Entry (A)

Supermarket assistant with cognitive disabilities:

**Problem:** “Whenever I use the online app to schedule my medical appointments, I have to re-type some information that I entered in a previous step.”

**Works well:** “The app automatically fills in information that I entered in previous steps.”

WCAG:

Information previously entered by or provided to the user that is required to be entered again in the same process is either:

- auto-populated, or
- available for the user to select.

Except when:

- re-entering the information is essential,
- the information is required to ensure the security of the content, or
- previously entered information is no longer valid.

## Sufficient Techniques

- [G221: Provide data from a previous step in a process](#)
- Not requesting the same information twice in multiple steps (Potential future technique)

# What's Removed

- 3.2.7 Visible Controls (Level AA)
  - Renamed from Hidden Controls
  - <https://w3c.github.io/wcag/understanding/visible-controls.html>
- 2.4.13 Focus Appearance (Enhanced) (Level AAA)
  - Renamed from Focus Visible (Enhanced)
  - <https://www.w3.org/WAI/WCAG22/Understanding/focus-appearance-enhanced>
- 2.4.14 Page Break Navigation (Level A)
  - Renamed from Fixed Reference Points
  - <https://w3c.github.io/wcag/understanding/page-break-navigation.html>
- <https://w3c.github.io/wcag/guidelines/22/#changelog>
- Plans add Visible Controls and Page Break Navigation to Supplemental Guidance
  - <https://www.w3.org/WAI/WCAG2/supplemental/>

Contents

GL: Predictable

Previous SC: Consistent Help

Next SC: Error Identification

## Understanding Success Criterion 3.2.7: Visible Controls

On this page:
Intent
Benefits
Examples
Techniques
Key Terms

Success Criterion [3.2.7 Visible Controls](#) (Level AA): When [user interface components](#) are invisible until hover or focus makes them visible, provide a visible indicator that the components are available, except when:

- The same functionality is available through a component on the same page, or on a different step in a multi-step process, without requiring pointer hover or keyboard focus to make it visible;
- The component provides keyboard-only functionality;
- A mechanism is available to make the components persistently visible;
- Hiding the visual indicator is [essential](#).

User interface components can be available through other visible components such as sub-menus, edit buttons, tabs, or thumbnails of media.

The working group is interested in feedback about whether there are Components determined by the user agent that should not be in scope.

## Status

This understanding document is part of the [draft WCAG 2.2 content](#). It may change or be removed before the final WCAG 2.2 is published.

# What's Watered Down

- 2.5.8 Target Size (Minimum) (Level AA)
  - Renamed from Pointer Target Spacing
  - Originally required 44 x 44 CSS pixel target size for controls
  - Now only requires 24 x 24 CSS pixel target size
  - <https://www.w3.org/WAI/WCAG22/Understanding/target-size-minimum.html>
- 44 x 44 target size minimum still only Level AAA
  - 2.5.5 Target Size (Enhanced) (Level AAA)
  - Renamed from 2.5.5 Target Size (Level AAA)
  - AAA not legally required

[Contents](#) [GL: Input Modalities](#) [Previous SC: Dragging Movements](#) [Next SC: Language of Page](#)

## Understanding Success Criterion 2.5.8: Target Size (Minimum)

Success Criterion [2.5.8 Target Size \(Minimum\)](#) (Level AA): The size of the [target](#) for [pointer inputs](#) is at least 24 by 24 CSS pixels, except where:

- **Spacing:** The [target offset](#) is at least 24 CSS pixels to every adjacent target;
- **Equivalent:** The function can be achieved through a different control on the same page that meets this criterion.
- **Inline:** The target is in a sentence or block of text;
- **User agent control:** The size of the target is determined by the user agent and is not modified by the author;
- **Essential:** A particular presentation of the target is [essential](#) or is legally required for the information being conveyed;

Targets that allow for values to be selected spatially based on position within the target are considered one target for the purpose of the success criterion. Examples include sliders with granular values, color pickers displaying a gradient of colors, or editable areas where you position the cursor.

### Status

This understanding document is part of the [draft WCAG 2.2 content](#). It may change or be removed before the final WCAG 2.2 is published.

On this page:
<a href="#">Intent</a>
<a href="#">Benefits</a>
<a href="#">Examples</a>
<a href="#">Related Resources</a>
<a href="#">Techniques</a>
<a href="#">Key Terms</a>

# What's Next

- WCAG 3.0 (Silver)
  - Technology neutral
    - web, ePub, PDF, applications, mobile apps, and other emerging technologies
  - More Usable / Plain Language documentation
  - No more A, AA, AAA levels
    - Now it's Bronze, Gold, Silver
  - No more e.g. 1.1.1 or 1.2.2 Numbered Success Criteria
    - Now it's "Text Alternatives" and "Captions"
  - No more all or nothing conformance
  - Likely take years or a decade to become legally required
    - I predict pushback due to major changes from WCAC 2.X to 3.0
- Latest published version:
  - <https://www.w3.org/TR/wcag-3.0/>
- Latest editor's draft:
  - <https://w3c.github.io/silver/guidelines/>
- WCAG 3 How-Tos
  - Text alternatives | How-To | WCAG 3 | Web Accessibility Initiative (WAI) | W3C
    - <https://www.w3.org/WAI/GL/WCAG3/2022/how-tos/text-alternatives/>

## W3C Accessibility Guidelines (WCAG)

3.0

[W3C Editor's Draft 07 July 2022](#)



▼ [More details about this document](#)

This version:

<https://w3c.github.io/silver/guidelines/>

Latest published version:

<https://www.w3.org/TR/wcag-3.0/>

Latest editor's draft:

<https://w3c.github.io/silver/guidelines/>

History:

<https://www.w3.org/standards/history/wcag-3.0>

[Commit history](#)

Editors:

[Jeanne Spellman](#) (TetraLogical)

[Rachael Bradley Montgomery](#) (Library of Congress)

[Shawn Lauriat](#) (Google, Inc.)

[Michael Cooper](#) (W3C)

Project Manager:

[Wilco Fiers](#) (Deque Systems, Inc)

Feedback:

[GitHub w3c/silver](#) ([pull requests](#), [new issue](#), [open issues](#))

Copyright © 2021-2022 W3C® ([MIT](#), [ERCIM](#), [Keio](#), [Beihang](#)). W3C liability, trademark and document license terms apply.

w3c

a11y

ag  
silver