Ideal HTML

Web Accessibility Standards

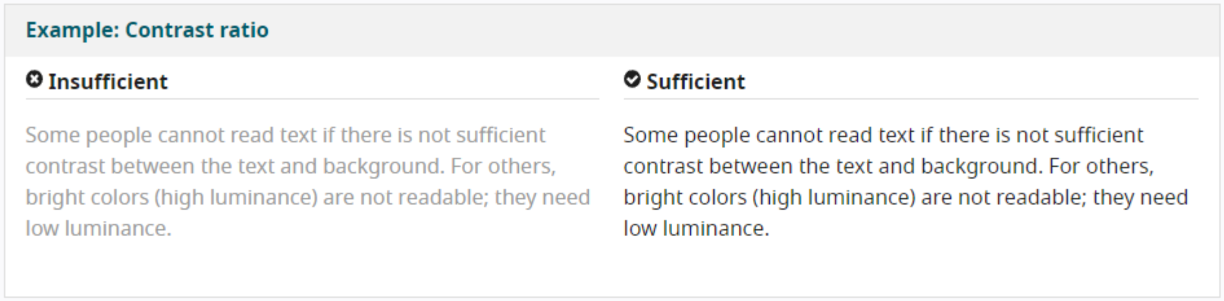
Source:

WAI (Web Accessibility Initiative)
https://www.w3.org/WAI/tips/
This page introduces some basic considerations to help you get started making your website more accessible to people with disabilities.

- Provide sufficient contrast between foreground and background.
- Don’t use color alone to convey information.
- Ensure that interactive elements are easy to identify.
- Provide clear and consistent navigation options.
- Ensure that form elements include clearly associated labels.
- Provide easily identifiable feedback.
- Use headings and spacing to group related content.
- Create designs for different viewport sizes.
- Include image and media alternatives in your design.
- Provide controls for content that starts automatically.
- Image Alternatives

Provide sufficient contrast between foreground and background

Foreground text needs to have sufficient contrast with background colors. This includes text on images, background gradients, buttons, and other elements. This does not apply for logos, or incidental text, such as text that happens to be in a photograph.



Example: Contrast ratio

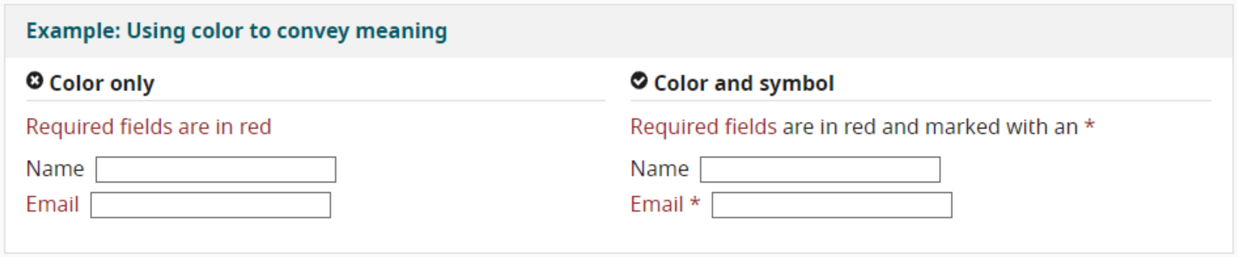
Insufficient

Some people cannot read text if there is not sufficient contrast between the text and background. For others, bright colors (high luminance) are not readable; they need low luminance.

Sufficient

Some people cannot read text if there is not sufficient contrast between the text and background. For others, bright colors (high luminance) are not readable; they need low luminance.

Don’t use color alone to convey information.



Example: Using color to convey meaning

Color only

Required fields are in red

Name

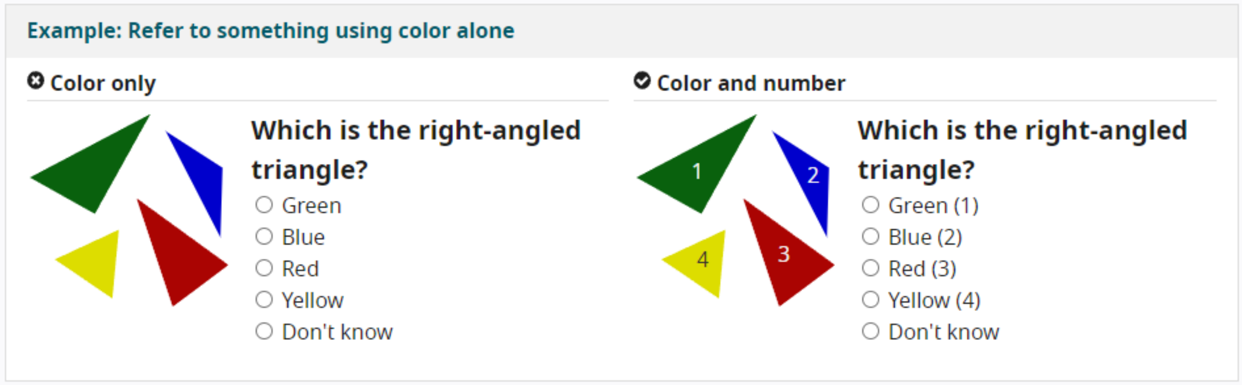
Email

Color and symbol

Required fields are in red and marked with an *

Name

Email *



Example: Refer to something using color alone

Color only

Which is the right-angled triangle?

- ○ Green
- ○ Blue
- ○ Red
- ○ Yellow
- ○ Don't know

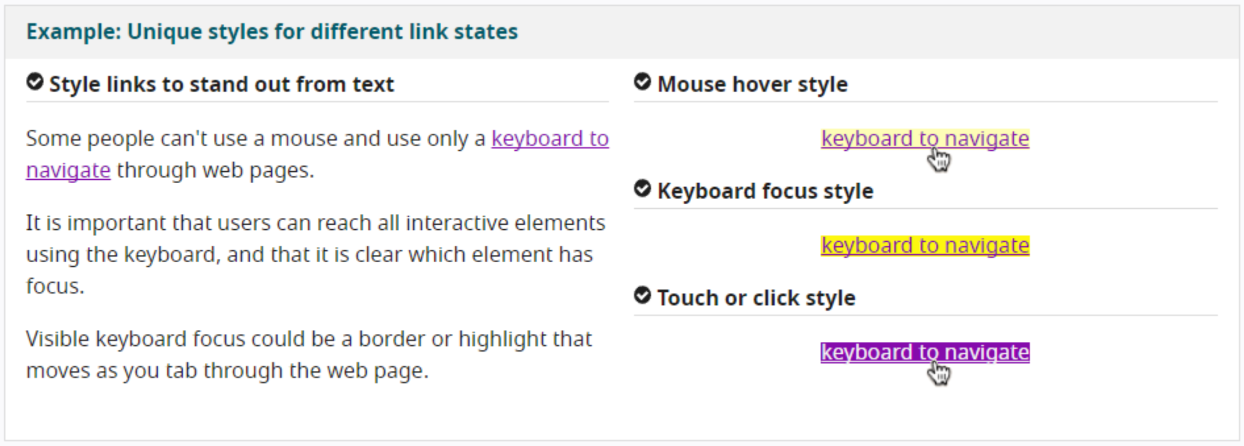
Color and number

Which is the right-angled triangle?

- ○ Green (1)
- ○ Blue (2)
- ○ Red (3)
- ○ Yellow (4)
- ○ Don't know

Ensure that interactive elements are easy to identify

Provide distinct styles for interactive elements, such as links and buttons, to make them easy to identify. For example, change the appearance of links on mouse hover, keyboard focus, and touch-screen activation. Ensure that styles and naming for interactive elements are used consistently throughout the website.


Example: Unique styles for different link states

Style links to stand out from text

Some people can't use a mouse and use only a keyboard to navigate through web pages.

It is important that users can reach all interactive elements using the keyboard, and that it is clear which element has focus.

Visible keyboard focus could be a border or highlight that moves as you tab through the web page.

Mouse hover style

keyboard to navigate

Keyboard focus style

keyboard to navigate

Touch or click style

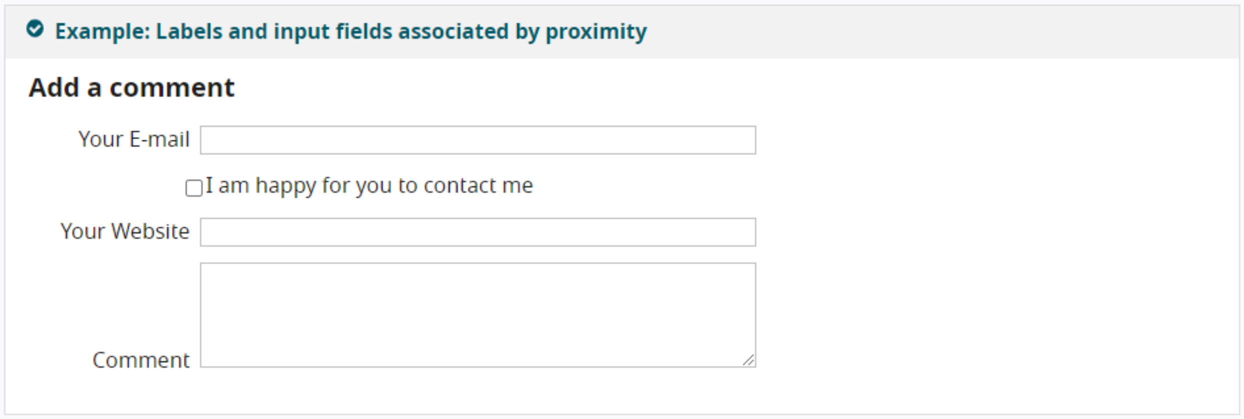
keyboard to navigate

Provide clear and consistent navigation options

Ensure that navigation across pages within a website has consistent naming, styling, and positioning. Provide more than one method of website navigation, such as a **site search** or a **site map**. Help users understand where they are in a website or page by providing orientation cues, such as **breadcrumbs** and **clear headings**.

Ensure that form elements include clearly associated labels

Ensure that all fields have a descriptive label adjacent to the field. For left-to-right languages, labels are usually positioned to the left or above the field, except for checkboxes and radio buttons where they are usually to the right. Avoid having too much space between labels and fields.


Example: Labels and input fields associated by proximity

Add a comment

Your E-mail

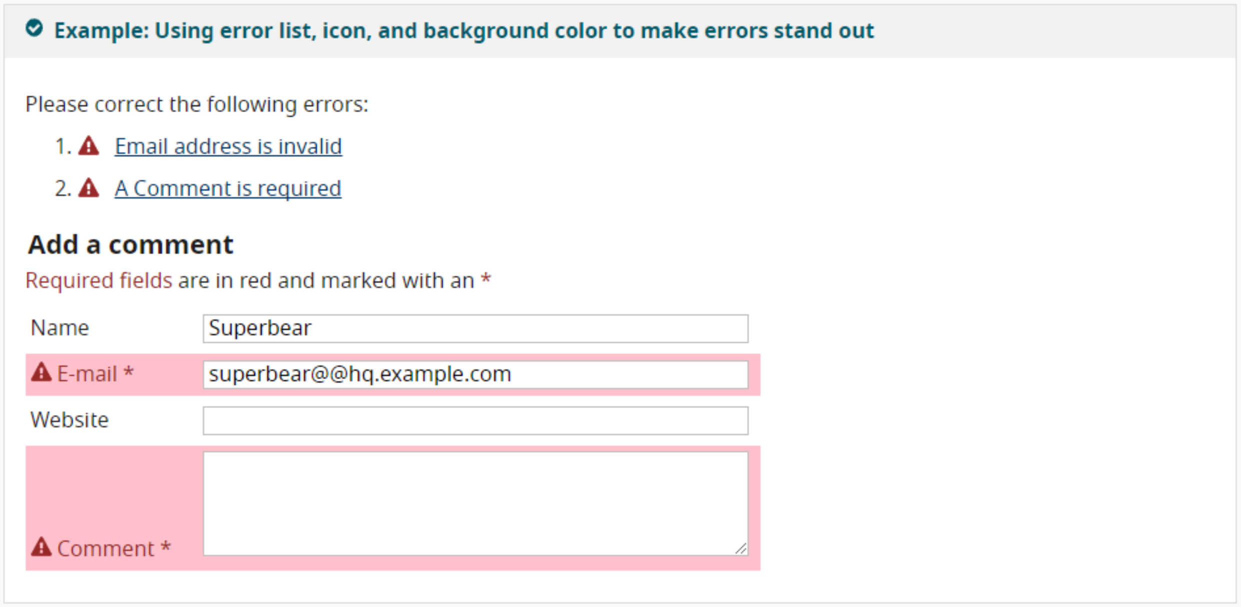
I am happy for you to contact me

Your Website

Comment

Provide easily identifiable feedback

Provide feedback for interactions, such as confirming form submission, alerting the user when something goes wrong, or notifying the user of changes on the page. Instructions should be easy to identify. Important feedback that requires user action should be presented in a prominent style.


Example: Using error list, icon, and background color to make errors stand out

Please correct the following errors:

1. Email address is invalid
2. A Comment is required

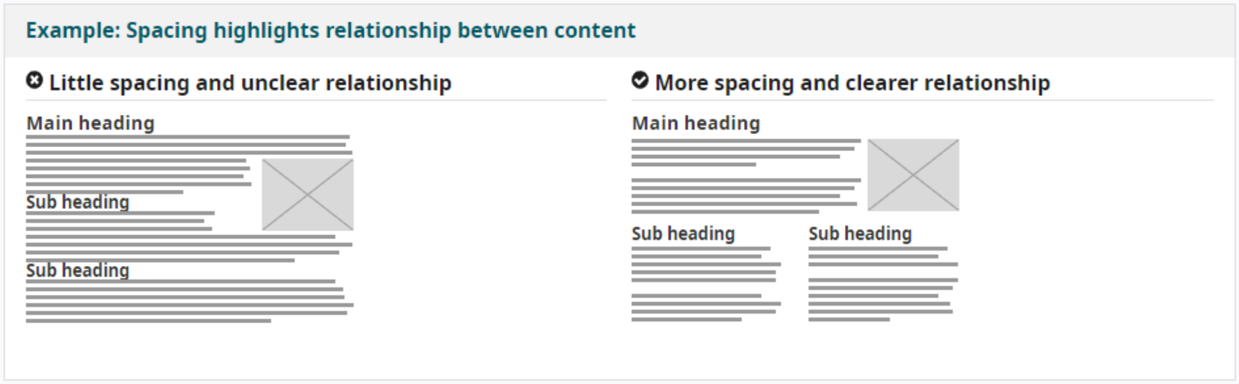
Add a comment

Required fields are in red and marked with an *

Name	Superbear
E-mail *	superbear@@hq.example.com
Website	
Comment *	

Use headings and spacing to group related content

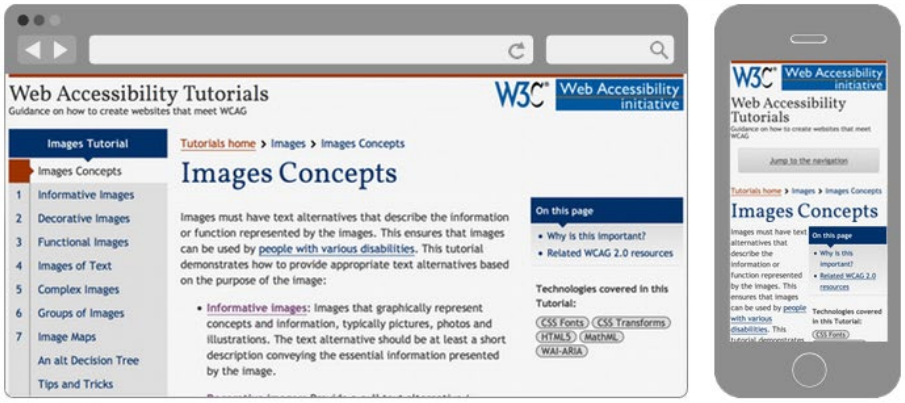
Use whitespace and proximity to make relationships between content more apparent. Style headings to group content, reduce clutter, and make it easier to scan and understand.


Example: Spacing highlights relationship between content
Little spacing and unclear relationship
Main heading
Sub heading
Sub heading
More spacing and clearer relationship
Main heading
Sub heading
Sub heading

Create designs for different viewport sizes

Consider how page information is presented in different sized viewports, such as mobile phones or zoomed browser windows. Position and presentation of main elements, such as header and navigation can be changed to make best use of the space. Ensure that text size and line width are set to maximize readability and legibility.

Example: Content and navigation adapt to smaller mobile screen



Web Accessibility Tutorials
Guidance on how to create websites that meet WCAG

Images Tutorial
Images Concepts
1 Informative Images
2 Decorative Images
3 Functional Images
4 Images of Text
5 Complex Images
6 Groups of Images
7 Image Maps
An alt Decision Tree
Tips and Tricks

Tutorials home > Images > Images Concepts

Images Concepts

Images must have text alternatives that describe the information or function represented by the images. This ensures that images can be used by people with various disabilities. This tutorial demonstrates how to provide appropriate text alternatives based on the purpose of the image:

- Informative images: Images that graphically represent concepts and information, typically pictures, photos and illustrations. The text alternative should be at least a short description conveying the essential information presented by the image.

On this page
- Why is this important?
- Related WCAG 2.0 resources

Technologies covered in this Tutorial:
CSS Fonts, CSS Transforms, HTML5, MathML, WAI-ARIA

Display in a wide window with small text uses multiple columns for primary content, visible navigation options, and visible secondary information.

Display in a narrow window, such as a mobile phone, or with large text uses single column for primary content, navigation options are revealed using an icon, and secondary information is also revealed via icon.

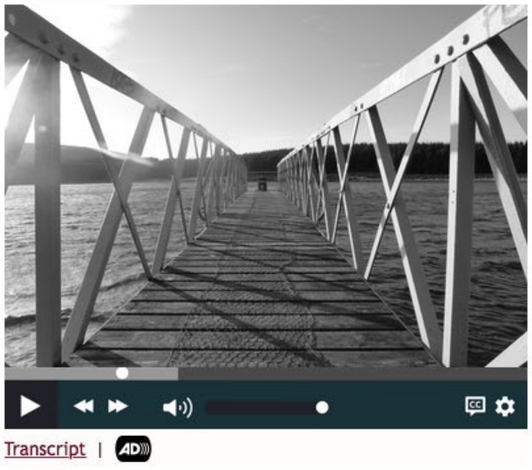
Include image and media alternatives in your design

Provide a place in your design for alternatives for images and media. For example, you might need:

- Visible links to transcripts of audio
- Visible links to audio described versions of videos
- Text along with icons and graphical buttons
- Captions and descriptions for tables or complex graphs

Work with content authors and developers to provide alternatives for non-text content.

Example: Design includes links to a transcript and to an audio described video



Provide controls for content that starts automatically

Provide visible controls to allow users to stop any animations or auto-playing sound. This applies to carousels, image sliders, background sound, and videos.

Example: Show play/stop and selection controls in carousel design

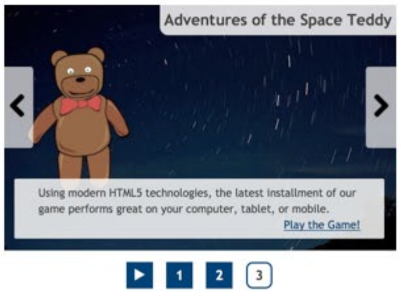

Adventures of the Space Teddy
Using modern HTML5 technologies, the latest installment of our game performs great on your computer, tablet, or mobile.
Play the Game!
1
2
3

Image Alternatives

Images must have text alternatives that describe the information or function represented by them. This ensures that images can be used by people with various disabilities.

Examples of text alternatives for images:

Informative images: Images that graphically represent concepts and information, typically pictures, photos, and illustrations. The text alternative should be at least a short description conveying the essential information presented by the image.

Decorative images: Provide a null text alternative (alt="") when the only purpose of an image is to add visual decoration to the page, rather than to convey information that is important to understanding the page.

Functional images: The text *alternative of an image used as a link or as a button* should describe the functionality of the link or button rather than the visual image. Examples of such images are a printer icon to represent the print function or a button to submit a form.

Images of text: Readable text is sometimes presented within an image. If the image is not a logo, avoid text in images. However, if images of text are used, the text alternative should contain the same words as in the image.

Complex images such as graphs and diagrams: To convey data or detailed information, provide a complete text equivalent of the data or information provided in the image as the text alternative.

Groups of images: If multiple images convey a single piece of information, the text alternative for one image should convey the information for the entire group.

Image maps: The text alternative for an image that contains multiple clickable areas should provide an overall context for the set of links. Also, each individually clickable area should have alternative text that describes the purpose or destination of the link.

Why Image Alternatives are important?

Images and graphics make content more pleasant and easier to understand for many people, and in particular for those with cognitive and learning disabilities.

They serve as cues that are used by people with visual impairments, including people with low vision, to orient themselves in the content.

However, images are used extensively on websites and can create major barriers when they are not accessible. Accessible images are beneficial in many situations, such as:

1) People using screen readers: The text alternative can be read aloud or rendered as Braille
2) People using speech input software: Users can put the focus onto a button or linked image with a single voice command
3) People browsing speech-enabled websites: The text alternative can be read aloud
4) Mobile web users: Images can be turned off, especially for data-roaming
5) Search engine optimization: Images become indexable by search engines

Writing for Web Accessibility - Table of Contents

Provide informative, unique page titles

For each web page, provide a short title that describes the page content and distinguishes it from other pages. The page title is often the same as the main heading of the page. Put the unique and most relevant information first; for example, put the name of the page before the name of the organization. For pages that are part of a multi-step process, include the current step in the page title.

Example: Page Titles

✓ Home page title

Space Teddy Inc.

✓ Page name followed by organization name

Latest News • Space Teddy Inc.

✓ Page name including step in a process

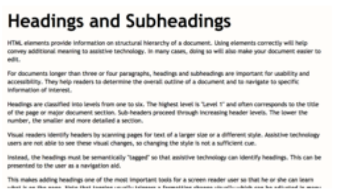
Buy Your Bear (Step 1 of 3) • Space Teddy Inc.

Use headings to convey meaning and structure

Use short headings to group related paragraphs and clearly describe the sections. Good headings provide an outline of the content.

Example: Using headings to organize content

Lack of headings



Headings and Subheadings

HTML elements provide information on structural hierarchy of a document. Using elements correctly will help convey additional meaning to assistive technology. In many cases, doing so will also make your document easier to edit.

For documents longer than three or four paragraphs, headings and subheadings are important for usability and accessibility. They help readers to determine the overall outline of a document and to navigate to specific information of interest.

Headings are classified into levels from one to six. The highest level is "Level 1" and often corresponds to the title of the page or major document section. Sub-headers proceed through increasing header levels. The lower the number, the smaller and more detailed a section.

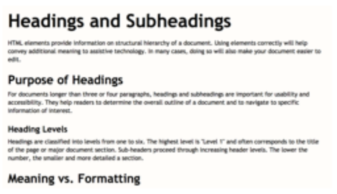
Visual readers identify headers by scanning pages for text of a larger size or a different style. Assistive technology users are not able to see these visual changes, so changing the style is not a sufficient cue.

Instead, the headings must be semantically "tagged" so that assistive technology can identify headings. This can be presented to the user as a navigation aid.

This makes adding headings one of the most important tools for a screen reader user so that he or she can learn

View inline example

✓ Using headings and subheadings



Headings and Subheadings

HTML elements provide information on structural hierarchy of a document. Using elements correctly will help convey additional meaning to assistive technology. In many cases, doing so will also make your document easier to edit.

Purpose of Headings

For documents longer than three or four paragraphs, headings and subheadings are important for usability and accessibility. They help readers to determine the overall outline of a document and to navigate to specific information of interest.

Heading Levels

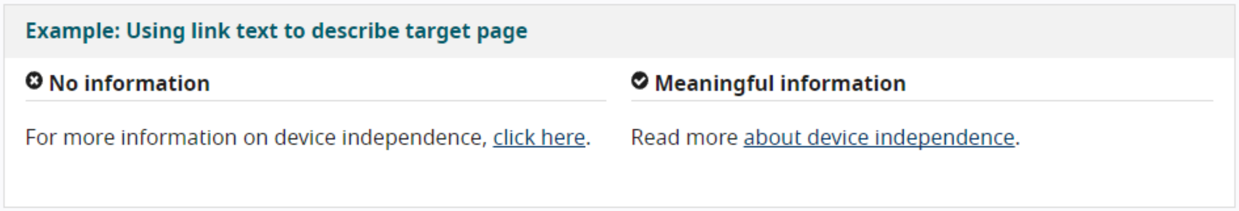
Headings are classified into levels from one to six. The highest level is "Level 1" and often corresponds to the title of the page or major document section. Sub-headers proceed through increasing header levels. The lower the number, the smaller and more detailed a section.

Meaning vs. Formatting

View inline example

Make link text meaningful

Write link text so that it describes the content of the link target. Avoid using ambiguous link text, such as ‘click here’ or ‘read more’. Indicate relevant information about the link target, such as document type and size, for example, ‘Proposal Documents (RTF, 20MB)’.

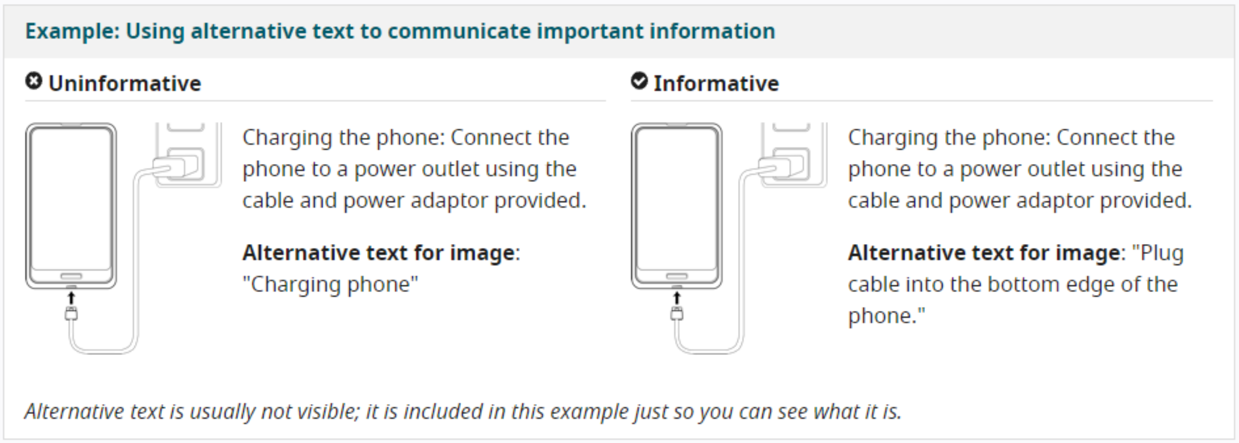


Example: Using link text to describe target page

✖ No information	✓ Meaningful information
For more information on device independence, click here.	Read more about device independence.

Write meaningful text alternatives for images

For every image, write alternative text that provides the information or function of the image. For purely decorative images, there is no need to write alternative text.



Example: Using alternative text to communicate important information

✖ Uninformative	✓ Informative
Charging the phone: Connect the phone to a power outlet using the cable and power adaptor provided. **Alternative text for image**: "Charging phone"	Charging the phone: Connect the phone to a power outlet using the cable and power adaptor provided. **Alternative text for image**: "Plug cable into the bottom edge of the phone."

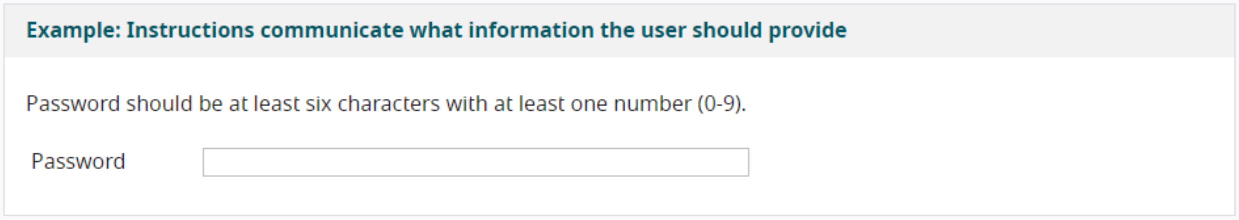
Alternative text is usually not visible; it is included in this example just so you can see what it is.

Create transcripts and captions for multimedia

For audio-only content, such as a podcast, provide a transcript. For audio and visual content, such as training videos, also provide captions. Include in the transcripts and captions the spoken information and sounds that are important for understanding the content, for example, ‘door creaks’. For video transcripts, also include a description of the important visual content, for example ‘Athan leaves the room’. Example: Making Audio and Video Media Accessible.

Provide clear instructions

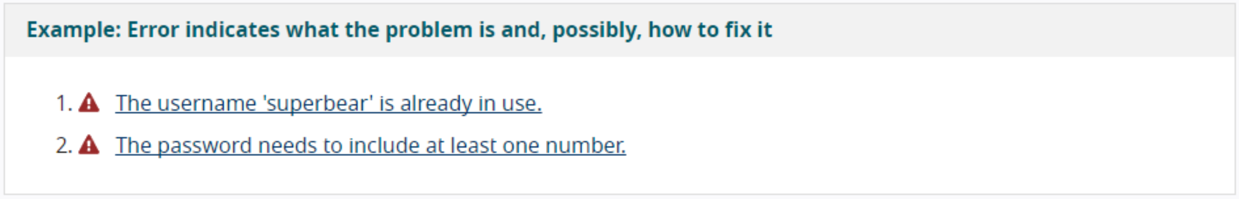
Ensure that instructions, guidance, and error messages are clear, easy to understand, and avoid unnecessarily technical language. Describe input requirements, such as date formats.



Example: Instructions communicate what information the user should provide

Password should be at least six characters with at least one number (0-9).

Password



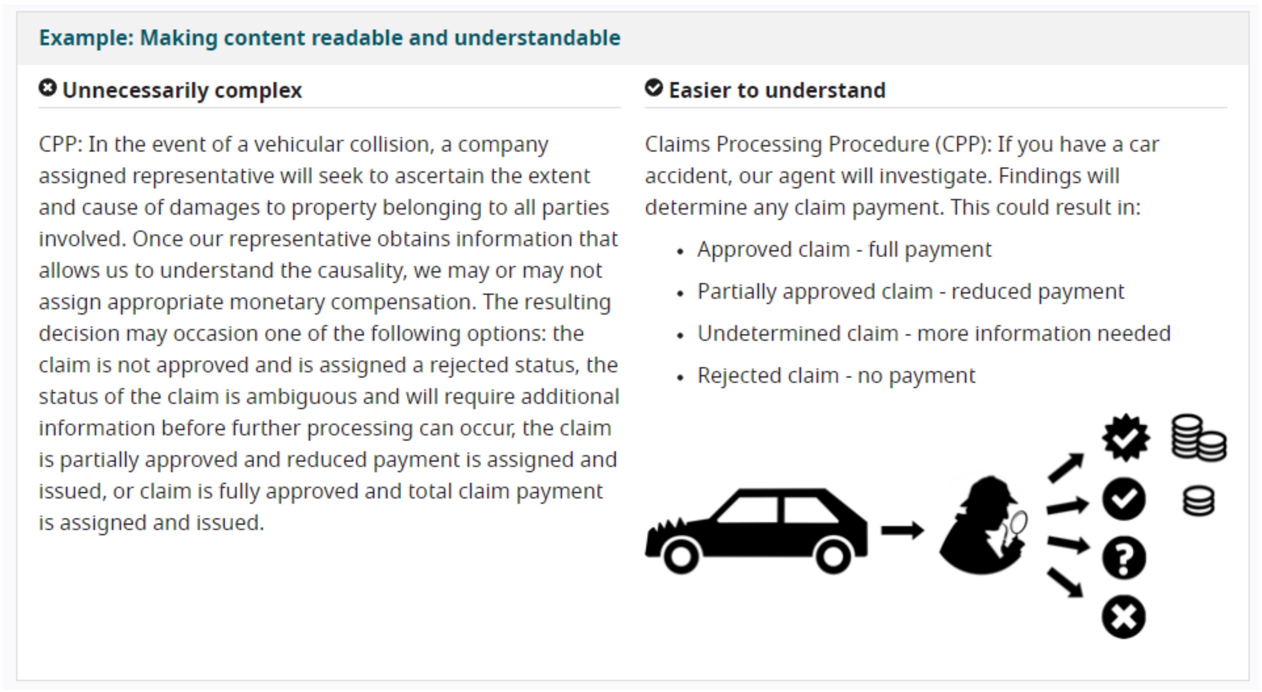
Example: Error indicates what the problem is and, possibly, how to fix it

1. The username 'superbear' is already in use.
2. The password needs to include at least one number.

Keep content clear and concise

Use simple language and formatting, as appropriate for the context.

- Write in short, clear sentences and paragraphs.
- Avoid using unnecessarily complex words and phrases.
- Expand acronyms on first use. For example, Web Content Accessibility Guidelines (WCAG).
- Consider providing a glossary for terms readers may not know.
- Use list formatting as appropriate.
- Consider using images, illustrations, video, audio, and symbols to help clarify meaning.



Example: Making content readable and understandable

Unnecessarily complex

CPP: In the event of a vehicular collision, a company assigned representative will seek to ascertain the extent and cause of damages to property belonging to all parties involved. Once our representative obtains information that allows us to understand the causality, we may or may not assign appropriate monetary compensation. The resulting decision may occasion one of the following options: the claim is not approved and is assigned a rejected status, the status of the claim is ambiguous and will require additional information before further processing can occur, the claim is partially approved and reduced payment is assigned and issued, or claim is fully approved and total claim payment is assigned and issued.

Easier to understand

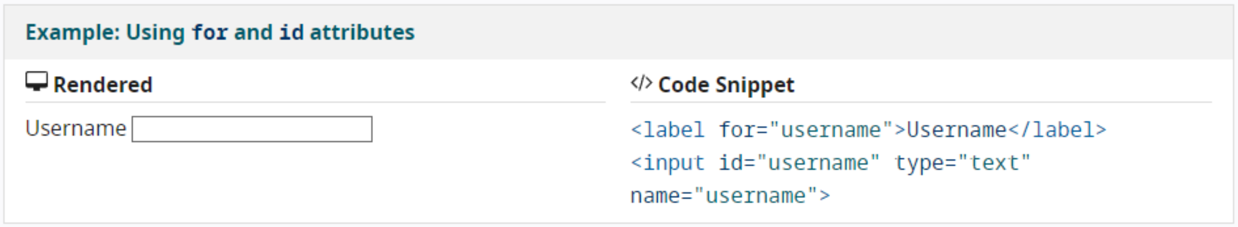
Claims Processing Procedure (CPP): If you have a car accident, our agent will investigate. Findings will determine any claim payment. This could result in:

- Approved claim - full payment
- Partially approved claim - reduced payment
- Undetermined claim - more information needed
- Rejected claim - no payment

Devloping for Web Accessibility - Table of Contents

Associate a label with every form control

Use a `for` attribute on the `<label>` element linked to the `id` attribute of the form element, or using WAI-ARIA attributes. In specific situations it may be acceptable to hide `<label>` elements visually, but in most cases labels are needed to help all readers understand the required input.



Example: Using `for` and `id` attributes

Rendered

Username

Code Snippet

```
<label for="username">Username</label>
<input id="username" type="text"
name="username">
```

Include alternative text for images

Ensure that alternative text for images is added to all informational and functional images. Use empty alternative text, `alt=""` for decorative images, or include them in the CSS instead. Text alternatives are usually provided by those responsible for written content.

Identify page language and language changes

Indicate the primary language of every page by using the `lang` attribute in the `html` tag, for example `<html lang="en">`. Use the `lang` attribute on specific elements when the language of the element differs from the rest of the page.

Use mark-up to convey meaning and structure

Use appropriate mark-up for headings, lists, tables, etc. HTML5 provides additional elements, such as `<nav>` and `<aside>`, to better structure your content. WAI-ARIA roles can provide additional meaning, for example, using `role="search"` to identify search functionality. Work with designers and content writers to agree on meanings and then use them consistently.

Example: Using HTML to provide structure and meaning

Rendered

Superbear saves the day

7 Aug 2015

The city's favorite bear yet again proves his mettle by rescuing a young cat from a tree. Witnesses say that Superbear's efforts were not appreciated by the feline, who inflicted some minor scratch wounds on his rescuer.

Related Articles

- Bear receives key to city
- Superbear stands for mayor

Code Snippet

```
<section>
  <article>
    <h2>Superbear saves the day</h2>
    <time datetime="2015-08-07">7 Aug
2015</time>
    <p>The city's favorite bear yet again
proves his mettle by rescuing a young cat from
a tree. Witnesses say that Superbear's efforts
were not appreciated by the feline, who
inflicted some minor scratch wounds on his
rescuer.</p>
    <aside>
      <h3>Related Articles</h3>
      <ul>
        <li><a href="#">Bear receives key to
city</a></li>
        <li><a href="#">Superbear stands for
mayor</a></li>
      </ul>
    </aside>
  </article>
</section>
```

Example: Search field using WAI-ARIA

Rendered

Search for [   ]
Search records by customer id or name
Go

Code Snippet

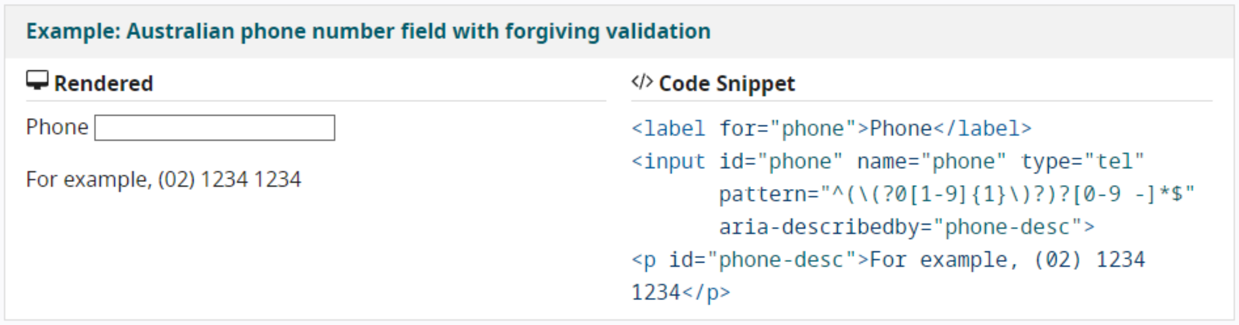
```
<form action="#" method="post">
  <div role="search">
    <label for="search">Search for</label>
    <input type="search" id="search" aria-
describedby="search-help">
    <div id="search-help">Search records by
customer id or name</div>
    <button type="submit">Go</button>
  </div>
</form>
```

Help users avoid and correct mistakes

Provide clear instructions, error messages, and notifications to help users complete forms on your site. When an error occurs:

- Help users find where the problem is
- Provide specific, understandable explanations
- Suggest corrections

Be as forgiving of format as possible when processing user input. For example, accept phone numbers that include spaces and delete the spaces as needed.



Example: Australian phone number field with forgiving validation

Rendered

Phone

For example, (02) 1234 1234

Code Snippet

```
<label for="phone">Phone</label>
<input id="phone" name="phone" type="tel"
       pattern="^(\(?0[1-9]{1}\)?)?[0-9 -]*$"
       aria-describedby="phone-desc">
<p id="phone-desc">For example, (02) 1234
1234</p>
```

Reflect the reading order in the code order

Ensure that the order of elements in the code matches the logical order of the information presented. One way to check this is to remove CSS styling and review that the order of the content makes sense.

Example: Reflecting the logical reading order in the code

Space trainers

Space trainer for a classic and stylish look.

Add to cart

✗ Image before heading may be missed

```
<img src="images/trainer.png" alt="...">
<h3>Space trainers</h3>
<p>Space...</p>
<a href="...">Add to cart</a>
```

+ View complete code example

✓ Heading marks the start of the section

```
<h3>Space trainers</h3>
<img src="images/trainer.png" alt="...">
<p>Space...</p>
<a href="...">Add to cart</a>
```

+ View complete code example

Write code that adapts to the user’s technology

Use responsive design to adapt the display to different zoom states and viewport sizes, such as on mobile devices and tablets. When font size is increased by at least 200%, avoid horizontal scrolling and prevent any clipping of content. Use progressive enhancement to help ensure that core functionality and content is available regardless of technology being used.

Example: Using media queries to adapt navigation

```
/* On narrow viewports, make the
navigation full width */
@media screen and (min-width: 25em) {
  #nav {
    float: none;
    width: auto;
  }
  #main {
    margin-left: 0;
  }
}
```

```
/* On wider viewports, put the navigation
on the left */
@media screen and (min-width: 43em) {
  #nav {
    float: left;
    width: 24%;
  }
  #main {
    margin-left: 27%;
  }
}
```

Provide meaning for non-standard interactive elements

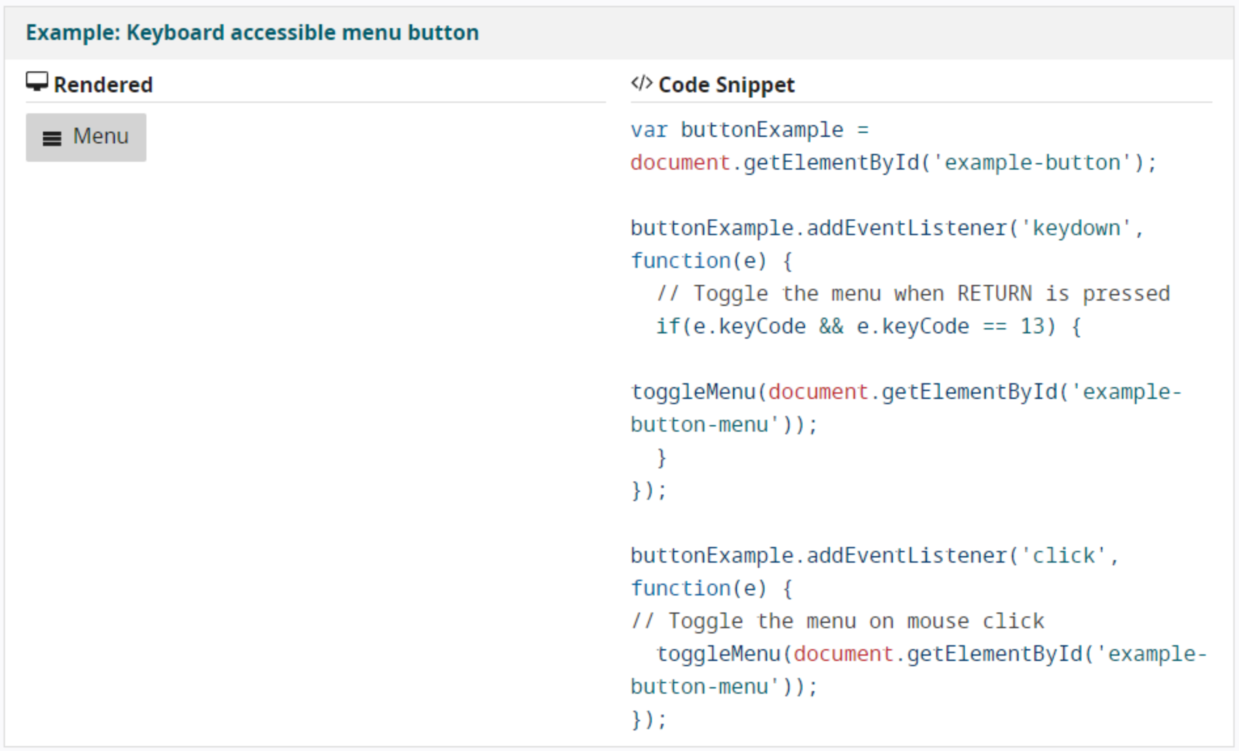
Use WAI-ARIA to provide information on function and state for custom widgets, such as accordions and custom-made buttons. For example, `role="navigation"` and `aria-expanded="true"`. Additional code is required to implement the behavior of such widgets, such as expanding and collapsing content or how the widget responds to keyboard events.

Example: Menu function and state identified using WAI-ARIA

```
<nav aria-label="Main Navigation" role="navigation">
  <ul>
    <li><a href="...">Home</a></li>
    <li><a href="...">Shop</a></li>
    <li class="has-submenu">
      <a aria-expanded="false" aria-haspopup="true" href="...">SpaceBears</a>
      <ul>
          <li><a href="...">SpaceBear 6</a></li>
          <li><a href="...">SpaceBear 6 Plus</a></li>
      </ul>
    </li>
    <li><a href="...">MarsCars</a></li>
    <li><a href="...">Contact</a></li>
  </ul>
</nav>
```

Ensure that all interactive elements are keyboard accessible

Think about keyboard access, especially when developing interactive elements, such as menus, mouseover information, collapsable accordions, or media players. Use `tabindex="0"` to add an element that does not normally receive focus, such as `<div>` or `<span>`, into the navigation order when it is being used for interaction. Use scripting to capture and respond to keyboard events.



```
var buttonExample =
document.getElementById('example-button');

buttonExample.addEventListener('keydown',
function(e) {
  // Toggle the menu when RETURN is pressed
  if(e.keyCode && e.keyCode == 13) {

toggleMenu(document.getElementById('example-
button-menu'));
  }
});

buttonExample.addEventListener('click',
function(e) {
// Toggle the menu on mouse click
  toggleMenu(document.getElementById('example-
button-menu'));
});
```

Avoid CAPTCHA where possible

CAPTCHAs create problems for many people. There are other means of verifying that user input was generated by a human that are easier to use, such as automatic detection or interface interactions. If CAPTCHA really needs to be included, ensure that it is simple to understand and includes alternatives for users with disabilities, such as:

- Providing more than two ways to solve the CAPTCHAs
- Providing access to a human representative who can bypass CAPTCHA
- Not requiring CAPTCHAs for authorized users.