Web Accessibility Checklist

Based on WCAG 2.2 AA

Background

Deque recommends using a combination of software tools and informed human analysis — using a checklist such as the one in this document — to test web accessibility.

Web Content Accessibility Guidelines

This checklist is based on the Web Content Accessibility Guidelines (WCAG) 2.2, which is a W3C (World Wide Web Consortium) Recommendation as of October 5, 2023.

Accessibility Training

- Find detailed online accessibility courses and reference materials on Deque University.
- Experienced Deque web accessibility instructors are also available.

Deque Accessibility Evaluation Software

axe DevTools is an automated and guided accessibility testing solution for Component Developers, Front-End Developers, Native Mobile App Developers and Test Engineers that allows you to easily find and fix 76-84% of accessibility errors before your applications get out of development. Solutions for Unit Testing, Test Automation, and integration with popular testing frameworks such as Selenium and QUnit, combined with in-browser testing using Deque's axe DevTools extensions for Chrome and Firefox, allow you incorporate accessibility testing into your existing local development and testing environment.

axe Auditor is a manual accessibility testing solution for Accessibility Testing Specialists that allows you to optimize your accessibility testing, tracking, and reporting with step-by-step guidance and report building. Based on The Deque Way of accessibility testing, your QA team can run both manual and automated tests in the same interface, and without needing to be accessibility experts themselves, they can create precise, consistent accessibility issue reports for developers to resolve, then rerun tests on new builds and across multiple browsers to verify requirements have been met.

axe Monitor is an enterprise-level scanning and reporting tool that can evaluate the web accessibility of entire web sites, including password-protected areas. axe Monitor can implement use case scripts, allow custom reporting options based on WCAG or Section 508 guidelines, and allow scheduled monitoring of accessibility on projects of any scale or complexity. It is designed to work in conjunction with the axe DevTools Browser Extension for Chrome and Firefox.

axe, the Accessibility Engine, is Deque's free, open-source JavaScript accessibility rules library and browser extension. It was developed to be fast and to return zero false errors or duplicate results, and it is available as a GitHub repository, browser plugin, or framework integration.

Notes for This Checklist

Where "(WCAG 2.1)" or "(WCAG 2.2)" is written after a WCAG criterion number, this indicates the WCAG version in which the requirement was introduced.

Section 1: Structure and Semantics

General Notes About Semantic Markup

If the technology used provides semantic structure to convey the relationships between information:

- Use semantic markup to mark emphasized or special text.
- Use text to convey information that is conveyed by variations of presentation of text.
- Separate information and structure from presentation. Do not use style to convey structure.

Convey relationships between information by correctly using web page structure, by using:

- Semantic markup when color cues are used
- Real HTML list elements - `ol`, `ul` and `dl` for lists (or `div role='list'` and `role='listitem'`).
- Real HTML heading elements `h1` – `h6` to identify headings. Headings must be informative and clearly describe the topic that follows it. (Text size is irrelevant for headings — size and style can be implemented via CSS).
- Functions of the Document Object Model to add content to a page.

Page Title

Topic	Technique	WCAG AA
Markup	**Correct Markup:** The page `<title>` MUST be present and MUST contain text.	Required WCAG 2.4.2
Meaningful Text	**Accurate and Informative:** The page `<title>` MUST be accurate and informative.	Required WCAG 2.4.2
	Dynamic Pages: The page `<title>` of dynamic pages (e.g. in single page apps) MUST be updated when the purpose of the page changes.	Required WCAG 2.4.2
	User Actions: If a page is the result of a user action or scripted change of context, the text of the `<title>` SHOULD describe the result or change of context to the user.	Best practice
	Concise: The `<title>` SHOULD be concise.	Best practice
	Unique: The page `<title>` SHOULD be unique, if possible.	Best practice
	Unique Info First: Unique information SHOULD come first in the `<title>`.	Best practice
	Match Heading: The page `<title>` SHOULD match (or be very similar to) the top heading (ideally marked as `<h1>`) in the main content.	Best practice

Language

Topic	Technique	WCAG AA
Page Language	**Page Language:** The primary language of the page MUST be identified accurately, using a standard language code, on the `<html>` element (e.g. `<html lang="en">` or `<html lang="fr">`).	Required WCAG 3.1.1
Language of Parts	**Language of Parts:** Inline language changes MUST be identified with a valid `lang` attribute.	Required WCAG 3.1.2
Language Code	**Two-Character Language Code:** The language code SHOULD be designated with a standard two-character ISO 639-1 code (e.g. `lang="en"`) to achieve maximum support across screen readers and browsers, though other codes (e.g. `lang="en-us"`) are technically allowable.	Best practice

Headings

Topic	Technique	WCAG AA
Headings to Bypass Blocks of Content	**Bypass Blocks:** Screen readers allow users to navigate by headings, so headings are an effective way to bypass blocks of content, as required by WCAG 2.4.1. **Note:** Headings are not absolutely required by WCAG to pass 2.4.1, but are highly recommended, along with landmarks and skip links.	Required WCAG 2.4.1
Meaningful Text	**Accurate, Informative Section Labels:** Headings MUST be accurate and informative, as labels for the sections of text they describe.	Required WCAG 1.3.1 WCAG 2.4.6
	Brevity: Heading text SHOULD be concise and relatively brief.	Best practice
Heading Markup	**Use Real Headings:** Text that acts as a heading visually or structurally MUST be designated as a true heading (`<h1>`, `<h2>`, etc.) in the markup.	Required WCAG 1.3.1
	Heading Markup for Headings Only: Text that does not act as a heading visually or structurally MUST NOT be marked as a heading.	Required WCAG 1.3.1
Outline/Hierarchy of Content	**Content Outline:** Headings SHOULD convey a clear and accurate structural outline of the sections of content of a web page.	Best practice
	Consecutive Levels: Headings SHOULD NOT skip hierarchical levels.	Best practice
	First Heading in the Main Content: The beginning of the main content SHOULD start with `<h1>`.	Best practice
	One <h1>: Most web pages SHOULD have only one `<h1>`.	Best practice

Landmarks

Topic	Technique	WCAG AA
Landmarks to Bypass Blocks of Content	**Bypass Blocks:** Screen readers allow users to navigate by landmarks, so landmarks are an effective way to bypass blocks of content, as required by WCAG 2.4.1. Other methods may be used as well as — or instead of — landmarks, such as skip links, headings, expand/collapse regions, etc.	Required WCAG 2.4.1
Landmark Structural Organization	**Page Layout Groupings:** Landmarks SHOULD be used to accurately designate predefined parts of the layout (e.g. the header, navigation, main content, and footer).	Best practice
	Content Within Landmarks: All text SHOULD be contained within a landmark region.	Best practice
	Landmark Names: Multiple instances of the same type of landmark SHOULD be distinguishable by different discernible names (using `aria-label` or `aria-labelledby`).	Best practice
	Only One Instance of Some Landmarks: A page SHOULD NOT contain more than one instance of each of the following landmarks: `header`/`banner`, `main`, and `footer`/`contentinfo`.	Best practice
	Limit the Number of Landmarks: The total number of landmarks SHOULD be minimized to the extent appropriate for the content. **Note:** Having a large number of landmarks defeats the main purpose of landmarks, which is to make it easy to navigate quickly to sections of the layout.	Best practice
Markup	**Markup:** Landmarks MAY be designated with either HTML tags or their equivalent ARIA roles (e.g. `<header>` or `role="banner"`, `<nav>` or `role="navigation"`, `<main>` or `role="main"`, `<footer>` or `role="contentinfo"`, etc.).	Best practice

Lists

Topic	Technique	WCAG AA
Markup	**List Markup:** Lists MUST be constructed using the appropriate semantic markup (i.e. `<ul>` or `<ol>` with `<li>` child elements, or `<dl>` with `<dt>` and `<dd>` child elements).	Required WCAG 1.3.1

Tables

Topic	Technique	WCAG AA
Table Headers	**Header Tag:** Table headers MUST be designated with `<th>`.	Required WCAG 1.3.1
	Meaningful Header: Data table header text MUST accurately describe the category of the corresponding data cells.	Required WCAG 1.3.1
	Header and Data Cell Associations: Table data cells MUST be associated with their corresponding header cells. **Note:** Use of scope (`<th scope="col">` and `<th scope="row">`) is highly recommended, though not always necessary (i.e. if all cells in the first row are marked as `<th>` without scope, most modern screen readers will infer that the scope is the column below each header cell).	Required WCAG 1.3.1
	Group Header Associations: Table data group headers (if any) MUST be associated with their corresponding data cell groups (e.g. via `scope="rowgroup"` or `scope="colgroup"`).	Required WCAG 1.3.1
	Complex Header Associations: Header/data associations that cannot be designated with `<th>` and scope MUST be designated with the `header` and `id` attributes.	Required WCAG 1.3.1
	Nested or Split Tables: Data table headers and data associations MUST NOT be referenced across nested, merged, or separate tables.	Required WCAG 1.3.1
Tabular Data	**Tables:** Tabular data SHOULD be represented in a `<table>`. **Note:** Even if the data are not represented in a table, WCAG 1.3.1 requires the data to be associated with their labels.	Best practice
Caption	**Caption:** Data tables SHOULD have a programmatically associated `<caption>` or name (e.g. via `aria-label` or `aria-labelledby`). **Note:** In most circumstances, `<caption>` is preferred, because it is the native method of giving a name to a table, and the `<caption>` is visible and available to all users by default.	Best practice
	Meaningful Caption: The name or `<caption>` of a data table SHOULD describe the identity or purpose of the table accurately, meaningfully, and succinctly.	Best practice
	Unique Caption: The name or `<caption>` of each data table SHOULD be unique within the context of other tables on the same page.	Best practice
Layout Tables	**Avoid Layout Tables:** Tables SHOULD NOT be used for the purpose of purely visual (non-data) layout.	Best practice
	Avoid Headers in Layout Tables: Layout tables MUST NOT contain headers.	Best practice

Iframes

Topic	Technique	WCAG AA
Iframe `title` Attribute	**Meaningful Iframe `title` Attribute:** The iframe `title` attribute (on the parent page) MUST be accurate and descriptive.	Required WCAG 2.4.1
	Unique `title` Attributes: Every iframe SHOULD have a unique title (in the context of the page).	Best practice
	Hidden or Empty Iframes: Hidden frames or frames that do not convey content to users SHOULD be hidden from assistive technologies using `aria-hidden="true"`.	Best practice
Page Title of Embedded Page	**Page Title of Embedded Page:** The source page of an iframe (the page embedded in the iframe) MUST have a valid, meaningful `<title>`.	Required WCAG 2.4.2

Markup Validity

Note: In WCAG 2.2, success criterion 4.1.1 Parsing is designated as "obsolete and removed." Versions 2.1 and 2.0 of WCAG now include this note concerning conformance with 4.1.1 Parsing:

"This Success Criterion should be considered as always satisfied for any content using HTML or XML."

See the WCAG 2 FAQ for the reasoning behind the obsoletion of this criterion.

Other Semantic Elements

Topic	Technique	WCAG AA
Headings	See the requirements for ***Headings***.	Required (Multiple)
Landmarks	See the requirements for ***Landmarks***.	Required (Multiple)
Lists	See the requirements for ***Lists***.	Required (Multiple)
Tables	See the requirements for ***Tables***.	Required (Multiple)
Links	See the requirements for ***Links***.	Required (Multiple)
Emphasis and Highlighting	**Emphasis:** Critical emphasis in the text SHOULD be conveyed in a text-based format, not visual styling alone.	Best practice
	Highlighting Markup: Highlighted text SHOULD be marked with the `<mark>` element.	Best practice
	Text-Based Highlighting: Critical highlighted text SHOULD be supplemented with a text-based method to convey the meaning of the highlighting.	Best practice
Quotations	**Blockquote:** The `<blockquote>` element SHOULD be used to designate long (block level) quotations.	Best practice
	Indentation: The `<blockquote>` element SHOULD NOT be used for visual styling (indentation) alone.	Best practice
	Inline Quotations: The `<q>` element (for inline quotations) SHOULD NOT be used as the only way to designate quotations, due to poor support in screen readers and some browsers.	Best practice
Strikethrough/Delete and Insert	**Strikethrough Markup:** Strikethrough text SHOULD be marked with the `<del>` element.	Best practice
	Strikethrough Supplemental Text: Critical strikethrough text MUST be supplemented with a text-based method to convey the meaning of the strikethrough.	Best practice
	Insert Markup: Text designated for insertion SHOULD be marked with the `<ins>` element.	Best practice
	Insert Supplemental Text: Critical text designated for insertion MUST be supplemented with a text-based method to convey the meaning of the insertion.	Best practice

Section 2: Links and Navigation

Links

Topic	Technique	WCAG AA
Semantic Markup and Purpose	**Link Markup:** Links MUST be semantically designated as such. • **Note:** Ideally this means using an `<a>` element with a valid `href` value. In rare problematic cases, setting `role="link"` (and adding all other aspects of link functionality) may be acceptable.	Required WCAG 4.1.2
	Links Versus Buttons: Links SHOULD be used only for actions that take the user to other locations and SHOULD NOT be used for button-like functionality. • **Note 1:** "Other locations" means other web pages, or to other locations in the same web page. Typically, the URL will change after activating a link. • **Note 2:** "Button-like functionality" means scripted features, including submitting forms.	Best practice
Link Text	**Discernible Text:** A link MUST have programmatically discernible text, as determined by the accessible name calculation algorithm.	Required WCAG 4.1.2
	Distinguishable Link Purpose: The purpose of each link MUST be understandable and distinguishable from other links on the same page, either from the link text alone (ideally), or from the immediate surrounding context of the link.	Required WCAG 2.4.4
	Avoid "Link" (or Similar) in the Link Text: The link text SHOULD NOT state its semantic role (e.g. don't say "link to..."), because screen readers already state the role before reading the link text.	Best practice
	Consistent Link Text Across Pages: Links to the same destinations MUST be consistently identified with the same (or very similar) link text across all pages of the site.	Required WCAG 3.2.4
	Links Opening in New Tab or Window: A link that opens in a new window or tab SHOULD indicate that it opens in a new window or tab.	Best practice
	Links to Non-HTML Files: A link to a file or destination in a non-HTML format (e.g. MS Word, PDF, plain text, etc.) SHOULD indicate the type of file or destination.	Best practice
	Links to External Sites: A link to an external site MAY indicate that it leads to an external site.	Best practice
Keyboard Accessibility	**Keyboard-Focusable:** Links MUST be keyboard-focusable.	Required WCAG 2.1.1
	Keyboard Activation: Links MUST activate with the enter/return key.	Required WCAG 2.1.1

Links *(continued)*

Topic	Technique	WCAG AA
Focus Order	**Focus Order:** The navigation order of focusable elements (links, form elements, etc.) MUST be logical and intuitive.	Required WCAG 2.4.3
	Tabindex: The `tabindex` attribute SHOULD NOT be used with positive values to customize the tab order (e.g. don't use `tabindex="1"`).	Best practice
Link Colors, Contrast, and Styles	**Links Visually Distinguishable from Non-Links:** Links MUST be visually distinguishable from surrounding non-link text.	Required WCAG 1.4.1
	Color as a Way to Visually Distinguish Links: Color alone MUST NOT be used as the only way to distinguish links from surrounding text unless the color contrast between the link and the surrounding text is at least 3 to 1 and an additional differentiation (e.g. underline, outline, etc.) is provided when the link is hovered or receives focus.	Required WCAG 1.4.1
	Link Contrast: Links MUST have a contrast ratio of 4.5 to 1 (for small text) or 3 to 1 (for large text) against their background.	Required WCAG 1.4.3
Target Size	**Target Size:** Links SHOULD measure a minimum of 44px by 44px. • **Note:** Inline links in paragraphs or blocks of text MAY be smaller.	Best practice
Visual Focus Indicator	**Focus Indicator:** All links MUST show a visual focus indicator when in focus.	Required WCAG 2.4.7
	Enhanced Focus Indicator: Links MAY have enhanced visual focus indicator styles.	Best practice
	Focus Indicator Contrast: The contrast of all visual focus indicators against the background MUST be at least 3 to 1.	Required WCAG 1.4.11 (WCAG 2.1)
Visual Hover Indicator	**Enhanced Hover Indicator:** Links SHOULD have enhanced visual hover indicator styles.	Best practice
	Hover Cursor Style: On hover over a link, the mouse cursor SHOULD appear as the pointer style, to provide a visual confirmation of the link role. • **Note:** On most browsers, the pointer style is represented by an icon of a hand with the index finger pointing forward.	Best practice
In-Page Navigation	**Skip Navigation:** A keyboard-functional "skip" link SHOULD be provided to allow keyboard users to navigate directly to the main content. • **Note 1:** The "skip" link SHOULD be the first focusable element on the page. • **Note 2:** The "skip" link MUST be either visible at all times or visible on keyboard focus. • **Note 3:** "Skip" links, landmarks (e.g. `<nav>`, `<main>`, etc.), and page table of contents are each valid and sufficient methods to satisfy the WCAG 2.4.1 requirement to allow users to bypass blocks of content. Ideally the design would include all of these techniques, where appropriate.	Required WCAG 2.4.1 (This technique is one way to satisfy this requirement)

Links *(continued)*

Topic	Technique	WCAG AA
In-Page Navigation *(continued)*	**Page Table of Contents:** A table of contents for the page MAY be included at the top of the content or in the header. • **Note:** Ideally, the links in the table of contents SHOULD correspond to the heading structure in the page content.	Required WCAG 2.4.1 (This technique is one way to satisfy this requirement)

Site Navigation

Topic	Technique	WCAG AA
Consistency	**Consistent Navigation Patterns:** Navigation patterns that are repeated on web pages MUST be presented in the same relative order each time they appear and MUST NOT change order when navigating through the site.	Required WCAG 3.2.3
	Consistent Identification: Elements such as labels, names, and text alternatives for content that have the same functionality across multiple screens must be consistently identified.	Required WCAG 3.2.4
	Consistent Help: Certain help mechanisms that are repeated on multiple web pages MUST occur in the same relative order to other page content, unless a change is initiated by the user.	Required WCAG 3.2.6 (WCAG 2.2)
Multiple Ways	**Multiple Ways:** Multiple ways must be available to find other web pages on the site — at least two of: a list of related pages, table of contents, site map, site search, or list of all available web pages.	Required WCAG 2.4.5
Navigation Lists	**Markup:** A navigation list SHOULD be designated with the `<nav>` element or `role="navigation"`. See also the ***Landmarks*** section.	Best practice
	Visible "You Are Here" Indicator: A navigation list SHOULD include a visible method of informing users which page within the navigation list is the currently active/visible page.	Best practice
	Non-visual "You Are Here" Indicator: A navigation list SHOULD include a method available to assistive technologies of informing non-visual users which page within the navigation list is the currently active/visible page.	Best practice

Within-Page Navigation

Topic	Technique	WCAG AA
Methods to Bypass Blocks of Content	**Bypass Blocks:** A method MUST exist to bypass repeated blocks of content. Possible techniques applicable to almost all pages include skip links, ***headings***, and ***landmarks***. It's best to use all of these techniques, if possible. Other techniques include page-specific table of contents links and expandable/collapsible regions.	Required WCAG 2.4.1
Reading/Focus Order	**Reading Order:** The reading order MUST be logical and intuitive.	Required WCAG 1.3.2
	Focus Order: The navigation order of focusable elements MUST be logical and intuitive.	Required WCAG 2.4.3
Skip Links	**Provide a Skip Link:** A keyboard-functional "skip" link SHOULD be provided to allow keyboard users to navigate directly to the main content.	Best practice
	First Focusable Element: The "skip link" SHOULD be the first focusable element on the page.	Best practice
	Skip Link Visibility: A skip link MUST be either visible at all times or visible on keyboard focus.	Best practice
Table of Contents	**Table of Contents Links:** A table of contents for the page MAY be included at the top of the content or in the header.	Best practice
	Reflect the Heading Structure: If a table of contents for the page is included, it SHOULD reflect the heading structure of the page.	Best practice
Paginated Views	**Visual "You Are Here" Indicator:** A paginated view SHOULD include a visible method of informing users which view is the currently active/visible view.	Best practice
	Non-Visual "You Are Here" Indicator: A paginated view SHOULD include a method available to assistive technologies of informing non-visual users which view is the currently active/visible view.	Best practice
Single-Key Shortcuts	**Single-Key Shortcuts:** If a single-character-key shortcut exists, then at least one of the following MUST be true: single-character-key shortcuts can be turned off, remapped, or are only active when the relevant user interface component is in focus.	Required WCAG 2.1.4 (WCAG 2.1)

Reading Order / Focus Order

Topic	Technique	WCAG AA
Reading Order	**Reading Order:** The reading order MUST be logical and intuitive. • **Note:** The default reading order is determined by the order of the elements in the DOM.	Required WCAG 1.3.2
Focus Order	**Focus Order:** The navigation order of focusable elements MUST be logical and intuitive. • **Note 1:** Focusable elements include links, form inputs and controls, buttons, and any element with a `tabindex` value of `0` or greater. • **Note 2:** The default reading order is determined by the order of the focusable elements in the DOM. • **Note 3:** Use the tab key to navigate forward through focusable elements, and shift + tab to navigate backward.	Required WCAG 2.4.3
Tabindex	**Positive Values:** A `tabindex` of positive values (e.g. `tabindex="1"`, `tabindex="2"`, etc.) SHOULD NOT be used, because the result is almost always an illogical reading and/or tab order.	Best practice
	Zero: Use `tabindex="0"` to make a custom widget or control focusable, if it is not already focusable. • **Note:** Use natively focusable elements whenever possible (e.g. links, form inputs and controls, buttons), rather than custom controls, for simplicity in markup and implementation.	Best practice
	Negative 1: Use `tabindex="-1"` to allow JavaScript to send the focus to an element, without putting it in the focus order.	Best practice

Section 3: Images and Visual Design

Images

Topic	Technique	WCAG AA
Informative Images and Active Images (e.g. Links, Buttons, or Controls)	**Alternative Text:** The image MUST have alternative text. Refer to the techniques for various image types: • **images using <img>** • **active images** • **form input images** • **SVG using <img>** • **SVG using <svg>** • **HTML5 canvas** • **icon fonts**	Required WCAG 1.1.1
	Meaningful Description: Alternative text MUST be meaningful (accurately conveying the purpose of the image, and the author's intent in a way that is useful to those who cannot see the image). • **Note 1:** Image links SHOULD describe the link destination. • **Note 2:** Button/control links SHOULD describe the purpose and/or resulting action of the button or control.	Required WCAG 1.1.1
	Concise: The length of the alternative text for informative images SHOULD be concise (no more than about 250 characters).	Best practice
	Avoid Restating that the Element Is an Image: Alternative text SHOULD NOT include words that identify the element as a graphic or image (e.g. avoid phrases like "graphic of," or "image of," etc.), because screen readers already state the role (e.g. by saying "graphic" then reading the alternative text).	Best practice
Unessential Images (Purely Decorative or Redundant, and Not Active)	**Unessential Images:** Images that do not convey content, are decorative, or are redundant to content that is already conveyed in text SHOULD be given null (empty) alternative text (`alt=""`), ARIA `role="presentation"`, or implemented as CSS backgrounds.	Best practice
Complex Images	**Complex Images:** Complex images MUST be briefly described using `alt` text AND MUST have a more complete extended description. • **Note:** It is not "wrong" to provide descriptive alternative text in these instances, but it is highly discouraged if the image is truly unessential.	Required WCAG 1.1.1
Background Images	**Informative Background Images:** If a background image conveys information, alternative text MUST be provided (e.g. via regular visible text, by adding `role="image"` and `aria-label`, or by other means).	Required WCAG 1.1.1

Images *(continued)*

Topic	Technique	WCAG AA
Background Images *(continued)*	**Active Background Images:** If a background image is the only "content" in an active element (e.g. a link, button, or control), the active element MUST have an accessible name (e.g. via `aria-label` or similar).	Required WCAG 1.1.1
	Color Contrast of Small Text: Small text (under 18 point regular font or 14 point bold font) MUST have a contrast ratio of at least 4.5 to 1 with the background.	Required WCAG 1.4.3
	Color Contrast of Large Text: Large text (at or over 18 point or 14 point bold) MUST have a contrast ratio of at least 3 to 1 with the background.	Required WCAG 1.4.3
	Unessential Background Images: Alternative text SHOULD NOT be provided for unessential background images.	Best practice
Images of Text	**No Images of Text:** An image MUST NOT include informative text if an equivalent visual presentation of the text can be rendered using real text (unless the text is essential — such as a logo — or the font, size, color, and background are customizable.).	Required WCAG 1.4.5
	Color Contrast of Small Text: Small text (under 18 point regular font or 14 point bold font) MUST have a contrast ratio of at least 4.5 to 1 with the background.	Required WCAG 1.4.3
	Color Contrast of Large Text: Large text (at or over 18 point or 14 point bold) MUST have a contrast ratio of at least 3 to 1 with the background.	Required WCAG 1.4.3
Animated Images	See the requirements for ***Animation, Motion, and Timed Content***.	Required (Multiple)
Image CAPTCHA	**Alternative text:** Image CAPTCHA elements MUST have alternative text describing the purpose of the image.	Required WCAG 1.1.1
	Sensory Alternatives: Image CAPTCHA elements MUST be supplemented with at least one auditory alternative method.	Required WCAG 1.1.1
	Deafblind Access: At least one text-based alternative SHOULD be provided that allows users who are both deaf and blind to pass the CAPTCHA. • **Note:** Deafblind users typically use screen readers to convert text to braille, using a refreshable braille device.	Best practice

Color and Contrast

Topic	Technique	WCAG AA
Use of Color	**Use of Color:** Any information conveyed by color MUST be accompanied by a programmatically discernible text alternative.	Required WCAG 1.4.1
	Visible Alternative: Any information conveyed by color MUST be accompanied by a visible alternative (text, image, etc.) that does not depend on color for meaning.	Required WCAG 1.4.1
Color Contrast	**Small Text Contrast:** Small text (under 18 point regular font or 14 point bold font) MUST have a contrast ratio of at least 4.5 to 1 with the background. **Note:** The contrast rule also applies to images of text, even though images of text are discouraged.	Required WCAG 1.4.3
	Large Text Contrast: Large text (at or over 18 point or 14 point bold) MUST have a contrast ratio of at least 3 to 1 with the background. **Note:** The contrast rule also applies to images of text, even though images of text are discouraged.	Required WCAG 1.4.3
	UI Component Contrast: The contrast of UI control boundaries compared to adjacent areas MUST be at least 3 to 1 to distinguish the UI control from the adjacent areas.	Required WCAG 1.4.11 (WCAG 2.1)
	Visual Focus Indicator Contrast: The contrast of all visual focus indicators against the background MUST be at least 3 to 1.	Required WCAG 1.4.11 (WCAG 2.1)
High Contrast Mode	**Retain Visible Information:** Web content SHOULD retain all essential visual information in Windows High Contrast Mode.	Best practice
	Don't Override: The design SHOULD NOT override Windows High Contrast Mode.	Best practice

Text Styles, Resize, Reflow, and Zoom

Topic	Technique	WCAG AA
Text Resize and Reflow	**Resize Text 200%:** The page SHOULD be functional when only the text is magnified to 200% of its initial size.	Required WCAG 1.4.4
	Mobile Zoom: The page MUST allow users to zoom in on mobile devices (The following is NOT allowed: `<meta name="viewport" content="user-scalable=no">`)	Required WCAG 1.4.4
	One Scroll Direction Only: Content MUST NOT require scrolling in two directions (both vertically and horizontally) — even when the viewport is set or zoomed to 320 CSS pixels wide (for vertically scrolling content) or 256 CSS pixels high (for horizontally scrolling content) — unless both scrolling directions are essential to the usage or meaning of the content.	Required WCAG 1.4.10 (WCAG 2.1)
Text in Images	**No Images of Text:** An image MUST NOT include informative text if an equivalent visual presentation of the text can be rendered using real text (unless the text is essential — such as a logo — or the font, size, color, and background are customizable).	Required WCAG 1.4.5
	Color Contrast of Small Text: Small text (under 18 point regular font or 14 point bold font) MUST have a contrast ratio of at least 4.5 to 1 with the background.	Required WCAG 1.4.3
	Color Contrast of Large Text: Large text (at or over 18 point or 14 point bold) MUST have a contrast ratio of at least 3 to 1 with the background.	Required WCAG 1.4.3
Text/Paragraph Styles	**Full Functionality with Altered Text Styles:** In content implemented using markup languages that support the following text style properties, no loss of content or functionality occurs by setting all of the following and by changing no other style property: • Line height (line spacing) to at least 1.5 times the font size • Spacing following paragraphs to at least 2 times the font size • Letter spacing (tracking) to at least 0.12 times the font size • Word spacing to at least 0.16 times the font size.	Required WCAG 1.4.12 (WCAG 2.1)
	Line Justification: Text SHOULD NOT be full justified.	Best practice
	Column Width: The number of characters or glyphs per line in any section or column of text SHOULD NOT exceed 80 (40 characters in Chinese, Japanese, or Korean)	Best practice
	Font Legibility: Fonts SHOULD be easily readable by sighted users.	Best practice

Text Styles, Resize, Reflow, and Zoom *(continued)*

Topic	Technique	WCAG AA
Color Contrast	See the requirements for ***Color and Contrast***.	Required (Multiple)
CSS-Generated Content	**Avoid CSS-Generated Content:** CSS-generated content SHOULD be avoided (unless it is for presentation/decorative purposes only).	Best practice
	Text Alternative for CSS-Generated Content: A text alternative for informative CSS-generated content MUST be provided, AND the CSS-generated text SHOULD be set to `aria-hidden="true"`.	Required WCAG 1.3.1
	Decorative CSS-Generated Content: Decorative or redundant CSS-generated content SHOULD be set to `aria-hidden="true"`.	Best practice
Emphasis and Highlighting	**Emphasis Through Visual Styling:** Critical emphasis in the text SHOULD be conveyed through visual styling.	Best practice
	Emphasis in Text-Based Format: Critical emphasis in the text SHOULD be conveyed in a text-based format.	Best practice
	Highlighting Markup: Highlighted text SHOULD be marked with the `<mark>` element.	Best practice
	Text-Based Highlighting: Critical highlighted text SHOULD be supplemented with a text-based method to convey the meaning of the highlighting.	Best practice
Quotations	**Blockquote:** The `<blockquote>` element SHOULD be used to designate long (block level) quotations.	Best practice
	Indentation: The `<blockquote>` element SHOULD NOT be used for visual styling (indentation) alone.	Best practice
	Inline Quotations: The `<q>` element (for inline quotations) SHOULD NOT be used as the only way to designate quotations, due to poor support in screen readers and some browsers.	Best practice
Strikethrough/Delete and Insert	**Strikethrough Markup:** Strikethrough text SHOULD be marked with the `<del>` element.	Best practice
	Strikethrough Supplemental Text: Critical strikethrough text MUST be supplemented with a text-based method to convey the meaning of the strikethrough.	Best practice
	Insert Markup: Text designated for insertion SHOULD be marked with the `<ins>` element.	Best practice
	Insert Supplemental Text: Critical text designated for insertion MUST be supplemented with a text-based method to convey the meaning of the insertion.	Best practice

Visual Cues

Topic	Technique	WCAG AA
Visual Meaning	**Visual Meaning:** Content MUST NOT rely solely on visual characteristics such as shape, size, visual location, or orientation to convey meaning.	Required WCAG 1.3.3
Color	See the requirements for ***Color and Contrast***.	Required (Multiple)
Visual Layout	**Visual Separation of Content Blocks:** Blocks of content SHOULD be visually separated and distinct from each other, via margins, padding, or other methods of achieving visual "white space."	Best practice
	Label Proximity: Labels SHOULD be visually adjacent to their controls.	Best practice
	One Visual Focal Point: The layout SHOULD have only one main visual focal point.	Best practice
	Draw Attention: The design SHOULD draw attention to the intended visual focus point.	Best practice

Adaptive and Responsive Output

Topic	Technique	WCAG AA
Device Orientation	**Horizontal or Vertical Orientation:** Content MUST NOT restrict its view and operation to a single display orientation, such as portrait or landscape, unless a specific display orientation is essential.	Required WCAG 1.3.4 (WCAG 2.1)
Adaptive/ Responsive Text	See the requirements for ***Text Styles, Resize, Reflow, and Zoom***.	Required (Multiple)
Other Adaptive/ Responsive Elements	Content that would cause scrolling in two directions (both horizontally and vertically) MUST respond to the viewport size or zoom state by resizing and/or reflowing, at viewport resolutions of up to 320px for vertically scrolling content, and 256px for horizontally scrolling content. • **Images** • **Forms** • **Tables** • **Objects and Plugins** • **UI Components** • **Video**	Required WCAG 1.4.10 (WCAG 2.1)
Simplification	**Simplified Content:** Features of the content MAY be simplified, reduced in size, or eliminated when magnified or when viewed on small viewports.	Best practice
	Simplified User Interface: Features of the interface SHOULD be simplified, reduced in size, or eliminated when magnified or when viewed on small viewports.	Best practice

Section 4: Multimedia, Animations, and Motion

Audio and Video

Topic	Technique	WCAG AA
Prerecorded Audio-Only	**Transcript:** Prerecorded audio-only content MUST have a transcript of dialog, narration, and other meaningful sounds.	Required WCAG 1.2.1
	Autoplay: A method MUST be provided to pause, stop, or hide any media content that begins playing automatically and which lasts 5 seconds or more.	Required WCAG 2.2.2
	Audio Control: A mechanism MUST be provided to stop, pause, mute, or adjust volume for audio that automatically plays on a page for more than 3 seconds.	Required WCAG 1.4.2
	Sign Language: Prerecorded audio-only content MAY include sign language interpretation.	Best practice
	Background Sounds: Background sounds in prerecorded audio-only content SHOULD be minimized (20dB lower than foreground sounds, except for occasional sounds of no more than 2 seconds) or eliminated during narration or dialog, or a method must be available to turn off background sounds.	Best practice
Prerecorded Video-Only (and Video-Like Animations)	**Alternative Text:** Prerecorded video-only content MUST include a text description. This may be in regular text (e.g. in a paragraph), and it can be short or long, depending on the nature of the video.	Required WCAG 1.2.1
	Audio Description: Prerecorded video-only content MUST include an audio description (narrated description) of important visual content.	Required WCAG 1.2.5
	Autoplay: A method MUST be provided to pause, stop, or hide any media content that begins playing automatically and which lasts 5 seconds or more.	Required WCAG 2.2.2
	Flashing Content: A page MUST NOT contain content that flashes more than 3 times per second unless that flashing content is sufficiently small, the flashes are of low contrast, and do not violate general flash thresholds.	Required WCAG 2.3.1
Prerecorded Multimedia	**Alternative Text:** Prerecorded multimedia content SHOULD include a text description of important visual content that is not conveyed through the audio (this can be included in the transcript).	Best practice
	Transcript: Prerecorded multimedia content SHOULD have a text transcript of dialog, narration, and other meaningful sounds.	Best practice
	Captions: Prerecorded multimedia content MUST have synchronized captions.	Required WCAG 1.2.2
	Audio Description: Prerecorded multimedia content MUST include an audio description (narrated description) of important visual content.	Required WCAG 1.2.5

Audio and Video *(continued)*

Topic	Technique	WCAG AA
Prerecorded Multimedia *(continued)*	**Autoplay:** A method MUST be provided to pause, stop, or hide any media content that begins playing automatically and which lasts 5 seconds or more.	Required WCAG 2.2.2
	Audio Control: A mechanism MUST be provided to stop, pause, mute, or adjust volume for audio that automatically plays on a page for more than 3 seconds.	Required WCAG 1.4.2
	Flashing Content: A page MUST NOT contain content that flashes more than 3 times per second unless that flashing content is sufficiently small, the flashes are of low contrast, and do not violate general flash thresholds.	Required WCAG 2.3.1
	Sign Language: Prerecorded multimedia content MAY include sign language interpretation.	Best practice
	Background Sounds: Background sounds in prerecorded audio-only and prerecorded multimedia content SHOULD be minimized (20dB lower than foreground sounds, except for occasional sounds of no more than 2 seconds) or eliminated during narration or dialog, or a method must be available to turn off background sounds.	Best practice
Live Audio-Only	**Audio Control:** A mechanism MUST be provided to stop, pause, mute, or adjust volume for audio that automatically plays on a page for more than 3 seconds.	Required WCAG 1.4.2
	Autoplay: A method MUST be provided to pause, stop, or hide any media content that begins playing automatically and which lasts 5 seconds or more.	Required WCAG 2.2.2
	Background Sounds: Background sounds in live audio-only content SHOULD be minimized (20dB lower than foreground sounds, except for occasional sounds of no more than 2 seconds) or eliminated during narration or dialog, or a method must be available to turn off background sounds.	Best practice
	Captions: Live audio-only content SHOULD have synchronized captions.	Best practice
Live Video-Only (and Video-Like Animations)	**Autoplay:** A method MUST be provided to pause, stop, or hide any media content that begins playing automatically and which lasts 5 seconds or more.	Required WCAG 2.2.2
	Flashing Content: Live video-only content MUST NOT contain content that flashes more than 3 times per second unless that flashing content is sufficiently small, the flashes are of low contrast, and do not violate general flash thresholds.	Required WCAG 2.3.1
	Alternative Text: Live video-only content MUST include a text description. This may be in regular text (e.g. in a paragraph), and it can be short or long, depending on the nature of the video.	Required WCAG 1.2.1

Audio and Video *(continued)*

Topic	Technique	WCAG AA
Live Video-Only *(continued)*	**Audio Description:** Live video-only content SHOULD include an audio description (narrated description) of important visual content.	Best practice
Live Multimedia	**Captions:** Live multimedia content MUST have synchronized captions.	Required WCAG 1.2.4
	Audio Control: A mechanism MUST be provided to stop, pause, mute, or adjust volume for audio that automatically plays on a page for more than 3 seconds.	Required WCAG 1.4.2
	Autoplay: A method MUST be provided to pause, stop, or hide any media content that begins playing automatically and which lasts 5 seconds or more.	Required WCAG 2.2.2
	Background Sounds: Background sounds in live multimedia content SHOULD be minimized (20dB lower than foreground sounds, except for occasional sounds of no more than 2 seconds) or eliminated during narration or dialog, or a method must be available to turn off background sounds.	Best practice
	Flashing Content: A page MUST NOT contain content that flashes more than 3 times per second unless that flashing content is sufficiently small, the flashes are of low contrast, and do not violate general flash thresholds.	Required WCAG 2.3.1
	Audio Description: Live multimedia content SHOULD include an audio description (narrated description) of important visual content.	Best practice
Page Background Sounds	**Audio Control:** A mechanism MUST be provided to stop, pause, mute, or adjust volume for audio that automatically plays on a page for more than 3 seconds.	Required WCAG 1.4.2
	Autoplay: A method MUST be provided to pause, stop, or hide any media content that begins playing automatically and which lasts 5 seconds or more.	Required WCAG 2.2.2
	If Informative: If the background sounds contain informative content, all of the requirements for prerecorded audio-only content apply (see above).	Depends
Page Background Video	**Autoplay:** A method MUST be provided to pause, stop, or hide any media content that begins playing automatically and which lasts 5 seconds or more.	Required WCAG 2.2.2
	Flashing Content: A page MUST NOT contain content that flashes more than 3 times per second unless that flashing content is sufficiently small, the flashes are of low contrast, and do not violate general flash thresholds.	Required WCAG 2.3.1
	See the requirements for **color contrast** (in the ***Color and Contrast*** section) (for text and UI elements overlayed on top of video).	Required WCAG 1.4.3

Audio and Video *(continued)*

Topic	Technique	WCAG AA
Page Background Video *(continued)*	**If Informative:** If the background video contains informative content, all of the requirements for prerecorded video-only content apply (see above).	Depends
Media Players	Media players themselves MUST be fully accessible (in particular to keyboard users and screen reader users). See the requirements for: • **Media player keyboard accessibility** • **Media player screen reader accessibility** • **Captions, transcripts, audio descriptions within the media player** • **User customization options within the media player** See also the requirements for ***Form Inputs, Labels, and Instructions*** See also the requirements for ***Custom JavaScript/ARIA Components***	Required (Multiple)

Animation, Motion, and Timed Content

Topic	Technique	WCAG AA
Audio and Video	See the requirements for ***Audio and Video***.	Required (Multiple)
Flashing	**Flashing:** A page MUST NOT contain content that flashes more than 3 times per second unless that flashing content is sufficiently small, the flashes are of low contrast, and do not violate general flash thresholds.	Required WCAG 2.3.1
Parallax Effects (backgrounds or foregrounds that move separately from each other)	**Keyboard-Accessible:** All content and features within parallax scrolling content MUST be accessible by keyboard.	Required WCAG 2.1.1
	Text Color Contrast: The contrast of the text against all parts of a moving background MUST be a minimum of 4.5 to 1 for small text or 3 to 1 for large or bold text.	Required WCAG 1.4.3
	UI Component Color Contrast: The contrast of UI control boundaries compared to adjacent areas SHOULD be 3 to 1 to distinguish the UI control from the adjacent areas.	Required WCAG 1.4.11 (WCAG 2.1)
User Control of Timing	**Autoplay:** A method MUST be provided to pause, stop, or hide any media content that begins playing automatically and which lasts 5 seconds or more.	Required WCAG 2.2.2
	Time Limits: A method MUST be provided to allow users to control time limits by turning them off, adjusting them, or extending them (unless the timing is essential or if the time limit is longer than 20 hours).	Required WCAG 2.2.1
	Replay Option: A method SHOULD be provided to allow users to replay timed content that is finished or expired (unless replaying the content fundamentally alters its purpose or meaning).	Best practice

Section 5: User Input, Forms, and Dynamic Content

Device-Independent User Input

Topic	Technique	WCAG AA
Keyboard	**Focusable with Tab Key:** Links, buttons, and interactive controls MUST be keyboard-focusable.	Required WCAG 2.1.1
	Logical Tab Order: The navigation order of focusable elements MUST be logical and intuitive.	Required WCAG 2.4.3
	No Positive tabindex Values: `tabindex` of positive values SHOULD NOT be used.	Best practice
	Visual Focus Indicator: All focusable elements MUST have a visual focus indicator when in focus.	Required WCAG 2.4.7
	Enhanced Visual Focus Indicator: Focusable elements SHOULD have enhanced visual focus indicator styles.	Best practice
	Color Contrast of Visual Focus Indicator: The contrast of all large visual focus indicators (at least 3px by 3px) against the background) SHOULD be at least 3 to 1.	Required WCAG 1.4.11 (WCAG 2.1)
	Focus Not Obscured (Minimum): When a user interface component receives keyboard focus, the component MUST NOT be entirely hidden due to author-created content.	Required WCAG 2.4.11 (WCAG 2.2)
	Focus Not Obscured (Enhanced): When a user interface component receives keyboard focus, the focus indicator SHOULD NOT be hidden at all (even partially) by author-created content.	Best practice
	Focus Appearance: When the keyboard focus indicator is visible, an area of the focus indicator SHOULD meet specified size and color contrast requirements.	Best practice
	Keyboard Functionality: Functionality MUST be available using the keyboard, unless the functionality cannot be accomplished in any known way using a keyboard.	Required WCAG 2.1.1
	Keyboard Traps: Keyboard focus MUST NOT be locked or trapped in a particular page element and the user MUST be able to navigate to and from all navigable page elements using only a keyboard.	Required WCAG 2.1.2
	Keyboard Shortcuts: Page-specified shortcut keys and accesskeys MUST NOT conflict with existing keyboard shortcuts in the browser, operating system, or assistive technologies.	Required WCAG 2.1.4 (WCAG 2.1)
	Custom Keystroke Instructions: When custom keyboard behavior is required to use a component, keyboard instructions MUST be provided.	Required WCAG 3.3.2
	ARIA Widget Instructions: ARIA widgets that require more than just the Tab key to interact with may be confusing to users (even when the widgets follow the WAI-ARIA Authoring Practices), so you MAY provide keyboard instructions.	Best practice

Device-Independent User Input *(continued)*

Topic	Technique	WCAG AA
Keyboard Focus Management During Interactions	**Set Keyboard Focus:** The focus MUST be purposely moved or set (via JavaScript) onto the appropriate element when the user's action requires a change of context or location for effective keyboard or touch interaction.	Required WCAG 2.4.3
	No Lost Focus: The focus MUST NOT become lost or reset to the top of the page, except when loading or re-loading a page.	Required WCAG 2.4.3
	Focus Target Has Text: When moving or setting focus, the destination element MUST contain discernible text (either standard text or programmatically associated text).	Required WCAG 1.3.1
Mouse	**Click Target Size (Minimum):** The size of the target for pointer inputs MUST be at least 24 by 24 CSS pixels (with some allowable exceptions). The allowable exceptions for this rule are: • **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	Required WCAG 2.5.8 (WCAG 2.2)
	Click Target Size (Enhanced): The click target size SHOULD be at least 44 by 44 CSS pixels. **Note:** Allowed exceptions include the following circumstances: • The target is available through an equivalent control of at least 44 by 44 CSS pixels • The target is inline (in a sentence or block of text) • The target size is user-customizable • The smaller target size is essential to the information or functionality	Best practice
	Visual Hover Indicator: An enhanced visual hover indicator SHOULD be provided.	Best practice
	Mouse Cursor Icon on Hover: The mouse cursor icon on hover SHOULD correspond to the type of action allowed by the element.	Best practice
	Pointer Cancellation: For functionality that can be operated using a single pointer, at least one of the following MUST be true: no down-event, can abort/undo, up reversal, or essential.	Required WCAG 2.5.2 (WCAG 2.1)

Device-Independent User Input *(continued)*

Topic	Technique	WCAG AA
Mouse *(continued)*	**Dragging Movements:** All functionality that uses a dragging movement for operation MUST BE achievable 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.	Required WCAG 2.5.7 (WCAG 2.2)
Voice	**Visual Labels Match Programmatic Labels:** The visual labels for links, buttons, form elements, and other controls SHOULD match the programmatic labels (to allow easy and intuitive voice commands).	Required WCAG 2.5.3 (WCAG 2.1)
	All the **keyboard requirements** above apply (***Device-Independent User Input***).	Required (Multiple)
Touch	**Touch Functionality:** Functionality MUST be available using standard touch methods, unless the functionality cannot be accomplished in any known way using a touch device.	Required WCAG 2.5.1 (WCAG 2.1)
	Touch Functionality with Screen Reader Enabled: Functionality MUST be available using screen reader touch methods (e.g. single tap or double tap actions), unless the functionality cannot be accomplished in any known way using a touch device. **Note:** The touch actions with the screen reader turned on are completely different from the touch actions when the screen reader is turned off.	Required WCAG 2.5.1 (WCAG 2.1)
	Single Pointer: Functionality MUST work with a single pointer (e.g. a single finger), unless multipoint activation is essential.	Required WCAG 2.5.1 (WCAG 2.1)
	Touch Cancellation: Touch events MUST NOT be activated on a down event (use up events instead), to allow users to cancel, abort, or undo touch events, unless the down event is essential.	Required WCAG 2.5.2 (WCAG 2.1)
	Focus Trap: Touch/gesture focus MUST NOT be locked or trapped in a particular page element and the user MUST be able to navigate to and from all navigable page elements using standard touch actions.	Required WCAG 2.1.2
	Touch Target Size (Minimum): The size of the target for pointer inputs MUST be at least 24 by 24 CSS pixels (with some allowable exceptions). The allowable exceptions for this rule are: • **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.	Required WCAG 2.5.8 (WCAG 2.2)

Device-Independent User Input *(continued)*

Topic	Technique	WCAG AA
Touch *(continued)*	**Touch Target Size (Enhanced):** The click/touch target size SHOULD be at least 44 by 44 CSS pixels, unless at least one of the following is true: • There is an equivalent control on the same page that is at least 44 by 44 CSS pixels, OR • The target is inline (in a sentence or block of text), OR • The user agent controls the target size, OR • The smaller size of the target is essential to the information being conveyed.	Best practice
	Gesture-Only Functionality: Features MUST NOT depend on swipe or gesture motions as the only activation method.	Required WCAG 2.5.1 (WCAG 2.1)
	Pointer Gestures: All functionality that uses multipoint or path-based gestures for operation MUST be operable with a single pointer without a path-based gesture, unless a multipoint or path-based gesture is essential.	Required WCAG 2.5.1 (WCAG 2.1)
	Dragging Movements: All functionality that uses a dragging movement for operation MUST BE achievable 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.	Required WCAG 2.5.7 (WCAG 2.2)
Motion Actuation	**Motion Actuation:** Functionality that can be operated by device or user motion MUST also be operable by user interface components and MUST allow the ability to disable motion actuation.	Required WCAG 2.5.4 (WCAG 2.1)
	Kinetic Motion of Device: Features MUST NOT depend on kinetic motion of the device (e.g. shake, raise, lower, tilt).	Required WCAG 2.5.4 (WCAG 2.1)

Form Inputs, Labels, and Instructions

Topic	Technique	WCAG AA
Labels for Inputs	**Programmatic Labels:** Labels MUST be programmatically associated with their corresponding elements.	Required WCAG 1.3.1
	Discernible Label Text: Labels MUST be available as programmatically discernible text.	Required WCAG 1.3.1
	Meaningful Labels: Labels MUST be meaningful.	Required WCAG 1.3.1
	Sensory Dependencies of Labels: Labels MUST NOT rely solely on references to sensory characteristics.	Required WCAG 1.3.3
	Icons as Labels: Icons/graphics MAY be used as the only visual label (without visual text) only if the meaning of the icon is visually self-evident AND if there is a programmatically associated semantic label available to assistive technologies.	Required WCAG 1.3.1
	Placeholder Text: Placeholder text is allowed but MUST NOT be used as the only method of providing a label for a text input.	Required WCAG 1.3.1
	Visible Labels: Labels MUST be visible.	Required WCAG 3.3.2
	Label in Name: For user interface components with labels that include text or images of text, the name MUST contain the text that is presented visually.	Required WCAG 2.5.3 (WCAG 2.1)
	Matching Programmatic Label and Visual Label: The programmatic label MUST include the same text presented in the visual label, to facilitate voice activation.	Required WCAG 2.5.3 (WCAG 2.1)
	Multiple Labels for One Input: When multiple labels are used for one element, each label MUST be programmatically associated with the corresponding element.	Required WCAG 1.3.1
	One Label for Multiple Inputs: When one label is used for multiple elements, the label MUST be programmatically associated with each of the corresponding elements.	Required WCAG 1.3.1
	Visually Adjacent Labels: A label SHOULD be visually adjacent to its corresponding element.	Best practice
	Programmatically Adjacent Labels: A label SHOULD be adjacent in the DOM to its corresponding element.	Best practice
Labels for Groups of Inputs	**Programmatic Group Labels:** Group labels MUST be programmatically associated with the group if the individual labels for each element in the group are insufficient on their own (e.g. a group of radio buttons that has a group label plus individual labels for each radio option).	Required WCAG 1.3.1
	Programmatically Determinable Text in Group Labels: Group labels MUST be programmatically determinable.	Required WCAG 1.3.1
	Meaningful Group Labels: Group labels MUST be meaningful.	Required WCAG 1.3.1

Form Inputs, Labels, and Instructions *(continued)*

Topic	Technique	WCAG AA
Labels for Groups of Inputs *(continued)*	**Sensory Dependencies of Group Labels:** Group labels MUST NOT rely solely on references to sensory characteristics.	Required WCAG 1.3.3
	Visible Group Labels: Group labels MUST be visible.	Required WCAG 3.3.2
	Matching Programmatic Label and Visual Label: The programmatic label MUST include the same text presented in the visual label, to facilitate voice activation.	Required WCAG 2.5.3 (WCAG 2.1)
	Visually Adjacent Group Labels: Group labels SHOULD be visually adjacent to the grouped elements.	Best practice
	Programmatically Adjacent Group Labels: Group labels SHOULD be adjacent in the DOM to the grouped elements.	Best practice
Identify Input Purpose	**Identify Input Purpose:** The purpose for each common input field that collects an individual's personal data MUST be programmatically defined based on the list of 53 Input Purposes for User Interface Components.	Required WCAG 1.3.5 (WCAG 2.1)
Instructions About Inputs	**Programmatic Association of Input Instructions:** Instructions for an element MUST be programmatically associated with the element.	Required WCAG 3.3.2
	Programmatically Determinable Text in Input Instructions: Instructions for an element MUST be available as programmatically determinable text.	Required WCAG 3.3.2
	Meaningful Input Instructions: Instructions for an element MUST be meaningful.	Required WCAG 3.3.2
	Visible Input Instructions: Instructions for an element MUST be visible.	Required WCAG 3.3.2
	Sensory Dependencies of Input Instructions: Instructions for an element MUST NOT rely solely on references to sensory characteristics.	Required WCAG 1.3.3
	Hidden Input Instructions: If the instructions for an element are not critical, the instructions MAY be hidden until the user requests them.	Best practice
	Visually Adjacent Input Instructions: Instructions for an element SHOULD be visually adjacent to the element.	Best practice
	Programmatically Adjacent Input Instructions: Instructions for an element SHOULD be adjacent in the DOM to the element.	Best practice
Instructions About an Entire Form, a Group, or a Section	**Programmatic Association of Group Instructions:** Instructions for groups or sections SHOULD be programmatically associated with the group.	Required WCAG 3.3.2

Form Inputs, Labels, and Instructions *(continued)*

Topic	Technique	WCAG AA
Instructions About an Entire Form, a Group, or a Section *(continued)*	**Programmatically Determinable Text in Group Instructions:** Instructions for groups or sections MUST be programmatically determinable.	Required WCAG 3.3.2
	Meaningful Group Instructions: Instructions for groups or sections MUST be meaningful.	Required WCAG 3.3.2
	Visible Group Instructions: Instructions for groups or sections MUST be visible.	Required WCAG 3.3.2
	Sensory Dependencies of Group Instructions: Instructions for groups or sections MUST NOT rely solely on references to sensory characteristics.	Required WCAG 1.3.3
	Hidden Form Instructions: If the instructions are not critical to understand the purpose of a group or section, the instructions MAY be hidden until the user requests them.	Best practice
	Visually Adjacent Group Instructions: Instructions for groups or sections SHOULD be visually adjacent to the grouped elements.	Best practice
	Programmatically Adjacent Group Instructions: Instructions for groups or sections SHOULD be adjacent in the DOM to the grouped elements.	Best practice
Required Fields	**Programmatic Designation:** Required fields SHOULD be programmatically designated as such. **Note:** At a minimum, WCAG requires an informative error message about the field after the user submits the form.	Best practice
	Visual Indicator: Required fields SHOULD have a visual indicator that the field is required.	Best practice
Data Input Restrictions	**Information About Data Input Restrictions:** If a field allows only restricted input (e.g. a certain date format, only integers, no more than 4 characters, etc.), the restrictions SHOULD be communicated in the label or instructions. **Note:** At a minimum, WCAG requires an informative error message about the field after the user submits the form.	Best practice
Disabled Fields	**Awareness of Disabled Fields:** If awareness of a disabled field is essential to understanding the content, an alternative way of communicating information about the disabled field MUST be provided (because disabled fields are not in the normal tab order by default, making it difficult for screen reader users to discover them).	Required WCAG 1.3.1

Form Inputs, Labels, and Instructions *(continued)*

Topic	Technique	WCAG AA
Time Limits	**Sufficient Time:** Users MUST be allowed sufficient time to complete the form, through at least one of the following methods: • No time limit • The ability to turn off the time limit • The ability to extend the time limit • The ability to adjust/customize time limit • A minimum of 20 hours to complete the form **Note:** This requirement can be waived if the time limit is essential to the meaning or purpose of the form (e.g. a timed auction).	Required WCAG 2.2.1
Entering Redundant Information	**Redundant Entry:** Information previously entered by or provided to the user that is required to be entered again in the same process MUST be either auto-populated or available for the user to select, with several allowable exceptions. The allowable exceptions when redundant entry may be required are 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.	Required WCAG 3.3.7 (WCAG 2.2)
Authentication	**Accessible Authentication (Minimum):** A cognitive function test (such as remembering a password or solving a puzzle) MUST NOT be 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:** "Object recognition" and "Personal 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.	Required WCAG 3.3.8 (WCAG 2.2)

Form Inputs, Labels, and Instructions *(continued)*

Topic	Technique	WCAG AA
Authentication *(continued)*	**Accessible Authentication (Enhanced):** A cognitive function test (such as remembering a password or solving a puzzle) SHOULD NOT be required for any step in an authentication process, unless there is an alternative authentication method without a required cognitive test or there is a mechanism to assist the user in completing the cognitive function test.	Best practice
Dynamic Forms	See the requirements for ***Dynamic Content (JavaScript, AJAX)***.	Required (Multiple)
Custom Form Inputs (JavaScript/ARIA)	See the requirements for ***Custom Widgets (JavaScript/ARIA) Components***.	Required (Multiple)

Form Validation and Feedback

Topic	Technique	WCAG AA
Labels and Instructions for Error Prevention	See the requirements and recommendations for ***Form Inputs, Labels, and Instructions***, including: • Labels for inputs • Labels for groups of inputs • Instructions about inputs • Instructions about an entire form, a group, or a section • Required fields (in the full list of recommendations) • Data input restrictions (in the full list of recommendations) • Disabled fields • Time limits	Required (Multiple)
	Context-Sensitive Help: Context-sensitive help SHOULD be available.	Best practice
Critical Error Prevention	Web pages that process user input for any of the following: • legal commitments, • financial transactions, • user-controllable data (e.g. user profile, social media posts, OR • test/quiz responses MUST implement at least one of the following error prevention techniques: • **Reversible:** Submissions are reversible. • **Checked:** Data entered by the user is checked for input errors and the user is provided an opportunity to correct them. • **Confirmed:** A mechanism is available for reviewing, confirming, and correcting information before finalizing the submission.	Required WCAG 3.3.4
Error Prevention (All Circumstances)	All web pages that process user input SHOULD implement at least one of the following error prevention techniques: • **Reversible:** Submissions are reversible. • **Checked:** Data entered by the user is checked for input errors and the user is provided an opportunity to correct them. • **Confirmed:** A mechanism is available for reviewing, confirming, and correcting information before finalizing the submission.	Best practice

Form Validation and Feedback *(continued)*

Topic	Technique	WCAG AA

Error Detection on Submit	**Error Identification:** If an input error is automatically detected, the item that is in error is identified and the error is described to the user in text.
	Valid techniques include the following:
	• Add `aria-invalid="true"` to the input
	• Identify the input (referencing the label):
	  ○ in a simple JavaScript alert
	  ○ with information associated with the input via `aria-describedby` (widely supported) or `aria-errormessage` (not yet widely supported)
	  ○ with error text added to the input's label (other techniques are more semantically correct, but this is a reliable method)
	  ○ with text on the web page (it may be appropriate to move the keyboard focus to the error message)
	  ○ with an `aria-live` or `role="alert"` announcement
	  ○ with information about the error in the page `<title>` if the submission causes a page reload or a new page load.
		Required WCAG 3.3.1

Dynamic Error Detection	**Visible Real-Time Error Messages:** Real-time error messages MAY be scripted to show on the screen for sighted users, but attempts to announce the real-time messages to screen reader users can be problematic (see the next two rows below). It is usually acceptable to wait to announce real-time errors until after form submission, assuming that no data has been saved yet.	Best practice
	Live Announcements per Keystroke: ARIA live error messages SHOULD NOT be scripted to occur with every keystroke (to avoid overwhelming screen reader users), unless there is a delay built into the script to avoid announcements while the user is actively typing.	Best practice
	Live Announcements on Leaving a Field (on blur): `aria-live` can be used to announce error messages on blur, but this technique should be used with caution, because the `aria-live` announcement may conflict with the screen reader's attempt to read the next element which receives focus, or with the user's keystrokes, causing the aria-live announcement to be interrupted or not announced at all.	A possible, but sometimes problematic way to meet WCAG 3.3.1
Error Message Characteristics	**Error Suggestion:** If an input error is automatically detected and suggestions for correction are known, then the suggestions are provided to the user, unless it would jeopardize the security or purpose of the content.	Required WCAG 3.3.3
	Programmatically Associated: Error feedback SHOULD be programmatically associated with the appropriate element.	Best practice

Form Validation and Feedback *(continued)*

Topic	Technique	WCAG AA
Error Message Characteristics *(continued)*	**Meaningful Error Message:** Error feedback MUST clearly and accurately describe the error and/or how to fix the error.	Required WCAG 3.3.1 for error messages WCAG 3.3.3 for error suggestions
	Visible Error Message: Error feedback MUST be visible.	Required WCAG 3.3.1
Success Messages	**Success Confirmation:** The web page SHOULD confirm successful submission of data. Possible techniques include the following: • a simple JavaScript alert • with confirmation text on the web page (it may be appropriate to move the keyboard focus to the error message) • with an `aria-live` or `role="alert"` announcement • with the confirmation message in the page `<title>` if the submission causes a page reload or a new page load.	Best practice

Dynamic Content (JavaScript, AJAX)

Topic	Technique	WCAG AA
Context Changes	**Context Changes on Focus:** The focus MUST be purposely moved or set (via JavaScript) onto the appropriate element when the user's action requires a change of context or location for effective keyboard or touch interaction.	Required WCAG 3.2.1
	Content Changes on Input: When a user inputs information or interacts with a control, it MUST NOT result in a substantial change to the page, the spawning of a pop-up window, an additional change of keyboard focus, or any other change that could confuse or disorient the user unless the user is informed of the change ahead of time.	Required WCAG 3.2.2
Finding Added Content	**Finding Added Content:** Silence is bad. When screen reader users activate a feature (link, button, control, etc.), or when an important part of the content changes, they need to hear feedback. One of the basic challenges with dynamic content is to figure out the best way to tell screen reader users about the dynamic changes.	Required WCAG 1.3.2
Keyboard Focus Management During Interactions	**Set Keyboard Focus:** The focus MUST be purposely moved or set (via JavaScript) onto the appropriate element when the user's action requires a change of context or location for effective keyboard or touch interaction.	Required WCAG 2.4.3
	No Lost Focus: The focus MUST NOT become lost or reset to the top of the page, except when loading or re-loading a page.	Required WCAG 2.4.3
	Focus Target Has Text: When moving or setting focus, the destination element MUST contain discernible text (either standard text or programmatically associated text).	Required WCAG 1.3.1
Timers	**Session Timeout – Timing Adjustable:** If there is a session time limit, users MUST be warned before the session ends and MUST be given time to save their data and/or extend the session.	Required WCAG 2.2.1
User Input and Feedback	**Input:** See the requirements for ***Form Inputs, Labels, and Instructions***.	Required (Multiple)
	Feedback: See the requirements for ***Form Validation and Feedback***.	Required (Multiple)
User Input Methods	See the requirements for ***Device-Independent User Input*** regarding: • Mouse • Keyboard • Touch • Voice	Required (Multiple)

Custom Widgets (JavaScript, ARIA)

Topic	Technique	WCAG AA
Name	**Name of Interactive UI Elements:** Every interactive UI element (e.g. links, buttons, controls for custom widgets, form inputs/elements) MUST have a name, according to the accessible name computation. **Note:** The name can come from the native text of the element (e.g. link text, `<button>` text), a value attribute (e.g. `<input type="submit" value="Name goes here">`), the `aria-label` text, the text referred to via the `aria-labelledby` ID value, or other attributes, such as `title` (depending on context).	Required WCAG 4.1.2
	Name of Static Semantic Elements: The following semantic elements MUST have an accessible name, according to the accessible name computation: • Headings (or elements with `role="heading"`) • Images (or elements with `role="image"`) • Links (or elements with `role="link"`), via link text, `aria-label` (e.g. when the link contains only a background image), or `aria-labelledby`) • Frames, via the `title` attribute • Iframes, via the `title` attribute	Required WCAG 4.1.2
	Other Semantic Elements Benefitting from a Name: Examples of other semantic elements that SHOULD have an accessible name, according to the accessible name computation include: • Landmarks (HTML 5 or ARIA landmarks); Names are especially helpful when there are two of the same type of landmark on the same page, such as two navigation landmarks • Tables (via `<caption>`, `aria-label`, or `aria-labelledby`)	Best practice
Role	**Role Specified:** The semantic meaning of elements MUST be communicated via appropriate native HTML element or ARIA role.	Required WCAG 4.1.2
	HTML versus ARIA Role: When an HTML element exists, the HTML element SHOULD be used *instead of* the equivalent ARIA role, whenever possible.	Best practice
Value	**Static ARIA Properties:** Static ARIA properties (e.g. `aria-valuemax`), MUST be specified.	Required WCAG 4.1.2
	Static Non-ARIA Properties: Static non-ARIA properties MUST be specified in text (or alternative text). **Note:** The static property may be included in the native text of an element, in its label, in associated text (e.g. via `aria-describedby`), in adjacent text, or in some other way that is available to assistive technologies.	Required WCAG 4.1.2
	Initial State: The initial state of a changeable UI element MUST be programmatically designated (e.g. via ARIA attributes such as `aria-expanded="false"`, `aria-selected="true"`, `aria-sort="ascending"`, etc.)	Required WCAG 4.1.2

Custom Widgets (JavaScript, ARIA) *(continued)*

Topic	Technique	WCAG AA
Value *(continued)*	**ARIA State Changes:** When the visual and/or functional state of an element changes (e.g. `aria-valuenow`, `aria-pressed`, `aria-expanded`, `aria-checked`), the ARIA state MUST be change accordingly.	Required WCAG 4.1.2 (WCAG 2.0) WCAG 4.1.3 (WCAG 2.1)
	Non-ARIA State Changes: If a state change cannot be communicated via a change in an ARIA attribute, when the state change is the result of a user action or request, the state change MUST be communicated to the user visually AND MUST be communicated to assistive technologies using a technique such as: • `aria-live` announcement • ARIA alert (inject text into a pre-existing container with `role="alert"`) • Move keyboard focus to the announcement (warning: moving the focus can be disorienting, so should be used with caution)	Required WCAG 4.1.2 (WCAG 2.0) WCAG 4.1.3 (WCAG 2.1)
Keyboard Focus Management During Interactions	**Set Keyboard Focus:** The focus MUST be purposely moved or set (via JavaScript) onto the appropriate element when the user's action requires a change of context or location for effective keyboard or touch interaction.	Required WCAG 2.4.3
	No Lost Focus: The focus MUST NOT become lost or reset to the top of the page, except when loading or re-loading a page.	Required WCAG 2.4.3
	Focus Target Has Text: When moving or setting focus, the destination element MUST contain discernible text (either standard text or programmatically associated text).	Required WCAG 1.3.1
Keyboard Conventions	See **WAI-ARIA Authoring Practices for Developing a Keyboard Interface**.	Best practice
Instructions	**Instructions for Custom Widgets:** Widgets with non-standard interaction models SHOULD have instructions explaining how to use them.	Best practice
	See also the requirements for ***Form Inputs, Labels, and Instructions***.	Required (Multiple)
Custom Widgets	**Accordion** widgets SHOULD conform to WAI-ARIA Authoring Practices for Accordions.	Best practice
	Alert widgets SHOULD conform to WAI-ARIA Authoring Practices for Alerts.	Best practice
	Alert Dialog widgets SHOULD conform to WAI-ARIA Authoring Practices for Alert Dialogs.	Best practice
	Autocomplete widgets SHOULD conform to WAI-ARIA Authoring Practices for Autocomplete widgets.	Best practice
	Breadcrumb widgets SHOULD conform to WAI-ARIA Authoring Practices for Breadcrumbs.	Best practice

Custom Widgets (JavaScript, ARIA) *(continued)*

Topic	Technique	WCAG AA
Custom Widgets *(continued)*	**Button** widgets SHOULD conform to WAI-ARIA Authoring Practices for Buttons.	Best practice
	Button (Toggle) SHOULD conform to WAI-ARIA Authoring Practices for Buttons.	Best practice
	Carousel widgets (based on a Tab Panel) SHOULD conform to WAI-ARIA Authoring Practices for Carousels.	Best practice
	Checkbox widgets SHOULD conform to WAI-ARIA Authoring Practices for Checkboxes.	Best practice
	Checkbox (Tri-State) widgets SHOULD conform to WAI-ARIA Authoring Practices for Checkboxes.	Best practice
	Combobox widgets SHOULD conform to WAI-ARIA Authoring Practices for Comboboxes.	Best practice
	Dialog (Simple Modal) widgets SHOULD conform to WAI-ARIA Authoring Practices for Modal Dialogs.	Best practice
	Dialog (Simple Alert Modal) widgets SHOULD conform to WAI-ARIA Authoring Practices for Modal Dialogs.	Best practice
	Dialog (Message Modal) widgets SHOULD conform to WAI-ARIA Authoring Practices for Modal Dialogs.	Best practice
	Dialog (Message Alert Modal) widgets SHOULD conform to WAI-ARIA Authoring Practices for Modal Dialogs.	Best practice
	Disclosure (Show/Hide or Expand/Collapse) widgets SHOULD conform to WAI-ARIA Authoring Practices for Disclosures.	Best practice
	Disclosure (Based on Details/Summary) widgets SHOULD conform to WAI-ARIA Authoring Practices for Disclosures.	Best practice
	Feed widgets SHOULD conform to WAI-ARIA Authoring Practices for Feeds.	Best practice
	Grid widgets (Interactive Tabular Data and Layout Containers) SHOULD conform to WAI-ARIA Authoring Practices for Grids.	Best practice
	Link widgets SHOULD conform to WAI-ARIA Authoring Practices for Links.	Best practice
	Listbox widgets SHOULD conform to WAI-ARIA Authoring Practices for List Boxes.	Best practice
	Menu widgets SHOULD conform to WAI-ARIA Authoring Practices for Menus.	Best practice
	Menubar widgets SHOULD conform to WAI-ARIA Authoring Practices for Menubars.	Best practice
	Menu Button widgets SHOULD conform to WAI-ARIA Authoring Practices for Menu Buttons.	Best practice
	Navigation (Hierarchical) widgets with Expand/Collapse widgets SHOULD conform to WAI-ARIA Authoring Practices for Disclosures.	Best practice
	Progress Bar (Bounded) widgets SHOULD conform to WAI-ARIA Authoring Practices for Progress Bars.	Best practice

Custom Widgets (JavaScript, ARIA) *(continued)*

Topic	Technique	WCAG AA
Custom Widgets *(continued)*	**Progress Bar (Unbounded)** widgets SHOULD conform to WAI-ARIA Authoring Practices for Progress Bars.	Best practice
	Radio and Radio Group widgets SHOULD conform to WAI-ARIA Authoring Practices for Radio Groups.	Best practice
	Slider widgets SHOULD conform to WAI-ARIA Authoring Practices for Sliders.	Best practice
	Slider (Multi-Thumb) widgets SHOULD conform to WAI-ARIA Authoring Practices for Multi-Thumb Sliders.	Best practice
	Spinbutton widgets SHOULD conform to WAI-ARIA Authoring Practices for Spinbuttons.	Best practice
	Tab Panel widgets SHOULD conform to WAI-ARIA Authoring Practices for Tab Panels.	Best practice
	Table widgets SHOULD conform to WAI-ARIA Authoring Practices for Tables.	Best practice
	Table (Responsive, Collapsible) widgets SHOULD maintain data relationships through **table structure** (see the ***Tables*** section), **list hierarchy** (see the ***Lists*** section), and/or **headings** (see the ***Headings*** section).	Best practice
	Table (Sortable) widgets SHOULD be constructed of a standard HTML table, if possible, and make full use of ARIA sort attributes. (WAI-ARIA Authoring Practices for Tables.)	Best practice
	Toolbar widgets SHOULD conform to WAI-ARIA Authoring Practices for Toolbars.	Best practice
	Tooltip widgets SHOULD conform to WAI-ARIA Authoring Practices for Tooltips.	Best practice
	Tooltip Dialog widgets SHOULD conform to WAI-ARIA Authoring Practices for Dialogs.	Best practice
	Tree View widgets SHOULD conform to WAI-ARIA Authoring Practices for Tree Views.	Best practice
	Window Splitter widgets SHOULD conform to WAI-ARIA Authoring Practices for Window Splitters.	Best practice

CAPTCHA

Avoid CAPTCHAs if Possible

CAPTCHAs require human problem-solving skills — which can be difficult or impossible for users with cognitive disabilities — and often require sensory abilities — which can be difficult or impossible for users who are blind, deafblind, or deaf. It is best to avoid CAPTCHAs altogether, and instead implement smart algorithms that do not require human input.

Topic	Technique	WCAG AA
Text Alternatives	**Text Alternative Describing The Purpose:** If the CAPTCHA is not text-based (e.g. image or audio), a text alternative MUST communicate the purpose of the CAPTCHA, so that the user knows that this task must be completed before proceeding to the next step. • **Note 1:** Ideally the alternative text would allow non-visual users to complete the task, but at a minimum it should inform users of the purpose of the CAPTCHA. • **Note 2:** If there is an alternative CAPTCHA (e.g. in text elsewhere on the page, or in audio), the user SHOULD be notified that the alternative exists, by mentioning it either in the alternative text for the original CAPTCHA, or in the surrounding text.	Required WCAG 1.1.1
	Text-Based CAPTCHA: A method SHOULD be available in a text-based format (either as the main CAPTCHA or as an alternative) that can be converted by a screen reader to braille output. • **Note:** Although WCAG does not require a text-based CAPTCHA, deafblind users *require* a text-based method, because neither visual nor audio methods will be sufficient.	Best practice
Sensory Alternatives	**Sensory Alternative:** If a non-visual user cannot pass the original CAPTCHA, an alternative method MUST be provided in another sensory modality (e.g. audio).	Required WCAG 1.1.1
Keyboard Accessibility	User input controls in a CAPTCHA (or in an alternative representation) must meet all the **keyboard functionality** requirements (see the ***Device-Independent User Input*** section).	Required (Multiple)
Dynamic Content	Any dynamic content in a CAPTCHA (or in an alternative method) must meet all the ***Dynamic Content (JavaScript, AJAX)*** requirements.	Required (Multiple)
Custom Widgets	Any custom widgets in a CAPTCHA (or in an alternative method) must meet all the ***Custom Widgets (JavaScript, ARIA)*** requirements.	Required (Multiple)
Color and Contrast	Any visual elements in a CAPTCHA (or in an alternative method) must meet all the ***Color and Contrast*** requirements.	Required (Multiple)