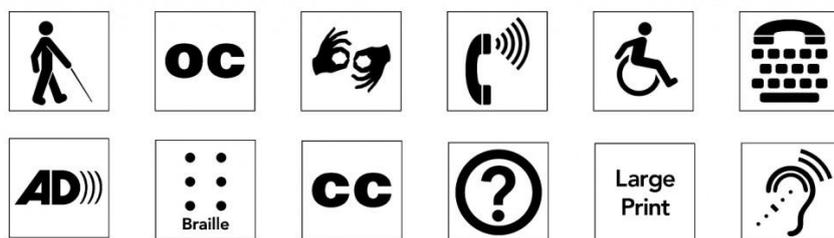


Accessibility on the web – an introduction



This document is intended as a general introduction to web accessibility. Specific guidance for web editors using the University's corporate content management system – OpenText – is published on the [Digital Resources Hub](#).

Accessibility on the web

The web offers many opportunities for people with disabilities that are unavailable through any other medium. It offers independence and freedom. However, if a web site is not created with web accessibility in mind, it may exclude a segment of the population that stands to gain the most from the internet. Most people do not intend to exclude people with disabilities. As organizations and designers become aware of and implement accessibility, they will ensure that their content can be accessed by a broader population.

“The social value of the Web is that it enables human communication, commerce, and opportunities to share knowledge. One of W3C's primary goals is to make these benefits available to all people, whatever their hardware, software, network infrastructure, native language, culture, geographical location, or physical or mental ability.”

[W3C Design Principles](#)

The work of the World Wide Web Consortium (W3C) is founded on the principles of compatibility, utility, interoperability and universal access¹. There's no doubt that the web is an empowering and transformative resource, and anyone can contribute.

To ensure that the work we do continues to benefit the most people, there are a few simple things we can do when publishing on the web that can make a big difference. To that end, this document provides an overview of some of the most important principles of accessible design. If you want to dive into the details, check the “links” section of this Toolkit resource.

Content

Headers for data tables

Tables are used to organize data and present it in rows and columns. Each row could represent a different item in a store, and the table columns could contain data of the same type for each row, like a quantity or a price. It's important to provide context for the data, but we often see tables where the column headings are only visually distinguished, for example by using emboldened text. To search engines and assistive technologies, these headings look just like data!

To give context to the data in tables, use the <th> element for column headings.

¹ <https://www.w3.org/TR/html-design-principles/>

Find out more about table headers at <https://webaim.org/techniques/tables>

Heading hierarchy

Content on a page is usually split into sections, with headings to indicate what the sections are about. In the code behind the page, these headings are given numbers, e.g. heading 1, heading 2, heading 3, and this helps to provide context about the page content to all users - including search engines.

Headings should follow a logical sequence, so don't skip levels. For example, don't jump from heading 2 to heading 4.

Find out more about hierarchy at <https://webaim.org/techniques/semanticstructure/>

Meaningful links

Links are possibly the most powerful features of the web; any page can link to any other. When you create a link you give it the address of the destination page, but you should also make sure that the text you transform into a link - the anchor - is meaningful. This helps all users, especially screen-reader users and search engines. **Don't use** "click here".

Links should open in the same tab/window. If you are linking to a document, let the user's browser default behaviour handle it.

Find out more about meaningful links at <https://webaim.org/techniques/hypertext>

Caption/transcripts for media

Videos and live audio must have captions and a transcript. With archived audio, a transcription may be enough.

Find out more about captioning at <https://webaim.org/techniques/captions/>

Descriptions for images ("alt text")

Alternative text provides a textual alternative to non-text content in web pages. It is especially helpful for people who are blind and rely on a screen reader to have the content of the website read to them.

So, if an image conveys meaning or has a function, add a description so that users of assistive technologies can also benefit from its context. This is even more important if you are using the image as a link anchor.

Even if the image is just for decoration, **never** leave the alt text blank. Simply add **alt=""** to the image source code. This lets a screen reader know that the image is not important to the content. (**Note:** OpenText automatically adds alt="" to all images that you haven't added a description to.)

Find out more about using image descriptions at <https://webaim.org/techniques/alttext/>

Accessible non-HTML content

This includes [PDF files](#), [Microsoft Word](#) documents, [PowerPoint](#) presentations and [Adobe Flash](#) content.

PDF documents and other non-HTML content must be as accessible as possible. If you cannot make it accessible, consider using HTML instead or, at the very least, provide an accessible alternative. PDF documents should also include a series of tags to make it more accessible. A tagged PDF file looks the same, but it is almost always more accessible to a person using a screen reader. See Toolkit's [Accessibility for Authors/Creators resource](#) for guidance on creating accessible documents.

Do not rely on colour to convey meaning

The use of colour can enhance comprehension, but do not use colour alone to convey information. That information may not be available to a person who is colourblind and will be unavailable to screen reader users.

Find out more about working with colour at <https://webaim.org/articles/visual/colorblind>

Content is clearly written and easy to read

There are many ways to make your content easier to understand. Write clearly, [use standard fonts](#), and [use headings and lists appropriately](#).

Find out more at writing clearly and simply at <https://webaim.org/techniques/writing/>

Code

Developers can also take steps to ensure the underlying code of a page is accessible to as many people as possible.

Keyboard navigation

Headings, lists, and other structural elements provide meaning and structure to web pages, and they can also facilitate keyboard navigation within a web page, for example by using the tab key.

Focus styles on elements should be used to clearly identify where the user's "focus" currently is. The default style is usually a dotted outline, but this can be improved upon to make it even clearer to all users.

Find out more about keyboard navigation at <https://webaim.org/techniques/keyboard/>

Forms can be completed and submitted

Ensure that every form element (text field, checkbox, dropdown list, etc.) has a label and make sure that label is associated to the correct form element using the <label> element. Also make sure the user can [submit the form and recover from any errors](#), such as the failure to fill in all required fields.

Find out more about accessible web forms at <https://webaim.org/techniques/forms>

Visible "skip to content" link

You should provide a method that allows users to skip navigation or other elements that repeat on every page. This is usually accomplished by providing a "Skip to Main Content," or "Skip Navigation" link at the top of the page which jumps to the main content of the page. This prevents keyboard users from having to tab through all links on a page to get to the content.

Find out more at <https://webaim.org/techniques/skipnav/>

Peek-a-boo

Many modern design patterns involve hiding and showing things, like tabs or accordions or carousels. Whenever you hide things, make sure that they can still be accessed by all users and search engines.

Find out more about show and hide at <https://webaim.org/techniques/javascript/>

WAI-ARIA

Much like alt text on images, there are times when a little contextual information can go a long way. Adding ARIA roles and attributes to your code can be a quick way to provide that information.

Find out more about ARIA roles and attributes at <https://www.w3.org/WAI/intro/aria>

Remember. [This is for everyone](#)².

² Quoted by Tim Berners-Lee, inventor of the World Wide Web, during his participation in the 2012 Summer Olympics opening ceremony.