Accessible Web Content Guide

## Authors:

James Church, Digital Accessibility Compliance Officer at Kent County Council
Sammy Bhurrut, Digital Support Officer at Kent County Council

## Version:

1.0

## Last Updated:

18.05.2020

Contents

[Text 2](#_gjdgxs)

[Images 3](#_30j0zll)

[Sensory & Colour 4](#_1fob9te)

[Video & Audio 5](#_3znysh7)

[Forms 6](#_2et92p0)

[Carousels 7](#_tyjcwt)

[Maps 8](#_3dy6vkm)

[Website Structure - Developer 8](#_1t3h5sf)

[3rd Party Content 14](#_4d34og8)

[Documents 14](#_2s8eyo1)

[Further Support 15](#_17dp8vu)

# Introduction

Under the [Public Sector Bodies (Website and Mobile Applications) Accessibility Regulations 2018](https://www.gov.uk/guidance/accessibility-requirements-for-public-sector-websites-and-apps), all websites and apps must meet accessibility standards ([WCAG 2.1 AA](https://www.w3.org/TR/WCAG21/)) and publish an accessibility statement explaining how accessible the website is. This includes all content and documents hosted on websites and systems.

When developing web-based systems and content, you are responsible for ensuring that the website content you produce meets the accessibility considerations detailed in this document. You must also ensure that existing web pages and content are reviewed and that any accessibility issues identified are resolved.

By following this guide when creating for the web, you will be improving user experience for all users.

# Text

Text is the most used element and therefore causes most accessibility issues on websites. However, minor tweaks to the way text is laid out can make a positive impact to all users.

For information on how to write in plain English, please see the [Plain English Guide](https://www.lexdis.org.uk/digital-accessibility/accessibility-tools/writing-in-plain-english/).

Additional Plain English resources:

* [gov.uk/guidance/content-design/user-needs](http://www.gov.uk/guidance/content-design/user-needs)
* [hemingwayapp.com](http://www.hemingwayapp.com/)

## Links

Links should be clearly defined to all users and assistive technologies, particularly when part of some text.

Ambiguous links throughout the page (‘Click Here’, ‘Read More’, a URL) will make it very difficult for the user to identify and determine the link’s purpose, without having to scan through the plain text content that surrounds it.

Links should have a link text that is clear and descriptive of the target.

## Fonts & Text Layout

All fonts used should be a Sans-Serif font such as Calibri or Aerial. These fonts have a structured letter spacing and line height plus are installed on most devices.

Text is best laid out inline as this helps with assistive technology compatibility and is the most natural way of reading, from left to right.

You should think about colour, size and emphasis when writing text.

* Ensure that the colour of the text contrasts sufficiently with the background colour so it is visible to most (minimum contrast of 4.5:1)
* Ensure that the default size of standard text is at least 11px
* Bold, underline and italic text are not always obvious so ensure that they are not the only way of conveying information.

# Images

By design, images are visual elements. The popular saying; ‘an image contains a thousand words’ is often true, which makes it vital that the same information can be provided to a user who cannot see the image.

## Alternate (Alt) Text

All images must be given a text alternative that is read out by a screen reader. This is in the form of an alt HTML tag. Most website editors will provide a field to input this for each image.

* Alt text should be meaningful and informative (1-2 sentences describing what the image is showing)
* If an image is purely for decoration and not providing information, the alt text should be left empty (alt=””).

For example, for the below image:

* A good alt would be “Kent County Council Logo”
* A bad alt would be “Image”.



## Text in Image

A visually impaired user may not be able to interpret text that is part of an image. This text should be placed elsewhere on the web page or in a different format.

Images must not contain text unless:

* Text is essential to what the image is conveying (e.g. a name on a name badge, company branding)
* Text is customisable (e.g. the image is an SVG which distinguishes the text to assistive technologies).

If text meets one of the above criteria and can be included in an image, you should still endeavour to make it as accessible as possible. This includes ensuring the alt text describes the image and its text suitably, and/or the image is elaborated on in accompanying text or similar.

Using the KCC logo (below) as an example, this contains text which is essential to the meaning of the image as branding and is therefore acceptable.



However, the following image contains text that is not essential to the image. This text should be presented as text on the web page or the image removed.



## Diagrams

Visual diagrams are images that often contain a lot of information and text that cannot be conveyed simply in a sentence as alt text. If including a diagram on a web page, you should ensure that the information being conveyed is described elsewhere on the page or available as an alternative format.

An example of how this could be done is using the longdesc HTML element to provide a long description of the diagram for assistive technologies. See [W3 Complex Images Approach 1](https://www.w3.org/WAI/tutorials/images/complex/#description-containing-structured-information) for more information on this technique.

# Sensory & Colour

The senses of every individual are tuned differently and by making content that only relies on one sense, you could be creating a barrier for users who cannot use that sense as expected.

Sensory characteristics of content can include shape, size, visual location, orientation, sound and colour.

Example: An instruction to click on the biggest green circle to place an order. Fix: ‘Click on the purchase button to order’

## Colour Contrast

Colour contrast is a key consideration when designing a website and content as picking two colours that do not contrast well can result in the information being lost to some users. You should ensure that:

* Contrast of text against its background is a minimum of 4.5:1. Ensure that text is against a solid-colour background.
* Contrast of a non-text element against its background is a minimum of 3:1, including focus indicators (the boxes highlighting an element when using Tab to navigate).

You should also consider the contrast within content, such as images and videos.

Check the contrast of two colours using [WebAIM’s Contrast Checker](https://webaim.org/resources/contrastchecker/).

## Screen Reader

All elements and content should be compatible with a screen reader. This can be done by using alternatives to non-text content where possible (e.g. alt tags on images) and ARIA tags (See ARIA developer’s section) if developing a site.

# Video & Audio

Videos are used extensively throughout the web in order to convey information or provide entertainment to users. The way video media is implemented can cause barriers to individuals, meaning that the content may not be accessible to them.

When including video content on your website, there are certain steps that can be taken to ensure the content is available to all users.

* The video player must be fully controllable via keyboard-only inputs to allow for keyboard-only users to access the content.
* Videos must not automatically play on page load as this can have a negative impact on the user due to an overlap of audio from multiple sources, which is especially bad for those using a screen reader.

## Closed Captions

Synchronised captions are required for all videos with the exception that the video is a media alternative for text content. Captions provide the user with a text version of speech and other audio content within the video, such as important background noises, whilst the video is playing. It is vital for those who are audibly impaired to be able to fully understand the content.

Captions should be ‘closed’ rather than ‘open’, meaning it can be deactivated and hidden if the user chooses. ‘Open’ or ‘burnt-in’ captions cannot be manipulated in terms of text size, colour and font; and are also not able to be understood by assistive technologies. Burnt in captions are also open to issues such as insufficient colour contrast and the overlapping and obscuring of content within the video.

## Subtitles

Subtitles are different to captions as they only account for the dialogue or verbal communication within a video, due to the assumption that the user is not deaf or hard of hearing and can hear other important sounds. This is why subtitles are often used for translation if the audio content is in a foreign language.

As subtitles do not convey all important audio content, they cannot be used as an accessibility feature. Captions will still be required.

## Transcript

A transcript is a text version of a video which includes dialogue, important sounds and any important visual content. They are vital for those who are deaf, hard-of-hearing and also visually impaired as they allow for the use of assistive technologies to be able to convey the content. They also prove beneficial to those who process textual content better than graphic content.

Transcripts must:

* Account for all dialogue within the video
* Include any essential sounds or sound effects such as laughter
* Describe any important visual content from the video

Transcripts and captions go hand in hand, as one can often be used to create the other.

## Live vs Pre-recorded

Pre-recorded time-based media is exempt from the accessibility regulations if it was published before 23rd September 2020, though KCC feels that it is important to bring about this change sooner rather than later and requests all future video content to be made accessibly.

Live media content is exempt from the regulations but will be classed as pre-recorded, and therefore in scope of the regulations, if held on a website for longer than 14 days.

# Forms

Forms are a major part of user interaction on many websites and systems and can be used to offer lots of common functionality such as logins, registration, purchasing and more. As they provide vital functionality, it is imperative that they are made accessible for all users.

## Labels

All form controls require an associated label that provides clear input instruction and purpose to the user. Labels must also be understood by assistive technologies. To ensure this they must be implemented implicitly or explicitly, with the preferred implementation being explicitly.

Explicit labels are bound to form controls by matching the <label> ‘for’ attribute to the form controls ID value. This ensures that the label is directly associated to the form control.

Implicit labelling is implemented by wrapping and containing the label text and form control in the label element. It is useful for situations where the use of an explicit label is not possible.

Visible labels of user interface components must also match the accessible name of the element. This is to ensure that content remains consistent regardless of whether the assistive technologies are used or not.

## Autocomplete

For form inputs that are numerical, or text based, appropriate autocomplete attributes should be defined. This allows browsers to better understand the type of input to be expected, which in turn can allow for more accurate pre-population of the inputs – something that is very important to those with physical or cognitive impairments.

## Errors and Feedback

Users can often make mistakes when using forms, so it is vital that any errors are identified, and that the user is made aware of these issues through textual feedback.

Forms must:

* Clearly identify the error and highlight the input field to the user
* Describe the error that was made, so that the user can rectify the issue
* Provide any suggestion towards fixing the issue, unless there would be a security issue

Providing sufficient error reporting and feedback in forms ensures that it is easier for the user to operate. This is especially useful for those who may have cognitive or other physical impairments.

## Legal and Financial Data

Forms that can cause financial transactions to occur, legal commitments to be made, or enables users to modify and delete data must have precautionary measures to ensure that errors are not made.

For any of the scenarios above, the form must have the ability for the user to double check their inputs and confirm that it is correct before submission. This can be achieved by allowing the user to go back through the form and check over their inputs, and by providing a checkbox that confirms that they have reviewed their inputs before submission.

# Carousels

A carousel (also known as a slider) is a commonly used panel that scrolls through several images. Due to most of its content being hidden at any one time, they are often criticised from a usability and accessibility perspective, but you can include them providing the following:

* Its contents are accessible – images should have alternatives and not contain text, contrast and instructions should be clear etc.
* The user has a control to pause movement
* All controls and navigation can be accessed via a keyboard

These controls make it easier for assistive technologies, those who are distracted by movement and people using keyboard navigation to interact with sliders.

# Maps

Online maps are visual in their nature, but accessible alternatives should be provided where a map is included to provide information.

For example, if you are displaying a map with a pin on it to denote an address, that address should be clearly described in text alongside the map.



# Website Structure - Developer

This section covers the website structure issues that the developers of a website can resolve via editing the code of the website.

W3, responsible for the Web Content Accessibility Guidelines, provide more [detailed developer tutorials](https://www.w3.org/WAI/tutorials/) on some of the topics covered in this guide.

## ARIA Tags

Some mark-up elements are non-standard or non-compatible with screen readers and other assistive technologies. ARIA tags are attributes for an HTML tag which help describe these elements and their information to the user via their assistive technologies.

You should aim to use ARIA as much as possible and particularly when standard HTML is not enough to provide an accessible experience.

See [WebAIM’s ARIA Techniques Page](https://webaim.org/techniques/aria/) for detailed information on implementation.

## Page Structure (HTML)

Structuring each web page’s mark-up in a standard and accessible manner improves the user experience and helps assistive technologies interpret the web page more accurately.

All pages should be given a unique title using the <title> tag in the HTML <header>.

Page regions should be defined using HTML5 tags to ensure that the keyboard tab order is as expected. This includes using <header>, <footer>, <nav>, <main> and <aside> to structure the web page. It is also useful to give each region ARIA labels (aria-labelledby=””), particularly if there are two or more regions of the same type (e.g. a sub-navigation bar).

Within a region, headings should be nested, sequential and ranked to allow assistive technologies to use them as a navigation facility within the page. Use <h1> for the main heading, use <h2> for a sub-heading and <h3> for a sub-sub-heading etc.

Avoid using HTML tables and whitespace for layout as this will be misinterpreted by a screen reader, disorientating some users.

## Skip to Content

When using a website, most users look and interact with the page’s main content as soon as it has loaded. Keyboard users, screen reader users and users with motor disabilities usually have to navigate through navigation and other header elements before reaching the main content which puts them at a disadvantage.

Adding a ‘Skip Navigation’ link as the first focusable element overcomes this, simply linking to the ID of the main content. This is usually hidden until focused on via a keyboard meaning it is not affecting sighted/able users but is significant to improve the user journey of some groups.



## Keyboard Navigation

Many use a keyboard as their only method of interacting with a website. This is universally using the following commands:

* Tab to move to the next element
* Shift+Tab to move to the previous element
* Enter/spacebar to select an element/control
* Arrow keys to move through an element

You should ensure that every element is usable with these keyboard commands, to the same effect as with a mouse.

The order that elements are focused on via the Tab command on a keyboard should also be logical and meaningful.

You can assess the order of keyboard stops using [Microsoft Accessibility Insights](https://accessibilityinsights.io/)’ ‘Tab Stops’ feature, with the following example of correct keyboard order using this tool.



When adapting an element for keyboard use, ensure that a keyboard user can easily enter and move on from that element, avoiding keyboard traps. For example, in a Twitter feed that contains many tweets, use the arrow keys to scroll but Tab to exit the element.

Finally, ensure that navigation and element identification is consistent throughout the website to ensure that all user journeys are consistent and that no users are disorientated by significant but unnecessary changes to layout. This includes appropriately identifying each element and keeping this identification uniform across the page.

## Content on Hover or Focus

When an element is focused on or hovered over, if additional content becomes visible as a result of this focus (e.g. a tooltip):

* The new content should be dismissible
* The new content should not disappear when focused on
* The new content should remain visible until focus is removed/the user dismisses it/the information being shown is no longer valid.

A component receiving focus should not change what the context of the web page is trying to achieve. For example, focusing on a form button should not open a new window or re-arrange what is currently on the page.

If popup content (e.g. a modal window) is activated, it should be announced via the screen reader and focus should go to this content. When tabbing through with a keyboard, the user should not be able to focus on any content behind the popup.

Finally, status messages (e.g. an announcement, cookie policy popup, timeout) should be focused on and announced to assistive technology without first receiving focus. If a timeout does exist, users should be warned prior to it happening, particularly if the timeout could result in data loss.

## Focus Indicators

All user interface components should have a visible focus indicator, when a keyboard is being used to navigate a website. They are important in allowing the user to see what element is currently in focus.

Focus indicators should be of high colour contrast to its surrounding elements, so that it can be clearly seen by the user, even if they have visual impairments. They should follow the minimum contrast requirements for non-text elements, with a ratio of 3:1 against the element they are surrounding.



## Input

Further considerations should be made when developing a website to improve user experience when using touch screens and other input techniques.

* Multi-point and path-based gestures are permitted but all functionality needs to work with a single-point gesture
* Events are not triggered until the pointer is released (not on the down-event)
* Functionality controlled by motion can be disabled with an alternative method of controlling the functionality provided through the user interface.

## No CSS

Some users browse a website without CSS enabled to declutter the site. You should ensure that the website is still meaningful and that elements all function correctly without the use of CSS.

Common issues when CSS is removed include:

* Repeated navigation
* Functionality not working/missing
* Hidden elements appear (e.g. timeout text).



## Language

The primary language of the web page should be defined in the <html> tag of each web page.

If the language is changed on the page, the element containing a different language to the primary language should be declared within the tag for that element. This indicates that a change of language is required to assistive technologies.

## Parsing Errors

Errors with the language used to mark-up the website (such as HTML) can result in assistive technologies not appropriately being able to interpret the element in question. Equally, different browsers may interpret elements differently or be unable to parse it in the first place, resulting in bugs and issues with functionality of some elements. Common parsing errors include:

* Element tags not opened/closed correctly
* Attributes used incorrectly
* Duplicate IDs

Ensure that when developing using a mark-up language, you fully conform to the specifications of that language to eliminate all parsing errors. You can check for parsing errors using automated accessibility tools like [Axe](https://www.deque.com/axe/), and validators such as [W3 Validator](https://validator.w3.org/).

## Flashing

Elements that flash can induce seizures in some user groups due to photosensitivity. Therefore, ensure that nothing on the page flashes to allow these users to view the website without harmful effects.

If an element must flash or refresh, this must not happen more than 3 times in any one second period.

## iFrames

A common way of including content from another web page within a web page is by using an iFrame (inline frame). Due to its partially nested nature, it is vital that suitable information is given to assistive technologies to prevent some users becoming disorientated.

* Provide a frame title
* Enable scrolling within the iFrame - scrolling=”auto”
* Ensure that content within the frame meets accessibility requirements where possible

Do not use frames for presentation purposes.

## Website Design Responsiveness

A responsive website design is critical as there are many variations of screen sizes, resolutions and configurations for how a user views a website.

You must ensure that:

* Content is usable in both a portrait and landscape orientation (unless the orientation is essential to the content)
* Text should be resizable to 200% without any loss of content or functionality
* Unless content is two dimensional (e.g. a table), horizontal scrolling should not be required at a window size of 320x256px (400% zoom on a 1280x1024px monitor).

With text, you should specifically ensure that there is no loss of content/functionality on the web page when one of the following occurs:

* Line height, also known as line spacing, is set to 1.5 times the font size
* Paragraph spacing is set to 2 times the font size
* Letter spacing is set to 0.12 times the font size
* Word spacing is set to 0.16 times the font size.

Having a mobile dedicated website is extremely helpful to the many users that rely on mobile devices to interact with web content.



## Multiple Ways of Access

Every individual web page should have at least two ways of accessing it to improve navigability for a user. These may include:

* Navigation bar
* Site search
* Site indexes/maps

It is also desirable to include a ‘breadcrumb’ navigation to allow a user to return to previous pages in the navigation hierarchy.

# 3rd Party Content

If you use third party content on your website (e.g. plugins, documents) then you should make best endeavors to work with the third party to make it accessible. If you cannot, processes should be in place to provide alternative formats.

All third-party content should be listed on your accessibility statement, including whether you have control over the accessibility of it or not.

# Documents

If downloadable documents are listed on the website, these also must be accessible. Archived documents (those listed purely for reference before 23rd September 2018) do not need to be made accessible, but the document must be made accessible as soon as it is changed/made part of an active process on the website.

Follow [the Document and PDF Guide](https://www.lexdis.org.uk/digital-accessibility/accessibility-tools/accessible-documents/) for information on how to make your documents accessible.

# Further Support

Once you have developed your content to be accessible, you should do your own checks to identify any remaining issues.

* Manual tests with assistive technologies (e.g. navigating your website with a screen reader and keyboard only)
* Automated tests such as Axe and Microsoft Accessibility Insights (these only pick up ~25% of all issues)

On top of WCAG 2.1 AA compliance, Public Sector websites are also required to list a compliant accessibility statement.

While every effort has been made to ensure the accuracy of this document at the time of publication, this is for general information only and is not legal advice.