Website accessibility checklist

The checks that catch most real problems, each with its WCAG 2.2 success criterion. It covers the common Level A and AA requirements, but it isn't a full audit; automated tools find only part of the issues, so test by hand too.

Start here: the 6 steps that matter most

If you only have an hour or two, do these first. The full list, with how to check each one, follows.

1. Give meaningful images a text alternative (1.1.1)
   Every meaningful image has alt text saying what it shows or does; decorative images have alt="".
2. Make text contrast at least 4.5 to 1 (1.4.3)
   All text meets these ratios, including text on images and buttons.
3. Make everything work with a keyboard (2.1.1, 2.1.2)
   You can reach and use everything on the page with the keyboard alone, and leave every part again.
4. Show where keyboard focus is (2.4.7)
   A clearly visible focus outline on every link, button and field.
5. Label every field (1.3.1, 3.3.2, 4.1.2)
   Each field has a visible label connected to it (for and id), and buttons have a name.
6. Set the page language (3.1.1)
   <html lang="en"> (or your language) on every page.

How to use this checklist

Every step has a priority (**Start here**, **Important**, **When you have time**), who usually does it (**You** or a **developer**) and a rough time. **Done when** says what finished looks like, **How to check** tells you how to confirm it yourself, and **Common mistake** is the slip we see most. Tick the box when a step passes its check.

Steps marked with a free tool can be checked in seconds at rrwebservices.com/tools/. Sources are numbered at the end; every fact was checked against them on November 9, 2026.

Your score: _____ of 24 steps done Date: _____

Images and media

- [ ] **1. Give meaningful images a text alternative (1.1.1)**

  Start here · You · 30 min

  Screen readers read alt text aloud. Images of text, icons that act as buttons and charts all need one.

  Done when: Every meaningful image has alt text saying what it shows or does; decorative images have alt="".

  How to check: The Accessibility Checker lists images without alt text.

  Common mistake: alt="image" or the file name.

  Check it free with the Accessibility Checker Source 1

- [ ] **2. Caption videos and provide transcripts (1.2.2, 1.2.1)**

  Important · You · Varies

  People who can't hear need captions for videos with sound, and a transcript for audio-only.

  Done when: Videos with speech have accurate captions; podcasts have transcripts.

  How to check: Turn the sound off and watch.

  Common mistake: Relying on uncorrected automatic captions.

  Source 1

- [ ] **3. Let people pause anything that moves (2.2.2)**

  Important · Developer · 30 min

  Carousels and animations that move for more than five seconds need a way to pause, stop or hide them.

  Done when: Auto-moving content has a visible pause button.

  How to check: Look for sliders and animated banners.

  Common mistake: Auto-rotating hero sliders with no pause.

  Source 1

Seeing the page

- [ ] **4. Make text contrast at least 4.5 to 1 (1.4.3)**

  Start here · Developer · 30 min

  Low contrast is one of the most common failures. Large text (about 24px, or 19px bold) needs at least 3 to 1.

  Done when: All text meets these ratios, including text on images and buttons.

  How to check: The Accessibility Checker measures contrast.

  Common mistake: Light gray text on white, or white text on a pale brand color.

  Check it free with the Accessibility Checker Source 1

- [ ] **5. Make icons, borders and focus rings visible (1.4.11)**

  Important · Developer · 20 min

  Parts needed to use the page, like input borders and icons, need at least 3 to 1 contrast with what's next to them.

  Done when: Form field borders, icons and focus indicators meet 3 to 1.

  How to check: Look at empty form fields: can you see where to type?

  Common mistake: Inputs with a very faint border.

  Source 1

- [ ] **6. Don't use color alone to show meaning (1.4.1)**

  Important · Developer · 20 min

  People who can't tell colors apart miss red-for-error or green-for-OK.

  Done when: Errors, required fields and links in text also use text, icons or underlines.

  How to check: View the page in grayscale.

  Common mistake: Links in body text shown only by color.

  Source 1

- [ ] **7. Let text resize to 200% and reflow at 320px (1.4.4, 1.4.10)**

  Important · Developer · 30 min

  People zoom in. Content must still work without losing text or scrolling sideways (except for things like maps and tables).

  Done when: At 200% zoom, and at 320 CSS pixels wide, nothing is cut off or overlapping.

  How to check: Zoom your browser to 200% and to 400% on a desktop.
  Common mistake: Fixed-height boxes that cut off enlarged text.
  Source 1

Using it with a keyboard

- [ ] **8. Make everything work with a keyboard (2.1.1, 2.1.2)**

  Start here · Developer · 30 min

  Some people can't use a mouse. Every link, button, menu and form must work with Tab, Enter, Space and arrow keys, and focus must never get trapped.

  Done when: You can reach and use everything on the page with the keyboard alone, and leave every part again.

  How to check: Put the mouse away and Tab through the whole page.
  Common mistake: Dropdown menus that open only on hover; pop-ups you can't close with Escape.
  Source 1

- [ ] **9. Show where keyboard focus is (2.4.7)**

  Start here · Developer · 15 min

  Keyboard users need to see which element is selected.

  Done when: A clearly visible focus outline on every link, button and field.

  How to check: Tab through the page and watch the focus.
  Common mistake: outline: none in the CSS with nothing in its place.
  Source 1

- [ ] **10. Don't hide the focused element behind sticky bars (2.4.11, new in 2.2)**

  Important · Developer · 20 min

  Sticky headers, cookie banners and chat buttons can cover the item that has focus.

  Done when: The focused item is never entirely hidden by other content.

  How to check: Tab through the page with the cookie banner open and the sticky header showing.
  Common mistake: A sticky footer bar covering focused links at the bottom of the screen.
  Source 2

- [ ] **11. Add a "skip to content" link (2.4.1)**

  Important · Developer · 15 min

  It lets keyboard and screen reader users skip the menu on every page.

  Done when: The first Tab on each page shows a working "Skip to content" link.

  How to check: Load a page and press Tab once.
  Common mistake: A skip link that points to an id that doesn't exist.
  Source 1

- [ ] **12. Make tap and click targets at least 24 by 24 pixels (2.5.8, new in 2.2)**

  Important · Developer · 20 min

  Small targets are hard to hit. WCAG 2.2 asks for at least 24 by 24 CSS pixels, or enough space around smaller targets.

  Done when: Buttons, icons and links in lists meet the size or spacing.

  How to check: Use your browser's inspector to measure small icons.
  Common mistake: Tiny social icons packed together.
  Source 2

- [ ] **13. Offer an alternative to dragging (2.5.7, new in 2.2)**

When you have time · Developer · 30 min

Anything done by dragging, like sliders or reordering, needs a single-click way too.

Done when: Drag interactions have buttons or inputs that do the same thing.

How to check: Find every slider and drag-and-drop on the site.
Common mistake: Price range sliders with no input boxes.
Source 2

Forms

- [ ] **14. Label every field (1.3.1, 3.3.2, 4.1.2)**

Start here · Developer · 20 min

Screen readers announce the label. Placeholder text disappears when typing and isn't a label.

Done when: Each field has a visible label connected to it (for and id), and buttons have a name.

How to check: The Accessibility Checker lists unlabeled fields and nameless buttons.
Common mistake: Search boxes and newsletter fields with only a placeholder.
Check it free with the Accessibility Checker Source 1

- [ ] **15. Explain errors in text, next to the field (3.3.1, 3.3.3)**

Important · Developer · 30 min

People need to know what went wrong and how to fix it.

Done when: Errors are written out ("Enter an email like name@example.com") and linked to the field.

How to check: Submit the form empty and with wrong data.
Common mistake: A red border only.
Source 1

- [ ] **16. Mark the purpose of personal fields (1.3.5)**

When you have time · Developer · 10 min

autocomplete attributes let browsers fill in name, email and address, which helps many users.

Done when: Fields like name, email, phone and address have the right autocomplete value.

How to check: View source on your contact form.
Common mistake: autocomplete="off" on contact forms.
Source 1

- [ ] **17. Don't make people type the same thing twice (3.3.7, new in 2.2)**

When you have time · Developer · 30 min

Information already entered in a process should be filled in or selectable.

Done when: Multi-step forms carry over earlier answers (like a shipping address to billing).

How to check: Go through your checkout or booking flow.
Common mistake: Asking for the email again on step three.
Source 2

- [ ] **18. Don't require puzzles to sign in (3.3.8, new in 2.2)**

When you have time · Developer · 15 min

Log-ins shouldn't depend on remembering or transcribing, unless there's another way, such as allowing paste and password managers.

Done when: Password fields allow paste; there's an alternative to any memory or puzzle test.

How to check: Try pasting into your login fields.
Common mistake: Blocking paste in password fields.
Source 2

Understanding and finding your way

- [ ] **19. Give every page a descriptive title (2.4.2)**

Important · You · 20 min

It's the first thing a screen reader announces and what shows in browser tabs.

Done when: Each page has a unique title describing it.

How to check: Look at the browser tab for a few pages.

Common mistake: Every page titled with the site name only.

Check it free with the SEO Health Dashboard · Source 1

- [ ] **20. Set the page language (3.1.1)**

Start here · Developer · 5 min

Screen readers use it to pronounce words correctly.

Done when: <html lang="en"> (or your language) on every page.

How to check: The Accessibility Checker checks it.

Common mistake: A template with lang missing.

Check it free with the Accessibility Checker · Source 1

- [ ] **21. Use real headings, in order (1.3.1, 2.4.6)**

Important · Developer · 20 min

Screen reader users jump between headings to find their way.

Done when: Headings are h1 to h6 elements in order, describing the section below.

How to check: The Accessibility Checker shows the heading outline.

Common mistake: Bold text styled to look like a heading.

Check it free with the Accessibility Checker · Source 1

- [ ] **22. Write link text that makes sense out of context (2.4.4)**

Important · You · 15 min

Screen reader users often list all links on a page.

Done when: No links that just say "click here", "more" or "read more" without context.

How to check: The Accessibility Checker lists vague link text.

Common mistake: Several "Learn more" links going to different pages.

Check it free with the Accessibility Checker · Source 1

- [ ] **23. Keep help in the same place on every page (3.2.6, new in 2.2)**

When you have time · Developer · 10 min

If you offer contact details, a chat or a help link, it should appear in the same relative place across pages.

Done when: Phone, contact link or chat sit in the same spot site-wide.

How to check: Click through five pages.

Common mistake: A contact button that moves around between templates.

Source 2

- [ ] **24. Publish an accessibility statement**

When you have time · You · 1 hour

It tells users what you support and how to report a problem. Not a WCAG requirement, but expected by many laws and buyers.

Done when: A page with your target standard (WCAG 2.2 AA), known issues and a contact for problems.

How to check: Link it in the footer.

Common mistake: Claiming full compliance without testing.

Check it free with the Accessibility Checker

Sources (checked November 9, 2026)

1. Web Content Accessibility Guidelines (WCAG) 2.2, W3C
2. What's new in WCAG 2.2, W3C WAI