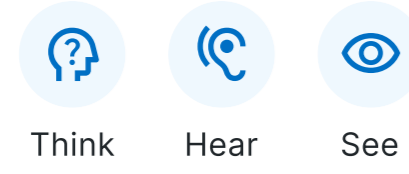


Perceivable



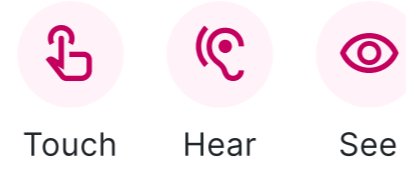
Your website **MUST** present information in a way people can recognise & understand

- ✓ You **MUST**
 - Make sure the colour of text and UI elements displays clearly against the background colour and adjacent elements
 - Describe each image with a text description (alt-tag)
 - Check content is understandable when the user changes their zoom settings or font size
 - Check the content is in the right order with the styling removed
- ✗ You **MUST NOT**
 - Rely on colour, size, shape or sound alone to communicate meaning, provide instructions or distinguish visual elements
 - Rely solely on proximity of content and functions

WCAG Perceivable: WCAG Criteria

- 1.1.1 **Non-text content** Level A
Provide alt tags for images
- 1.3.1 **Info and relationships** Level A
Content structures are also communicated in ways that assistive technologies can understand
- 1.3.2 **Meaningful sequence** Level A
Content can be read in a logical order, even when stylesheets are disabled.
- 1.3.3 **Sensory characteristics** Level A
Do not use colour, size, shape, sound or location as the only way to convey instructions
- 1.4.1 **Use of colour** Level A
Do not use colour alone to convey information
- 1.4.3 **Contrast minimum** Level AA
The colour of text contrasts clearly against the background colour.
 - 3:1 for text larger than 24px, or 19px bold
 - 4.5:1 for text smaller than 24px, or 19px bold
- 1.4.4 **Resize text (in the browser)** Level AA
All tasks can be completed when text is resized up to 200%.
- 1.4.10 **Reflow** Level AA
Content will reflow when zoomed. All content and functionality is available on 320px wide screens.
- 1.4.11 **Non-text contrast** Level AA
Non-text elements (icons, form fields, controls) have at least a 3:1 contrast ratio against the background

Operable



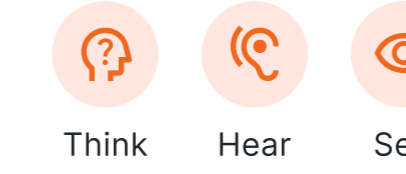
Users **MUST** be able to access your content by mouse, voice or keyboard.

- ✓ You **MUST**
 - Write meaningful headings and labels
 - Make sure heading levels follow the hierarchical page order of the content, over their visual styles and placement
 - Provide descriptive links so users know where a link will take them or the action it will perform
 - Make sure users can navigate in a logical order using just the keyboard and tell which element has keyboard focus
- ✗ You **MUST NOT**
 - Use labels like “read more”, “click here” or “edit” — instead, describe where the link will lead
 - Add a focus state to anything static that isn't an interactive element (use aria-hidden)

WCAG Operable: WCAG Criteria

- 2.1.1 **Keyboard** Level A
Every task can be completed using only a keyboard
- 2.1.2 **Keyboard trap** Level A
Keyboard users do not get stuck when navigating through content.
- 2.1.4 **Character keyboard shortcut** Level A
Keyboard shortcuts can be re-mapped or switched off.
- 2.4.1 **Bypass blocks** Level A
Keyboard and screen reader users can skip to the main content.
- 2.4.2 **Page titled** Level A
Every page has a unique and helpful title that indicates the purpose of the page.
- 2.4.3 **Focus order** Level A
Interactive controls receive focus in an order that makes sense, when users navigate through them with the keyboard.
- 2.4.4 **Link purpose** Level A
The purpose of every link is clear from the link text alone.
- 2.4.6 **Headings and labels** Level AA
Headings and labels describe the topic or purpose of the content in the section or field.
- 2.4.7 **Focus visible** Level AA
It is easy to tell which element has keyboard focus

Understandable



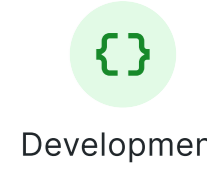
Your website **MUST** be easy for people to use and understand.

- ✓ You **MUST**
 - Use plain English and short sentences
 - Be predictable — use similar navigation controls
 - Use form field labels — these must be descriptive, instructive and permanently visible
 - Make errors easy to understand. Explain what happened, how to fix and show input format
- ✗ You **MUST NOT**
 - Surprise the user by automatically triggering actions
 - Be random — always use the same icons and naming conventions
 - Auto-advance form fields

WCAG Understandable: WCAG Criteria

- 3.2.1 **On Focus** Level A
Navigating to an interactive control with the keyboard doesn't trigger any action, or move the keyboard focus somewhere else
- 3.2.2 **On Input** Level A
Changing the state of a form input (like a radio button) does not cause anything surprising to happen, like submitting a form, significantly changing the content on the page, or moving the keyboard focus.
- 3.2.3 **Consistent navigation** Level AA
Navigation controls are consistent across pages.
- 3.2.4 **Consistent identification** Level AA
If a user interface component exists on multiple pages, the way it looks and the way it is named is consistent across pages.
- 3.3.1 **Error identification** Level A
Form errors are clearly described and identified.
- 3.3.2 **Labels or instructions** Level A
Labels make it clear how users should fill in a form, and optionally provide extra hints to help them avoid errors.
- 3.3.4 **Error prevention** Level AA
Users who are making a legal commitment, a financial transaction or updating personal data, can review and check the information they've entered before submitting it.

Robust



Your website **MUST** work with different web browsers and assistive technologies.

- ✓ You **MUST**
 - Make sure all text that looks and acts like a heading has a HTML heading tag
 - Make sure code works reliably across all supported browsers and assistive technologies
 - Let the user return to what they were doing after they've interacted with a status message or modal input
- ✗ You **MUST NOT**
 - Make sure layout relationships conveyed visually are also identified in the code
 - Make sure input fields are labelled in code

WCAG Robust: WCAG Criteria

- 3.2.1 **Parsing** Level A
The HTML does not contain markup errors that are known to cause conflicts with assistive technologies (such as incorrect nesting of elements, or duplicate ids).
- 3.2.2 **Name, role, value** Level A
The code enables assistive technologies to understand the name, role and state of every user interface component.
- 3.2.3 **Status messages** Level AA
Status messages are identified in code, so that assistive technologies can convey them to users.