Web Accessibility Checklist

Based on WCAG 2.0 AA Requirements (marked with "must") and best practices (marked with "should")

Part 1: Semantic Structure

Topic	Accessibility Requirements	WCAG
Page title	**The page MUST have a meaningful title** (e.g. <title>About us</title>), even when included via iframe. • Unique information SHOULD go first (e.g. "Products | Deque University"). • Result pages SHOULD describe the result (e.g. "Error on form" or "Search results loaded"). • Single-page applications and AJAX scripts SHOULD update the title when the URL changes or when the page content changes significantly.	2.4.2
Language	**The page MUST specify the language** (<html lang="en">).	3.1.1
	Changes in the language within the page MUST be specified (e.g. Hola).	3.1.2
Landmarks	**Pages SHOULD have accurate, logical landmark structure** (e.g. <header>, <nav>, <main>, <footer>), so that screen reader users can navigate by landmark, and all content SHOULD be inside a landmark.	(2.4.6)
Headings	**The page MUST have meaningful headings to label each major section**, which SHOULD start with <h1> (at the beginning of the main content, or at the beginning of every section of aggregated content, or at the beginning of modal dialogs), and SHOULD NOT skip heading levels, to allow screen reader users to navigate the tree structure of the heading hierarchy.	2.4.6
Links and Navigation [See also Custom Widgets in Part 3 for dynamic menus (drop-down accordion, etc.)	**Links MUST have readable text** Be especially careful with links that contain only images (which need alt text) and background images/icon fonts (which need text via aria-label on the link or text within the link, hidden via CSS).	2.4.9, 4.1.2
	The link text MUST make sense in context, and should make sense when taken out of context (problematic phrases include: "click here," "more," "read more," etc.).	2.4.4, 2.4.9
	Linked content SHOULD be grouped in a single link where appropriate. For example: an icon and its adjacent text SHOULD NOT be two separate links if they go to the same location.	3.2.4
	Navigation features (e.g. main menu) MUST be placed in a consistent location across pages.	3.2.3
	Navigation features MUST be identified in a consistent way across pages.	3.2.4
	A "skip navigation" or "skip to main content" SHOULD be provided as the first link in the design, to allow sighted keyboard users to quickly arrive at the main content (Note: the link can be invisible until the user tabs to it, but it MUST NOT remain invisible when it receives keyboard focus).	2.4.5
Tables	**Header cells (<th>) MUST be associated with their respective data cells** (via scope or headers + id).	1.3.1
	Tables SHOULD have an accessible name (e.g. <caption>, aria-label, or aria-labelledby).	(4.1.2)
	Layout tables (no header/data associations) MUST NOT contain <th> or other header markup.	1.3.1
Lists	**Lists MUST be marked up appropriately** according to the semantics of the list (e.g. , , <dl>).	1.3.1
Iframes	**Frame title attribute MUST be specified** (<iframe title="Video about...").	4.1.2
	The page within the iframe MUST have an accurate, meaningful <title>.	2.4.2
	Iframes with no readable content (e.g. only JavaScript) SHOULD be set to aria-hidden="true"	(4.1.2)
Form Markup (See also Form Validation and Feedback in Part 3)	**Inputs, buttons, and controls MUST have labels which are programmatically-associated** (e.g. via <label>, aria-label, or aria-labelledby) **and always visible on the screen** (they don't disappear when the user starts typing).	1.3.1, 3.3.2
	Required fields SHOULD be marked as such, e.g. via aria-required="true" (on the input, not on the label), or have the word "required" in the <label> text.	3.3.2
	Form field instructions SHOULD be associated with inputs or buttons using techniques such as: • Putting the instructions in the <label> • Associating the instructions with the field using aria-describedby (Note: text associated via aria-describedby SHOULD be relatively brief) • Putting the instructions in a <fieldset> with <legend>	3.3.2
	Groups of form elements MUST have group labels (e.g. <fieldset> and <legend>, or referenced from the inputs via aria-labelledby="id-of-group-label id-of-self-label").	1.3.1
Parsing and Validity	**The page MUST NOT contain duplicate IDs** because accessibility features frequently reference IDs.	4.1.1
	Attributes (e.g. ARIA) MUST contain only allowable values, in the proper parent-child hierarchy.	4.1.1
	Parent-child relationships of elements & attributes (e.g. ARIA roles) MUST follow the specification.	4.1.1
	The page MUST NOT contain syntax errors that affect semantic meaning (e.g. elements or attributes that don't close properly [either explicitly or implicitly])	4.1.1

Part 2: Sight and Sound

Topic	Accessibility Requirements	WCAG
Color and Other Sensory Characteristics	**Sensory characteristics (e.g. color, size, shape, visual placement, visual orientation, sound, etc.) MUST NOT be the only way to convey information.**	1.1.1 1.3.3
	Color MUST NOT be used as the only way to distinguish links from regular text.	1.4.1
Contrast	**Contrast of text against the background MUST meet the minimum threshold** (use automated tools to determine if the contrast passes or fails).	1.4.3
Images	**Informative images MUST have brief, meaningful alt text describing the *purpose* of the image.**[1]	1.1.1
	Actionable images (images used as links, buttons or controls) MUST have brief, meaningful alt text describing the destination or result of the action.	1.1.1
	CAPTCHAs MUST have alternatives that do not rely on sensory experiences (i.e. vision, sound).	1.1.1
	Complex images (images requiring more than about 200 characters to describe them) MUST be given extended text descriptions. Provide the text below (or above) the image, or on a separate page (via a link), or in a dialog (via a button), or similar technique.	1.1.1
	Non-informative decorative or redundant Images MUST be coded to be ignored by screen readers by using alt="" on the <img> or Putting the image in the background with CSS.	1.1.1
	Images MUST NOT be used to convey text, except in logos and other essential situations (use real text instead, so users can magnify it, change contrast, change colors, etc.).	1.4.5
Videos and Audio	**Videos (prerecorded or live) MUST have synchronized captions** for the deaf.	1.2.2, 1.2.4
	Videos MUST have narrated audio descriptions for the blind, if the audio track (dialog, narration, sounds) of the video does not convey all important visual information.[2]	1.2.3, 1.2.5
	Videos and audio (prerecorded) MUST have a text transcript for the deaf and deafblind, on the page, available via a link, available via a dialog (activated by a button), or similar technique.	1.2.1
	Videos and audio SHOULD not auto-play to avoid interfering with screen reader speech.	(1.4.2)
	Volume level MUST be adjustable via a control in the media player.	1.4.2
Zoom and Reflow	**Users MUST be able to use pinch-to-zoom** on touch devices.	1.4.4
	A responsive design (with text reflow relative to the viewport width) SHOULD be available on desktop as well as mobile designs for the benefit of people with low vision who zoom in.	(1.4.4)
Visual Proximity	**Related information (labels, descriptions, feedback, controls, etc.) SHOULD be close in visual proximity** so that screen magnifier users can see the related parts on the screen at the same time.	LV[3]
Flashing Effects	**There MUST NOT be any flashing or blinking effects** faster than 3 times in any one-second period, to avoid causing a seizure in people susceptible to them.	2.3.1
Typographic Design	**Fonts SHOULD be easy to read** (avoid cursive fonts, overly decorative fonts, etc.).	n/a
	Text SHOULD NOT be full-justified, to improve readability.	(1.4.8)
	Within paragraphs, line spacing SHOULD be at least 1.5 whenever possible.	(1.4.8*)
	Between paragraphs, spacing SHOULD be at least 1.5 times larger than the line spacing.	(1.4.8*)
	The design SHOULD use standard font face wherever possible (eliminate or minimize phrases written in all capitals, long sections of italic or bold text, etc.).	n/a
	If emphasis is critical to understand the text, it MUST be written in words (e.g. "important" or "warning") in the text or in alternative text. Text variants (bold, italic, all capitals) aren't enough.	(1.1.1)
Screen Orientation	**Designs SHOULD support both vertical (portrait) and horizontal (landscape) orientation**, so that the design displays correctly for users who cannot switch orientation (e.g. if they have the device attached to a wheelchair).	n/a
Hidden Content	**Content that is hidden from sighted users SHOULD also be hidden from screen reader users** (e.g. using CSS display:none), except where the hidden content is intended only for screen reader users.	(1.3.2)
CSS Generated Content (e.g. font icons, etc.)	**Designs SHOULD NOT rely on CSS-generated content to convey information,** due to incomplete support across screen reader/browser combinations (especially IE). Supplement with CSS clipped content in the HTML[4], aria-label/aria-labelledby (on focusable elements), or similar technique.	1.1.1

Part 3: Interactivity and Dynamic Content

Topic	Accessibility Requirements	WCAG
Keyboard Input	**All features MUST be fully functional** when using only the keyboard (drop-down menus, etc.)	2.1.1
	Keyboard tab order MUST be logical, and SHOULD match the order of the visual design (Note: Achieve this via proper order in the DOM. Avoid tabindex of positive values).	2.4.3
	Keyboard focus MUST follow the action (e.g. the focus goes to a dialog when the dialog is activated, and when the dialog is dismissed, the focus returns to the button that activated the dialog).	2.4.3
	Visual focus indicator MUST always be visible when tabbing through the page.	2.4.7
	The page MUST NOT have a keyboard trap. Users MUST be able to navigate to and past all links, buttons, inputs, and controls, both forward (using the tab key) and backward (using shift + tab)	2.1.2
	Keyboard shortcuts MUST NOT interfere with shortcuts in the browser, screen reader, or OS.	2.1.1
Touch Input	**All features SHOULD be fully functional** when using only touch.	M
	All features SHOULD be fully functional by touch with the screen reader turned on. Click actions are required, because JavaScript swipe events, and similar, are disabled when the screen reader is on.	M
	Whenever possible, the clickable target SHOULD be at least 9mm high and 9mm wide, so that users can activate them with their finger, without having to zoom in.	M
Form Validation and Feedback (See also Form Markup in Part 1)	**Screen reader users MUST receive either a confirmation message or error message** immediately after the form is submitted (silence is bad) through methods such as: • Focus is sent to the confirmation/error message OR • Focus is sent to the first field with an error (and the error is associated with the field) OR • The page <title> contains the confirmation/error message (if the user is sent to a new page or if the page reloads when the form is submitted).	3.3.1, 3.3.3
	Error messages about an input (as opposed to the form as a whole) MUST be associated with that input (e.g. via <label> or aria-describedby)	3.3.3
	Error messages MUST describe the error in enough detail to allow users to fix the error.	3.3.3
	Forms with legal, financial, or data functions MUST protect users from errors by one of the following: 1) allow actions to be reversible, 2) automatically check and correct errors, or 3) allow users to review/confirm/correct submissions.	3.3.4
Custom Widgets	**Labels MUST be specified** on all controls and buttons (e.g. via <label>, aria-label, or aria-labelledby)	3.3.2
	Roles MUST be specified, where appropriate via HTML or ARIA (e.g. <button> or role="button", role="slider", role="dialog", role="tablist", etc.). See https://www.w3.org/TR/wai-aria/roles	4.1.2
	Property states and values (and changes to states and values) MUST be specified and updated on controls where appropriate (e.g. aria-expanded="false", aria-selected="true", aria-valuenow="100").	4.1.2
	Relationships MUST be specified, where appropriate (aria-controls, aria-owns, aria-activedescendant).	4.1.2
	Keyboard behaviors SHOULD follow ARIA conventions specific to the type of widget.	ARIA[5]
Manage Focus	**Scripts MUST manage keyboard focus when necessary (but only when necessary)** (e.g. clicking on a button sends the focus to a dialog; closing the dialog sends the focus back to the original button).	2.4.3
Timing	**Users MUST be allowed to turn off, adjust, or extend time limits** (e.g. timed scripts, session timeout, page reload) unless the fundamental nature of the activity requires otherwise (e.g. real-time events).	2.2.1
Motion and Animation	**Users MUST be able to pause, stop, hide, or control the timing** of content that blinks, moves, auto-scrolls, or auto-updates.	2.2.2
Dynamic Updates (including in a single-page application framework)	**Screen reader users MUST be made aware of important updates or changes to the content** (e.g. content loaded by AJAX or changed by JavaScript, etc.) by methods such as the following: • aria-live announcement (appropriate when the keyboard focus should not move) OR • move the keyboard focus to the new content (only applicable if the user performs an action to request the content change, e.g. by clicking a button or link).	1.3.2
	Updates/additions SHOULD be loaded near (preferably below) the user's current position.	(1.3.2)
	"On focus" MUST NOT trigger a change of context (e.g. user agent, focus, viewport, or content).	3.2.1
	"On input" MUST NOT trigger a change of context unless the user is advised in advance.	3.2.2
Objects and Plugins	**All <object> elements MUST have alternative text.**	1.1.1
	An object or plugin SHOULD conform to the best practices of the accessibility API of that object or plugin, if available (e.g. the PDF format has an accessibility API).	n/a
	If an object or plugin is not compatible with assistive technologies, an alternative representation MUST be available in an accessible format.	1.1.1

Part 4: Understandability

Topic	Accessibility Requirements	WCAG
Content	**The text SHOULD be written in the simplest and clearest language possible**, appropriate to the content.	COG[6]
	Uncommon words and phrases – including technical terms, idioms, jargon, foreign phrases, etc. – SHOULD be defined or explained in the text.	(3.1.3*)
	Abbreviations and acronyms SHOULD be expanded or explained in the text.	(3.1.4*)
	The page SHOULD provide supplemental versions of the content (e.g. illustrations, videos, audio, simplified text version, etc.) to enhance comprehension, if the reading level is more advanced than a lower secondary education level.	(3.1.5*)
User Experience	**The information architecture of the site and design SHOULD be intuitive**, allowing the user to easily find, navigate, read, and interact with the content.	COG
	The design SHOULD eliminate or minimize visual distractions.	COG
	The site SHOULD minimize the number of steps required for users to complete actions.	COG
	The site SHOULD provide ways for users to find or request help (e.g. context-sensitive help, tutorials, online chat with a customer service representative, discussion forum, a form to submit a question, contact information to call or write for assistance, etc.)	(3.3.5*)
	Functionality SHOULD be easily discoverable (e.g. a menu that can be revealed only by a finger gesture swiping across the screen from the left to the right would not be easily discoverable).	COG
	The design SHOULD follow common conventions for design and user interaction, unless there is a compelling (e.g. research-validated) reason to break from convention.	COG
	Users SHOULD NOT be required to react quickly to information or user interface features.	2.1.1
	Users SHOULD NOT be interrupted by scripted events, or SHOULD be able to suppress or postpone interruptions, except in the case of emergencies.	(2.2.4*)
	Users SHOULD NOT be required to perform mathematical calculations, unless the fundamental purpose of the page requires them to do so.	COG
	Users SHOULD NOT be required to remember data from one page (or view, or step of a process) to another, unless the fundamental purpose of the content requires them to do so.	COG

Notes:

* Items marked with asterisk are required at the WCAG AAA level (not at the A or AA level, but are still best practices at all levels).

---

[1] Note that the purpose of the image depends on the context, and may not be a literal description of what is in the image. The text is usually provided via the alt attribute, but may also be provided via aria-label, aria-labelledby or, in less straightforward scenarios, via hidden text clipped by CSS.

[2] The audio descriptions can be on a separate version of the video, with a link provided to that version.

[3] Refers to information published by the W3C task force on low vision https://www.w3.org/TR/2016/WD-low-vision-needs-20160317/

[4] Example in the HTML:
Our Facebook page.
The CSS clip method:
.visually-hidden {position:absolute; clip:rect(0 0 0 0); border:0; height:1px; margin:-1px; overflow:hidden; padding:0; width:1px;}

[5] Refers to the WAI-ARIA documentation (https://www.w3.org/TR/wai-aria-practices)

[6] Refers to the information published by the W3C task force on cognitive disabilities https://www.w3.org/TR/coga-user-research/