Supplier Accessibility Guidance

Authors: Natalia Flindall and George Rhodes

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 0.1 | 24/04/19 | Natalia Flindall | Initial Draft |
| 0.2 | 03/05/19 | George Rhodes | Amendments |
| 0.3 | 13/05/19 | Natalia Flindall | Revised version |
| 0.4 | 22/05/19 | George Rhodes | Further amendments |
| 1.0 | 22/05/19 | Natalia Flindall | Live document |
| 2.0 | 24/07/19 | George Rhodes | Update and additions |

[**Introduction**](#_hw9fmwz0ne9p) **2**

[**Image Content**](#_q42wvvz4aga3) **2**

[Image Content Example](#_i36b4rfojnlw) 2

[**Skip to Content**](#_hnk0p1f0d3tp) **3**

[Skip to Content Example](#_jlpmaxijw74) 3

[**Video Content**](#_vfx1upqhzezp) **3**

[**Colour Contrast**](#_2lnr0n4efwm1) **4**

[Colour Contrast Examples](#_xfo1owuhzfli) 4

[**Maps**](#_fguxpb6t6ikc) **5**

[Maps Example](#_yk24y8capi6m) 5

[**Plain Text**](#_skv7c32egqee) **5**

[**Links**](#_gg48catnbg1f) **6**

[Links Example](#_uop9x9ouujdy) 6

[**PDFs**](#_4h87ihl55hmm) **6**

[**Meaningful Sequence and Logical Structure**](#_br4vywcwxbj2) **7**

[**Disclaimer**](#_kql0gnbldg3n) **7**

# 

# Introduction

This document has been created to help suppliers and our responsible staff to better understand the practicalities of making accessibility improvements in line with the [Web Content Accessibility Guidelines (WCAG) 2.1 AA](https://www.w3.org/TR/WCAG21/) standard as part of our responsibilities under the [Public Sector Bodies (Websites and Mobile Applications) Accessibility Regulations 2018.](http://www.legislation.gov.uk/uksi/2018/952/contents/made)

When we as an organisation provide feedback on digital systems as part of our accessibility auditing process we often received frequently asked questions on what specific action need to be taken from a technical perspective. This guide tries to easily explain some of the most common topics we receive questions on. Far more information on these topics is available online.

# Image Content

It is quite important to make sure users can access information on your website regardless of disabilities. Images that have alternative text provide screen readers/text to speech applications the ability to read this out for users who could be partially/visually impaired, and Dyslexic.

## Image Content Example



‘Image of New York’ would not be an appropriate alt tag in this scenario. ‘Aerial view of Central Park in New York City’ would be an appropriate alt tag instead. WebAim have provided what is appropriate on a website including information on images:

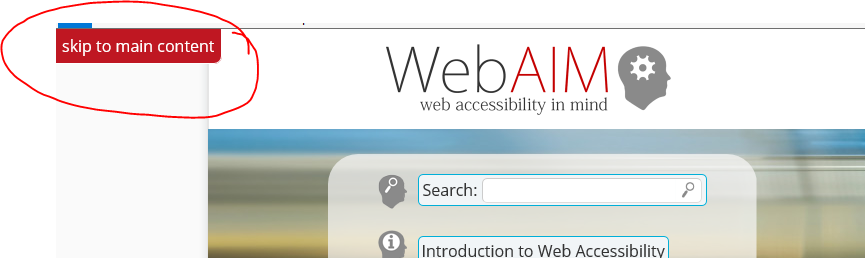
* [WebAIM AltText technique article](https://webaim.org/techniques/alttext/)
* [WebAIM designers resources](https://webaim.org/resources/designers/)

# Skip to Content

A skip to content button needs to be included on all pages on websites to allow users to not have to tab through all navigation menu each time they change page.

The skip to content button should be made visible when tabbed on via keyboard. This allows the user to go straight to the content without going through the navigation menu system, users who have physical injuries or disabilities may find this easier and will more than likely be using keyboard control.

## Skip to Content Example



If your website has many links it is advisable to have a skip to content as users who rely on keyboard will have to go through those links repeatedly before they can access the information needed.

More information about skip to content can be found on the [WebAIM skipnav technique article](https://webaim.org/techniques/skipnav/).

# Video Content

This is a common accessibility issue, videos that do not have captions will make it impossible for screen readers to identify. Providing captions as well as audio for videos allow users who are hearing or visually impaired have access to the video like all users. By providing captions and audio you are giving users a second alternative.

Relying on YouTube’s automated captions is not a complete solution as it does not have a 100% accuracy rate and occasionally misinterprets words. According to the University of Minnesota, YouTube captioning is about 60% accurate on average as they state in the [University of Minnesota YouTube autocaption article](http://www.d.umn.edu/itss/classroom/captioning/youtube_autocap.html).

Also, there is a [video by popular youtubers that highlights poor captioning](https://www.youtube.com/watch?v=hVNrkXM3TTI&list=PLA220BA20D4D3DE46).

# Colour Contrast

Colour contrast issues are another common accessibility issue, in relation to the example below, white text on a bright blue background would fail accessibility standards.

You need to consider whether the colour of text or overlays are the right colour contrast or else users on your platform may have difficulty accessing or easily consuming your information. The reason why this is important because it could cause users additional strain to focus and read text.

The [WebAIM colour contrast checker](https://webaim.org/resources/contrastchecker/) is an excellent tool to use to check if your website has colour contrast issues. Also, W3 have provided information on [what is appropriate colour contrast](http://www.w3.org/TR/2008/NOTE-WCAG20-TECHS-20081211/working-examples/G183/link-contrast.html).

WCAG 2.1 AA Standard requirement is that content has a 4.5:1 contrast and navigation 3:1.

## Colour Contrast Examples

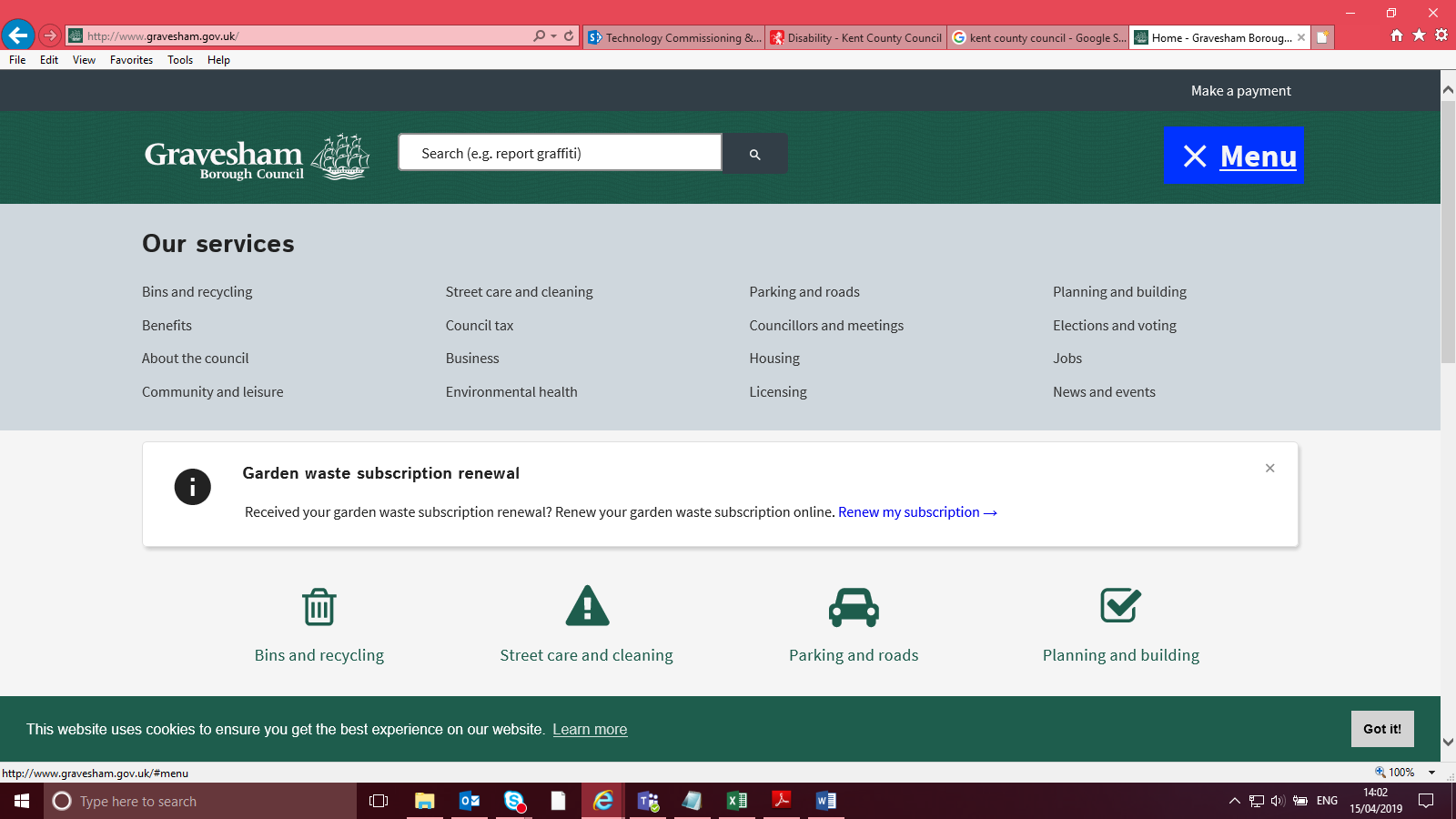


Image 1 of a menu button would not be appropriate colour contrast as bright blue with white text can cause some difficulty in reading including strain to remain focused on the content.

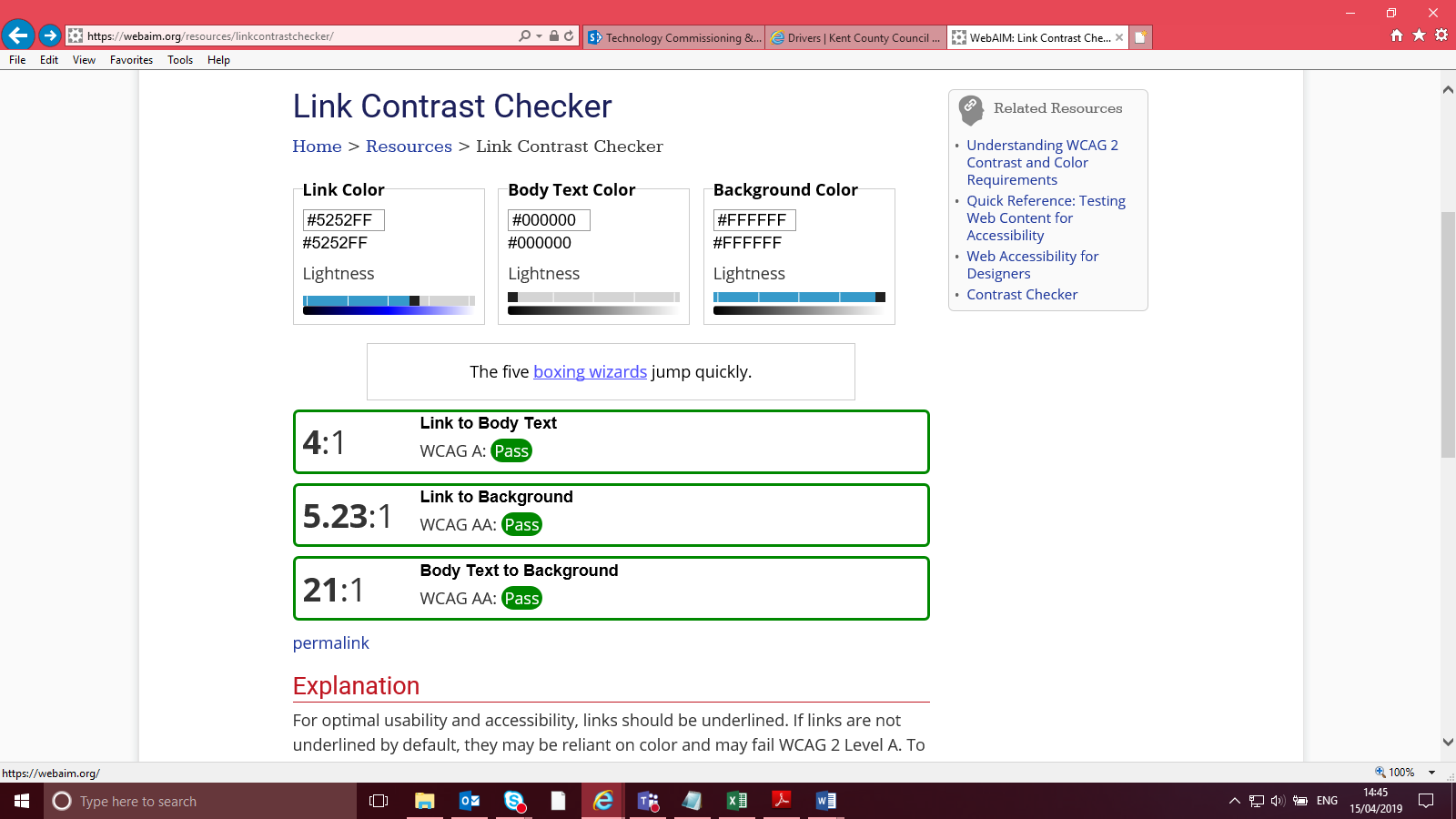


Image 2 of plain text would be an appropriate colour contrast as this is easy to read, and a user would not find it difficult to focus on the text.

a contact us link, the words contact us, are in dark grey while the background is purple. The words are hard to read because the colour contrast is low.

Image 3 of a contact us link in the footer of a website would not be an appropriate colour contrast as the dark grey text on the purple background is difficult to read. The contrast level of this image is 2.56:1.

# Maps

Google Maps and other map applications are a common feature on a lot of websites and are useful. However, it is not accessible for users who rely on keyboard usage or screen readers. If a map is part of an administrative task this would require it to be accessible (which is fundamentally impossible due to the visual nature of maps) or providing an alternate means of completing the administrative task without having to use the map. If it’s a user journey there needs to be an alternative completion route, while an image and alt text solution may be useful for maps that are purely informational in certain situations.

By providing an alternative for users, it brings no digital barriers to accessing information. If this still isn’t possible to do in your situation, provide a suitable contact (Contact number/email) so users can request the information from alternative channels.

## Maps Example

Consider you have a map on your website contact us page that shows users where your drop in offices are. If you do not provide this information in any other format you may be excluding users for contacting you. You could provide your address information in plain text alongside the map on the page so that the important information can still be picked up be a screen reader.

# Plain Text

The standard way of accessing content on a page are using the ‘Tab’ key to access links, forms, dropdowns and buttons. The rest, such as plain text can be navigable by using the arrow keys. In some cases, in the HTML a ‘tab index’ attribute may be used.

An example of using this is as follows: Javascript widgets/menus use <span> and <div> tags and those elements are not natively focusable, so a way of fixing this problem is to set the tab index to 0.

WebAim mentions: “By default, users can only navigate to links, buttons, and form controls with a keyboard. These natively-accessible elements should be used whenever possible. If this is not possible, you may need to use tabindex="0" to ensure an element can receive keyboard focus.”

The links below explain this in more detail:

* [WebAIM Tab Indexing techniques article](https://webaim.org/techniques/keyboard/tabindex)
* [Mozilla Developer keyboard navigable Javascript Widgets article](https://developer.mozilla.org/en-US/docs/Web/Accessibility/Keyboard-navigable_JavaScript_widgets)

# Links

The way in which your links are presented is very important for users navigating with keyboard and screen readers. When tabbing through interactive elements such as links a user will hear what the link text is. If every link on a page says ‘Click Here’, it is not easy for a user to quickly identify links from one another without also being able to scan the plain text content around the link elements.

The key thing to think is that if you are reading the link without reading the plaintext is it still clear where the link is actually going to take you?

## Links Example

The below text is an example of a badly written link:

“We update the reports for this area annually. The latest figures are included in the 2019 sector report. Click here to view the report.”

To improve the above poor example you would want to change the content to link as follows:

“We update the reports for this area annually. The latest figures are included in the 2019 sector report.”

With the improvements we have made to the example above a user will now clearly be able to identify what the link will be taking them to while quickly tabbing through links on a page.

# PDFs

A lot of old PDF’s are known to not be accessible by screen reader. However, there are ways of making this accessible.

These simple resources explain how to make PDF’s accessible in simple steps:

* [Gov.uk guidance on publishing accessible PDFs](https://www.gov.uk/guidance/how-to-publish-on-gov-uk/accessible-pdfs)
* [Adobe help - creating accessible PDFs](https://helpx.adobe.com/acrobat/using/create-verify-pdf-accessibility.html)

If your PDF’s have been created before 23/09/2018 they are exempt from the current legislation and need to be labelled historical documents unless used as part of an active administrative process.

Alternatively, creating a HTML5 document rather than a PDF is now known to be a suitable alternative and in the long term will be easier to find information, use and maintain according to the [GDS blog on publishing in HTML rather than PDFs](https://gds.blog.gov.uk/2018/07/16/why-gov-uk-content-should-be-published-in-html-and-not-pdf/).

# Meaningful Sequence and Logical Structure

One of the most important aspects of accessing content on the website is in what order the user can access the content and if it is in a logical manner (ie. If the content is read out by a screen reader in the same order as a sighted user would reasonably progress through the content).

Meaningful structure is where content has a clear layout and is labelled correctly. A logical structure is the technical term for content being ordered in a consistent manner when reorganised to fit a different sized window or being read out by a screen reader.

For example, if there is a search area/bar, logically the user will go on a search field and then proceed on to the search button and not the other way around. Or if there is a blog post list with each article having an associated picture, it will not read out all pictures before reading out the post headings. It should instead read out all the information associated with one post before moving on to the next.

Those resources are a helpful read to understand this:

* [WCAG section on meaningful sequence](http://www.w3.org/WAI/WCAG21/Understanding/meaningful-sequence.html)
* [Better Testing logical document structure article](http://www.bettertesting.co.uk/content/?p=1619)

A logical structure and meaningful sequence would be content that can be read either left to right or up and down and labelled correctly. The navigation bar should be accessed by keyboard from left to right and the main content should be accessible by keyboard from top to bottom.

Failing to do this will change the context of the content. The images below show examples of no meaningful sequence and logical structure and then how it can be improved:

A screenshot of a cell phone screen with text

Description automatically generated

Figure 1: An example of no meaningful sequence.

The previous image shows an example of no meaningful sequence. Each of the headings for the articles would be read out, followed by each of the content blocks and links. A user that experiences this may have a hard time telling what content is associated with each heading.

The following image shows an example of good meaningful sequence. Each of the articles is read in full before moving onto the next one. By doing this a user will hear a title, then related content, before clearly moving onto a new segment.

A screenshot of a cell phone

Description automatically generated

These images of meaningful sequence have been taken from the [W3 before and after demo website](https://www.w3.org/WAI/demos/bad/after/home.html).

Ensuring buttons, heading and fields are labelled correctly will also ensure that you have a meaningful sequence as this provides context to the user filling out forms and logging in.

# Disclaimer

This disclaimer states that the information and resources supplied found is up to date as of July 2019. We are not to be held responsible over inaccuracy of the information found from 3rd party resources provided in this guide. The sole purpose of this guide is to advise staff and suppliers responsible for the development and accessibility compliance of digital content and not to be used as material for any other purpose.

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. If you require clarification or legal advice of how this impacts your organisation, please consult with your organisations legal adviser.