

# Web Accessibility Checklist

By Web Dev Simplified <https://courses.webdevsimplified.com>

This checklist is based on the [WCAG 2.2 guidelines](#)

**A** Essential **AA** Standard **AAA** Enhanced

## HTML

- ☐ Set page language **A**  
Add lang attribute to <html> (e.g., lang="en")
- ☐ Set unique page titles **A**  
Each page should have its own <title>
- ☐ Use semantic HTML **A**  
<header>, <nav>, <main>, <footer>, <aside>
- ☐ Ensure linear content flow **A**  
DOM order should match visual reading order
- ☐ Ensure zoom functionality **AA**  
<meta name="viewport" content="width=device-width, initial-scale=1.0" />

## Keyboard

- ☐ Keyboard accessible elements **A**  
Interactive elements work without the mouse
- ☐ Remove keyboard traps **A**  
Users can always tab away from elements
- ☐ Don't use positive tabindex **A**  
-1 and 0 can be used for specific cases
- ☐ Properly handle modal focus **A**  
Users should not be able to interact with background elements while a modal is open
- ☐ Don't remove focus styles **AA**  
Add a custom focus style if default is removed

## Appearance & Animation

- ☐ Don't convey info by color alone **A**  
Use patterns, icons, or text alongside color
- ☐ Use subtle animations **A**  
Keep animations subtle, no rapid flashing
- ☐ Support responsive reflow **AA**  
Content does not horizontal scroll at 320px
- ☐ Support custom fonts **AA**  
Especially useful with dyslexia-friendly fonts
- ☐ Respect prefers-x settings **AAA**  
This includes all settings such as:
  - prefers-reduced-motion
  - prefers-color-scheme
  - prefers-contrast
  - prefers-reduced-transparency

## Forms

- ☐ Group related form fields **A**  
Group related inputs: <fieldset>, <legend>
- ☐ Display errors in accessible list **A**  
Clearly identify errors in text, not just color
- ☐ Associate errors with inputs **A**  
Use aria-describedby to link error to input
- ☐ Associate labels with all inputs **A**  
Every input needs a visible <label> with for/id
- ☐ Add autocomplete attribute to all relevant inputs **AA**
- ☐ Ensure controls have focus states **AA**  
Interactive elements need visible focus styles

## Content

- ☐ Don't rely on sensory characteristics A  
Avoid "click the red button" or "see left sidebar"
- ☐ Make labels unique and descriptive A
- ☐ Use semantic table elements for tabular data A  
<table>, <thead>, <tbody>, <tr>, <td>, etc.
- ☐ Use <th> with scope for headers A
- ☐ Use <caption> to describe tables A
- ☐ Mark language changes in content AA  
Add lang attribute to foreign phrases (lang="fr")
- ☐ Define unusual terms AAA  
Provide definitions for jargon and idioms
- ☐ Explain abbreviations AAA  
Expand abbreviations on first use (e.g., HTML)

## Links & Buttons

- ☐ Make links visually recognizable A  
Don't rely solely on color; use underlines too
- ☐ Write self-explanatory link text A  
Avoid "click here" or "read more"
- ☐ Use correct element for purpose A  
Use <a> for navigation, <button> for actions
- ☐ Label buttons that have no text A  
Add aria-label or visually hidden text
- ☐ Identify links that open new window AA  
Add (opens new tab) or icon with aria-label
- ☐ Ensure adequate target size (24px) AA  
Touch targets should be at least 24x24px
- ☐ Use enhanced target size (44px) AAA  
Increase touch targets to 44x44px

## Color Contrast

- ☐ Check normal text contrast AA  
4.5:1 ratio for text under 24px or 19px bold
- ☐ Check large text contrast AA  
3:1 ratio for text over 24px or 19px bold
- ☐ Check text contrast over images AA  
Verify text over images is easily readable
- ☐ Verify contrast on all color themes AA  
Test both light and dark mode
- ☐ Check icon contrast AA  
3:1 ratio for icons and graphics
- ☐ Check UI component contrast AA  
Input borders, buttons, and focus rings at 3:1
- ☐ Meet enhanced contrast (7:1) AAA  
7:1 for normal text, 4.5:1 for large text

## Headings & Structure

- ☐ Use headings to structure content A  
Don't use headings just for a larger font size
- ☐ Use only one <h1> per page A
- ☐ Don't skip heading levels A  
Don't skip from <h2> to <h4> for example
- ☐ Use proper list markup for lists A  
Use <ul>, <ol>, or <dl>
- ☐ Provide skip navigation link A  
This should be the first link on the page
- ☐ Offer multiple ways to find content AA  
Provide search, sitemap, navigation menu, etc.
- ☐ Write descriptive headings AA  
Headings should be descriptive enough even if read alone

## Images

- ☐ Add alt text to all images A  
Describe meaning, not just appearance
- ☐ Use alt="" for decorative images A
- ☐ Add descriptions for complex images A  
Provide descriptions for charts, graphs, etc. near the chart or a link to a description
- ☐ Include image text in alt A  
If an image contains text, include it in the alt
- ☐ Avoid images of text AA

## Media (Audio & Video)

- ☐ Prevent media from autoplaying A
- ☐ Make all media pausable A
- ☐ Ensure media controls are accessible A  
Custom players need proper ARIA roles
- ☐ Provide video captions A
- ☐ Provide audio transcripts A
- ☐ Remove seizure triggers A  
Nothing flashes more than 3 times per second

## ARIA & Custom Widgets

- ☐ Avoid focus-triggered changes A  
Changing focus shouldn't trigger unexpected changes
- ☐ Avoid input-triggered changes A  
Modifying input shouldn't auto-submit form
- ☐ Prefer native HTML over ARIA A
- ☐ Use ARIA roles for custom elements A  
role="button", role="tab", etc.
- ☐ Update ARIA states dynamically A  
aria-expanded, aria-selected, aria-checked
- ☐ Provide accessible names for elements with no visible label A  
Use aria-label or aria-labelledby
- ☐ Hide decorative content from screen readers A  
Use aria-hidden for decorative content
- ☐ Announce dynamic content changes AA  
Use aria-live="polite" for non-urgent updates  
Use aria-live="alert" or aria-live="assertive" for urgent updates

## Testing & Validation

- ☐ Run accessibility audits  
Lighthouse is a great free option
- ☐ Add linting to build process  
eslint-plugin-jsx-a11y in CI/CD
- ☐ Validate color contrast  
Lighthouse does this automatically
- ☐ Test keyboard-only navigation  
Don't use the mouse at all
- ☐ Test with screen readers  
VoiceOver, NVDA, TalkBack, JAWS, etc.
- ☐ Test at 200% zoom
- ☐ Test with increased font size  
This can be set in your operating system
- ☐ Test color blindness modes  
Browser devtools have this built-in
- ☐ Test blurry vision simulation  
Browser devtools have this built-in
- ☐ Verify focus management  
All elements receive focus in logical order
- ☐ Test on mobile devices  
Touch-only testing on real devices