

ACCESSIBILITY

A WEB

FOR ALL

Ashley Firth gives the lowdown on how to design and develop your websites to be truly inclusive

6 SIMPLE WAYS TO ACCESSIBILITY

Some of the key areas to explore



“

WE'RE ALL ONLY TEMPORARILY NOT DISABLED – BY CONDITIONS OR CONTEXT. PLUS ACCESSIBLE DESIGNS ARE BETTER FOR EVERYONE. SO CREATE INCLUSIVE SERVICES FOR: YOU, THOSE YOU LOVE AND ALL USERS, NOW AND IN FUTURE. IT'S RIGHT, SENSIBLE AND COST-EFFECTIVE

James Buller
Access needs team – Home Office Digital

ARIA ROLES

Accessibility starts with markup. ARIA tags cover everything from annotating a page to identifying states and properties of elements, and help users with screen readers understand everything about your web page. The 'markup' section (pg. 76) covers all of this, and how to both show and hide content for this platform.

IMAGE & SVG ACCESSIBILITY

Imagery can be one of the biggest barriers to user interaction. Learn how screen readers interpret imagery and, with the advent of SVGs, find out the different ways you can make your graphics accessible, and the potential pitfalls of using icons to infer meaning.

The World Health Organisation (WHO) first published its formal definitions of disabilities in 1980, but updated them at the turn of the 21st century to not simply refer to a disability as a “personal attribute”, but to be “context dependent... reflecting the interaction between features of a person’s body and features of the society in which he or she lives.”

The WHO’s point was that disability happens during interactions between a person and society on a physical and cognitive level, and the web is a huge part of that. There are mismatches between how a user interacts with the web, or even their computer, that differs from that of the people that made it.

These conflicts can prevent a person from engaging, and sometimes even interacting, with the content of a site.

It is the job of designers and developers to factor in these cases and create inclusive web experiences that work for the largest number of people possible.

The W3C has a comprehensive list of requirements that can be completed to achieve web accessibility, be that at AA or the stricter AAA level. However these are not enforced and, as a result, often overlooked, and the last full version of the WCAG (Web Content Accessibility Guidelines) was released nearly a decade ago. However, there are in fact many simple ways to make your sites more accessible, and ensure that the most people possible can enjoy your content.

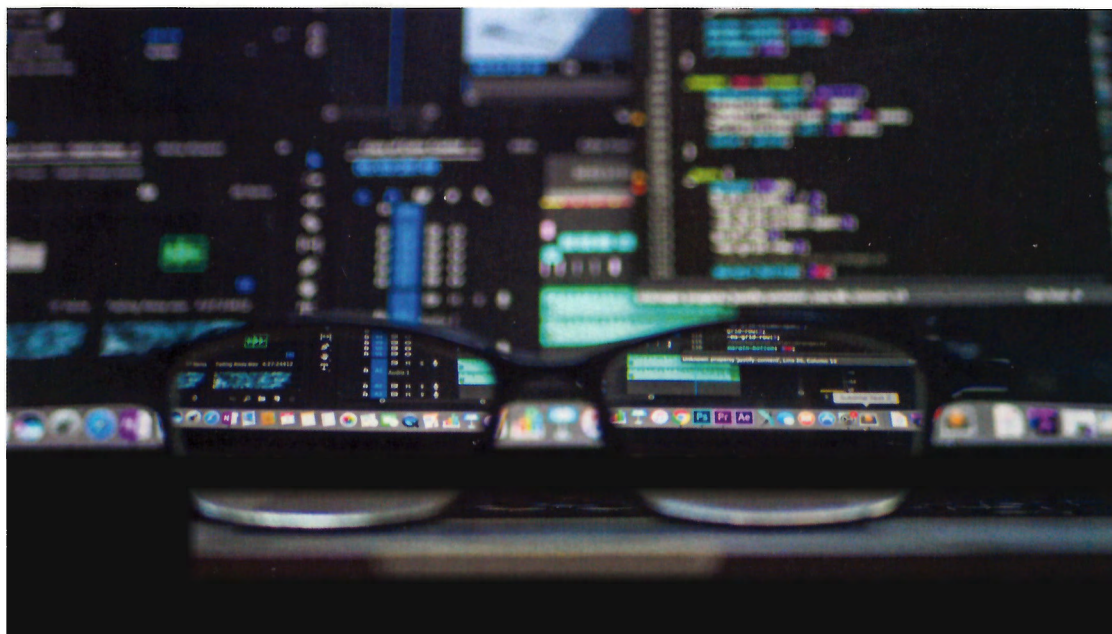
Last year the Home Office released some posters offering guidance for designers and developers on how to make sites accessible to everyone.

They outlined rules to help those with low vision, deafness, dyslexia, anxiety, users on the autism spectrum, or those using screen readers.

We will look over some of the basic changes you can make in markup, styling, and imagery to improve your site’s accessibility, and then use these pointers to tackle some of the design and development challenges associated with specific disabilities.

My own approach to accessibility is with Octopus Energy. As an energy supplier, we need to cater to a very wide range of needs. Everybody needs energy and it’s a basic human right. However in this industry it’s dominated by large suppliers using legacy systems and generally poor support for accessibility. There will be some improvements and advancements we’ve made in this article as practical examples.

“
THE W3C HAS A
COMPREHENSIVE LIST
OF REQUIREMENTS
THAT CAN BE
COMPLETED TO
ACHIEVE WEB
ACCESSIBILITY, BE
THAT AT AA OR THE
STRICTER AAA LEVEL



CONTRAST LEVELS

This is an area that affects not just users with low vision, but those with dyslexia and autism. You may be pleased with your design, but is the colour scheme preventing some users from interacting with your content? It’s easily tested for, and there are ways to turn this issue into a positive addition to your website.

SKIP TO MAIN CONTENT

As part of making your website accessible to keyboard-only users and screen readers, learn how to implement a small ‘skip to main content’ button to save your users time and unnecessary stress. It requires only a few simple lines of markup and styling to achieve, and you can see how in the ‘styling & imagery’ section (pg. 77).

COLOUR BLINDNESS

Colour blindness can encompass such a wide range of things visually, meaning that it’s very likely that users are seeing your site aesthetically from thousands of slightly differing perspectives. In the ‘low vision’ section (pg. 79) we talk about the different types, how to test pages under those circumstances, and how to plan for them.

SUBTITLES & CAPTIONING

Videos are a great way to provide information to your users in an engaging format, however without subtitles or captioning, they can alienate users who are deaf and even those who don’t speak the language in your videos. Check the ‘deaf/hard of hearing’ section (pg. 80) to learn more about subtitles and captions.

BETTER MARKUP

The easiest first step towards an accessible site



HEADING STRUCTURES

All headers in a web page should have a hierarchy, beginning with the most important header to the least, and the markup of the page should reflect that. In HTML, the most important title is an `<h1>` tag, and can go all the way down to an `<h6>`. You can use as many or as few of these as you like, but they must start with an `<h1>` and must always be used in order. The reason for this is that heading tags denote importance to screen readers, and out of order headers may well confuse a user. Proper heading structure also benefits the SEO of your page.

You can check this using the HeadingsMap extension for Chrome. When run on a page, this generates a tree based on the headings on your page, and highlights any that are out of place or have been skipped entirely. **Tip:** To dispel a common myth, having multiple `<h1>` tags in a page is no longer a huge taboo. Google itself states that you can have up to three `<h1>` tags in a page that are used to garner its content. It's important that those `<h1>` tags remain on the same hierarchical level as each other, and that as you make your way down the heading ladder, no rung is skipped or added out of place.

HTML5 OUTLINE

HTML5 Outline is the concept that – rather than being restricted to a cascading style when it comes to headers – the structure of a document

could be split into sections through markup and that each of these sections could have their own heading hierarchy. The user agent could then more easily apply this information to generate a table of contents which could then be used by assistive technology to help the user navigate.

For example this would be the current approach with headings always cascading:

```
<div class="section"
  id="brazil-weather" >
  <h1>Brazil weather</h1>
  <p>...</p>
  <div class="subsection"
    id="brazil-landmarks">
    <h2>Landmarks</h2>
    <p>...</p>
  </div>
</div>
```

You could have a system like this:

```
<section>
  <h1>Brazil weather</h1>
  <section>
    <h1>Landmarks</h1>
    <p>...</p>
  </section>
</section>
```

The key here would be to make sure that each `<section>` tag is easily identified, and has an appropriate header, and that no parts of the page are "marooned" from a screen reader's perspective without context. Although this feature has been in the W3C spec since 2008, it has yet to make it beyond a theoretical stage. However, it's possible that it could make the whole page – not just its headers – more digestible for screen readers.

OFF-SCREEN HEADERS

Whilst auditing your site, you may find that certain sections lack context as they don't have – and in many cases visually need – a header. A common example of this would be navigation. With this example you can tackle it with the 'navigation' aria-role or using the `<nav>` tag, but what about less obvious sections?

You can create a header with a small class that visually hides it, but will show up on a screen reader like so:

```
<h1 class="screenreader-
visible">This is my section</
h1>
```

```
.screenreader-visible {
  position: absolute;
  width: 1px;
  height: 1px;
  padding: 0;
  margin: -1px;
  overflow: hidden;
  clip: rect(0,0,0,0);
  border: 0;
}
```

Tip: Bootstrap has utility classes built in that can include content for screen readers only and enables only those assistive devices to focus on it, see `.sr-only` and `.sr-only-focusable` in the latest version.

ARIA ROLES

Accessible Rich Internet Applications (ARIA) defines a way to make web content and web applications more accessible to people with disabilities.

ARIA Landmark Roles help assistive device users navigate your site, enabling them to tell the difference between different sections. There is a full list on the W3C website, but these are the important ones to remember since they cover the core area of a typical page:

banner – Typically the header of your web page that includes the name of the website

search – For the search form

main – This would designate the main content area on your site

navigation – Use on any navigation list, typically on the nav element

contentinfo – Typically the footer of your web page

To add an ARIA role, you simply add the 'role' attribute to your element with one of the attributes.

```
<header role="banner"
  class="site-header">
```

As well as the above landmark roles, there are also roles relating to 'widgets' (button, slider), composite roles (menubar, tablist), and document structure roles (region, toolbar).

States and properties

As well as properly marking the sections and roles of a page, there are also ARIA tags related to state and properties, enabling you to provide context as an action is being carried out. For example, you can inform people with screen readers of certain visual functionality such as an autocomplete, if a radio button is checked or not, or even the min, max, and current values of components like sliders. There are even ARIA state roles for drag and drop functionality.

It's worth taking some time to look over the full list (<http://bit.ly/2EB2JJb>) to acquaint yourself with them, as it is one of the biggest accessibility gains you can make on a website.

Tip: HTML5 implicit mapping In HTML5, several of the landmark roles are implicitly defined within certain tags to make things easier.

main = `<main>`

navigation = `<nav>`

contentinfo = `<footer>`

article = `<article>`

complementary = `<aside>`

region = `<section>`

Aria hidden

Sometimes it can be just as important to hide content as it is to label and annotate it. The 'aria-hidden' property is really handy for hiding components that are on a page for purely visual purposes, such as icons, so people who are using screen readers don't have to spend time listening to them needlessly.

```
<p aria-hidden="true">
  This content is hidden.
</p>
```

STYLING & IMAGERY

Making a site better for eyes and ears

Focus styles

There are default browser styles for focused elements, but it's fairly common for developers to apply blanket rules to these selectors due to the less than preferable blue line it shows when a user clicks them. However, these are really important. For example, try navigating a web page using just the keyboard and you'll see how important focus styles are to make you aware of where you are on a page.

Since they're seen when a user clicks the element as well as when they're focused via keyboard, focus styles need to tread the line between being obvious and not negatively impacting a page's design. Nice alternatives to the typical outline rule can be box shadows or changes in background and border colour.

Skip to main content

A massive gain you can make in accessibility primarily through styling is the 'skip to main content' link. As most pages on a site have a common navigation or header at the start of every page, it can leave users on screen readers listening to the same content as they navigate around your site, or having to skip through elements and guess where each page's unique content starts. Instead, you can make a simple link to take them right to it in two easy steps:

1. Add the markup for the link as the first piece of content on the page

semantically, and have the link point to a '<div>' containing your page's main content:

```
<a class="skip-to-content"
href="#content">Skip to main
content</a>
...
<div id="content">
...
</div>
```

2. Use CSS similar to the 'screenreader-visible' class we mentioned in the 'markup' section to effectively hide the link from the normal page style:

```
.skip-to-content {
position: absolute;
top: -400px;
}
```

As screen readers are only interested in the content of the page, this will mean that, as the first item on the page, the user will be immediately presented with the option to use the link to skip down to the main content.

This feature can be useful for users navigating your site using just a keyboard with a small addition, and it saves them a lot of tabbing! Add some styles to your link when it's focused to undo the absolute positioning, and it will appear as the user starts tabbing through the page's content:

```
.skip-to-content:focus {
position: relative;
top: 0;
}
```

Base font sizes

Raising the base font size of your site will really help those with low vision. The gov.uk site has 19px as the lowest font size, ensuring everyone can easily read its content, and prevent the need for them to zoom in on content. Make sure you don't have large text in paragraph tags masquerading as headers. The visual meaning attributed to it by styles will never be conveyed via a screen reader.

No text in imagery

The main problem with using an

image to show text is it takes away the accessibility for blind and visually impaired users. For instance, a blind user might have a screen reader set up on their computer to read out the text and menu items for them, but it can't read text that's contained in an image unless specified in the 'alt' tag. It simply reads any text it comes across on the screen.

It is left to the 'alt' tags to describe not only the image contents itself, but the text within the image, and so you should try and keep the two separate as much as possible.



ALT TAGS

Ever questioned the importance of 'alt' tags? Have them read out to you at speed by a screen reader and you'll be persuaded. The only real rule here is to ensure that the text used actually describes what is in the image, rather than the image's purpose within the page. For example, the 'alt text' for an image of a tree used as an "About Us" page banner should be "an image of a tree," rather than "About us banner image."

CSS LAYOUTS

CSS makes it possible to create intricate layouts and designs in order to make a web page more attractive. But, taking this idea too far can end up excluding certain users. You need to avoid the assumption that it's obvious that two pieces of content are linked because they are close to one another visually. A good example is creating a layout that isn't ambiguous and is capable of adapting to all screen sizes.

ICONOGRAPHY

Icons are a neat way to display a status or action without the need for too much text. However, sometimes an icon is obvious to you but perhaps not to others. If it's a custom icon it could become open to interpretation and misunderstanding, meaning that your attempt to create something cleaner and less cluttered actually results in confusion. It's important that you ground icons that are not universally used in context.

“

THE GOV.UK SITE HAS 19PX AS THE LOWEST FONT SIZE, ENSURING EVERYONE CAN EASILY READ ITS CONTENT

AUTISM

COMPLEX LAYOUTS

Autistic people often have heightened sensory awareness, and so can find busy pages overwhelming. It is particularly important to avoid complex layouts and 'walls of text'. Look at condensing paragraphs into a series of bullet points, or providing a small 'overview' section for longer prose.

MOBILE-FIRST APPROACH

This idea of logic and simplicity in layout and content comes up in relation to a few different disabilities, and realistically almost all users. Visitors don't want to be overwhelmed. A mobile-first approach is good here, and a good thing for accessibility in general. Learn to make the simplest layout due to screen restrictions and then working up is a good mentality for web in general.

DESCRIPTIVE BUTTONS

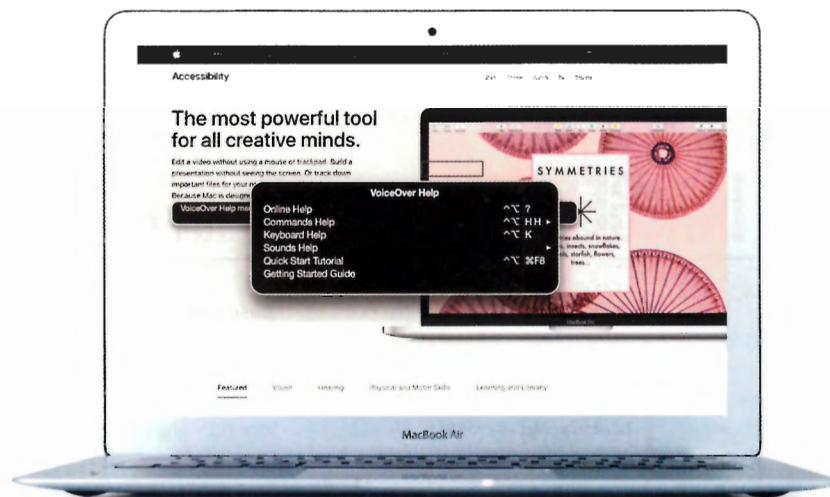
Phrases like 'click here' and 'get started' may seem obvious but are ambiguous and say very little independently. Whereas links such as 'attach files' or 'subscribe to feed' are clear, concise, and unambiguous.

FIGURES OF SPEECH

You should always attempt to write in plain English instead of using figures of speech or idioms. They can create ambiguity and lead to a complete misunderstanding of the point that you're trying to make.

SELF-CONTAINED ACTIONS

Regardless of the context within which you display a button/checkbox/radio button on a page, you should look at each action you ask a user to perform as an independent function; does it make sense by itself?



SVG ACCESSIBILITY & SCREEN READERS

How to make the new wave of imagery work with audible software

Screen readers have become much more prominent in the last decade, and where previously third-party software was a necessity to enable blind and visually impaired users to operate a computer, there is now an in-built system for every major operating system.

Not only is this great for users requiring the software, it's also made it universally accessible to developers; enabling them to hear a site they may have only ever seen before.

They're made more powerful by the upgrades we've covered already on the previous pages, but it's still important to consider accessibility when new technology is introduced. A great example of this is the surging popularity of scalable vector graphics (SVG) - here are a few approaches to make sure they can be heard.

Alt & Role attribute

If you are displaying SVGs through the '' tag, it's important to have both the 'alt' attribute describing the asset, but also an 'aria-role' tag of 'img' to make the asset's purpose clear. There have been issues with VoiceOver on Safari where it will skip the image entirely, even if it has an 'alt' attribute, if its role is missing.

```

```

aria-labelledby and aria-describedby

The better way to give your SVGs the same level of accessibility as images is to have them inline in your markup, and provide a 'title' and 'desc' tag just inside of the main 'svg' tag. You can then match up the 'labelledby' and

'describedby' tags with the IDs of these elements like so:

```
<svg aria-labelledby="logo-title" aria-describedby="logo-desc" role="img">
  <title id="logo-title">Apple logo</title>
  <desc id="logo-desc">A silhouette of an apple with a bite taken out of it</desc>
  ...
</svg>
```

This enables screen readers to understand the SVG much like an 'alt' tag would.

Hidden element

A common and recommended approach when it comes to browser support and consistency across screen readers is to add a screen-reader-only element next to the 'svg'. It's a little bit more involved in comparison to the others, but tests have proven that all browser and screen reader combinations are able to announce the link with the expected text.

```
<div>
  <svg class="icon" aria-hidden="true">
    ...
  </svg>
  <span class="visuallyhidden">Close</span>
</div>
```

If you choose this approach, it's also worth adding the 'aria-hidden="true"' attribute to the SVG itself. As covered in the 'markup' section, we don't want screen readers to try and interpret the asset itself if we provide it with a separate description instead.

FINDING A SCREEN READER

Mac - VoiceOver

Built into macOS, iOS, tvOS and watchOS, VoiceOver is an excellent screen reader that provides audible assistance for blind and visually impaired users across the entire operating system.

Google - TalkBack

Google TalkBack is an accessibility service for Android that helps blind and visually impaired users to interact with their devices. TalkBack handles user interactions with the entire OS and not just the web, but is fully capable of acting as a screen reader inside the default Chrome app or any other downloadable browser from the Google Play store.

Windows - Narrator

Narrator has been part of the Windows OS in some form since Windows 2000. It was originally designed as a way to help users install a fully-featured screen reader. Now in Windows 10 it is available in twelve languages and is also included on all Windows phones. There is also a popular screen reader for Windows called JAWS (Job Access With Speech) that can be installed to read the screen with either text-to-speech like Narrator, or a refreshable Braille display.

Linux - LSR & Orca

There are two popular Linux-based screen readers that are based on open source. LSR was started in 2006 by IBM and uses speech, Braille and screen magnification to aid users. Orca is GNOME's default screen reader.

LOW VISION

Understanding the different conditions and how to design for them

Contrast levels

This is a tricky issue to spot sometimes, but easy to test. W3C states that the contrast level between foreground and background colours should be at least 7:1 for regular text to be AAA compliant, and 4.5:1 for AA compliance. This is especially important for users with low vision or certain types of colour blindness, who may have trouble focusing on text coloured similarly to the background.

The trick here is walking the line between creating the design and style that you want aesthetically, whilst not compromising the content or experience for those with visual difficulties.

One approach that is quite involved, but can have brilliant results is allowing for customisation within your design. This way you don't end up alienating your users for the sake of the design.

We tackled this problem at Octopus Energy. The typical design style for us is white text on a dark blue background. Although this passes AAA standards for contrast ratio, customers with visual impairments – and even some with dyslexia – reported problems being able to read or focus on the content. The solution we implemented was to enable each customer to invert the colours of the communications they received, meaning they could choose an alternative option of dark text on a light background;

The contrast levels between the two are essentially identical, but there was a barrier with the former that there wasn't with the latter. More importantly, it empowered users to customise their experience with us and helped them engage with the communications more freely.

COLOUR BLINDNESS

When using colour to convey meaning or action, it is important to

always support it with text and sometimes shapes. A reliance on colour can create a hugely confusing experience for colour-blind users, and sometimes result in a complete loss of contextual meaning

One added complexity is ensuring that the colour scheme you're using works for the multiple types of colour blindness that a user could have:

Red-green confusion

This is the most commonly known type of colour blindness. However within spectrum there are several types of red-green confusion:

- **Deuteranopia & Deuteranomaly (also known as green-blind)**

- Reduced sensitivity to red light, and a missing or reduced wavelength cone for seeing green hues.

- **Protanopia & Protanomaly**

- This is where the 'L-cones' in the eye are either missing or defective, meaning that the user cannot see reds. The challenge here is that red and green are the most commonly used colours to infer positive or

negative messages. This makes it more important to remember to ground that status with a message and/or icon.

However it's not the only colour combination that causes issues with colour-blind users:

Yellow-blue confusion

- **Tritanopia or Tritanomaly**

- Users have normal red and green vision, but have trouble distinguishing yellow and red from pink, and blues appear greener.

A very famous example of this was the white & gold / blue & black dress argument that gripped the internet a couple of years ago. The ramifications of this colour confusion aren't as dramatic on websites as red-green, but it is worth keeping in mind, particularly for things like graphs that use colour to associate segments with a legend of names. As always, it's worth testing your pages just in case!

Monochromacy

This is the most severe form of colour blindness, and prevents the person from seeing anything but black, white, and shades of grey.

Colour blindness can encompass such a wide range of things visually, it's very likely that users are seeing your site from thousands of slightly differing perspectives. The reality is you won't be able to create a colour scheme that can be fully interpreted, which means that you shouldn't rely on colour to convey meaning.

TESTING YOUR WORK

This may seem daunting, but there are some great online tools for you to check your work against quickly and easily that cover every contrast ratio and major strain of colour blindness mentioned. Check out the tools section for more information



DYSLEXIA

USE IMAGES & DIAGRAMS

Try and use images and diagrams to support text you're displaying, rather than using blocks of heavy text. Not only does it look more appealing to engage with, but it makes it more manageable to those with dyslexia.

DON'T REQUIRE ACCURATE SPELLING

You shouldn't demand accurate spellings for certain data. Components such as autocorrects (that we will cover later) allow for spelling errors and offer suggestions based on the text that the user has input.

PROVIDE REMINDERS

Rather than forcing users to remember things from previous pages, you can provide reminders or suggestions to alleviate the need to use short-term memory.

CONSIDER OTHER FORMATS

People with dyslexia often find information easier to learn or digest when it's provided in different ways – consider producing materials in other formats such as audio or video to add variety.

ALLOW FOR A CONTRAST CHANGE

Some dyslexic users have trouble reading on certain foreground/background colour combinations (such as white text on a black background) even though it has a high contrast ratio. Consider offering the ability to change it.

DEAF/HARD OF HEARING

Make sure that those who can't hear aren't missing out

SUBTITLES/CAPTIONS

It's worth noting that captions and subtitles are not the same thing: Captions are intended for deaf and hard-of-hearing audiences. The assumed audience for subtitling is hearing people who do not understand the language of dialogue. Also captions move around a screen to denote who is speaking, whereas subtitles are almost always set to the bottom centre of the screen.

They are however implemented in the same way technically, and so whichever challenge relates to your site, here's how to tackle it:

The <track> element

HTML5 enables us to specify subtitles for a video using the '<track>' element. This lets us specify the type of content that we're adding, the language it's in, and a reference to the text file that contains the subtitle information.

```
<video controls
  preload="metadata">
  <source src="videos/video.
  mp4" type="video/mp4">
  <source src="videos/video.
  webm" type="video/webm">
  <track label="English"
  kind="subtitles" srclang="en"
  src="captions/vtt/video-en.vtt"
  default>
  <track label="Deutsch"
  kind="subtitles" srclang="de"
  src="captions/vtt/video-de.vtt">
</video>
```

You'll see in the example above that can specify the source, track type and source language - the default is English.

VTT files

The files that contain the actual subtitle data are simple text files that follow a specified format. The format is Web Video Text Tracks (WebVTT), the specification is still being worked on, but major parts of it are stable and ready to be used today. They are simple text-style files, in the following format:

WEBVTT

```
00:01.000 --> 00:04.000
- This is the common sloth in
its natural habit
00:05.000 --> 00:09.000
```

- It moves very slowly along a branch
- in search of food

You simply specify the start and end time for the part of the video you want to annotate and then add the text within that time frame. You can also place this text over multiple lines if you want.

Taking it further

In addition to what we've covered, you could create your own 'closed caption' button to allow toggling between showing and hiding subtitles/captions. We don't have space to cover this here but implementation is straightforward and a quick online search will find some instructions.

VTT files also allow for 'voice spans' to distinguish certain subtitles. For example, the following markup:

```
00:00:00.000 --> 00:00:12.000
<v Test>[Test]</v>
would allow you to target that
particular caption in CSS like this:
::cue(v[voice="Test"]) {
  color: white;
  background: red;
}
```

FORMS OF COMMUNICATION

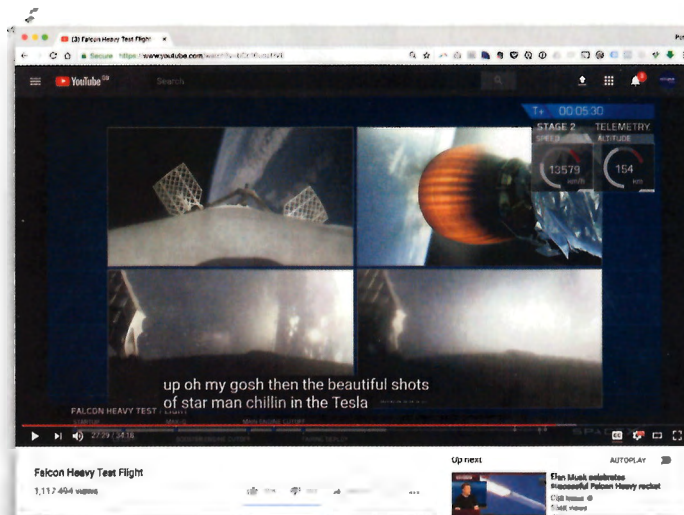
Placing a phone number on your website can be useful to some, but restricting to others.

Therefore you should ensure that you offer multiple means of communication so as not to exclude a certain demographic - it's good for accessibility and for your business!

Your customers should be able to contact you in a mode of communication that works for them. Email is the natural alternative to the phone, since it doesn't require an audible interaction. However some people still want the instant responses that often comes with phoning. In this case live chat, or even Facebook Messenger integration provides the best of both worlds. We offer all of these forms of communication at Octopus Energy, and have found that it not only enables us to communicate effectively with all of our customers, but the freedom often results in them taking the time to provide more feedback on how we're doing in general, which is invaluable.

“

CAPTIONS MOVE AROUND A SCREEN TO DENOTE WHO IS SPEAKING, WHEREAS SUBTITLES ARE ALMOST ALWAYS SET TO THE BOTTOM CENTRE OF THE SCREEN



MOTOR DISABILITIES

SHORT TIMEOUTS

Commonly seen on eCommerce sites, and becoming more of a problem as they become more popular. For users that haven't had a lot of practice with journeys - checkout processes, for example, or who like to take time to read content and check input data - short timeouts encourage hastily completed actions and invite human error. The end result is frustration.

LARGE HIT AREAS

This is a design pattern that would help the web as a whole. Making clickable actions on a web page large rather than demanding precision means that you can avoid more of the frustration mentioned above by allowing for a much larger margin of error. This is especially important on mobile devices.

PROVIDE SHORTCUTS

Designers and devs should look at a page and work out if there are areas that could be simplified. It can improve conversion.

A good example of this is converting a series of manual address inputs into a postcode lookup followed by the user selecting their one from a list.

TAB INDEX

Many users prefer a keyboard to navigate. Most browsers are very good at allowing keyboard interaction. However, sometimes there are a series of elements that would be easier to understand in an order different to how they're displayed visually.

The 'tabindex' attribute specifies the tab order of an element (when the Tab button is used for navigating). A value of '1' keeps it in sequential order, and anything higher works on a priority system with the highest number being focused first.

7 TOOLS FOR BETTER ACCESSIBILITY

Finding problems are sometimes just as hard as fixing them. This collection offers some of the most important tools out there



SIM DALTONISM

bit.ly/1JHRCIJ

If you're using a Mac or iOS, this great app by Michael Fortin enables you to overlay a window directly over any web page or application you're viewing and see a live preview of what it looks like which each form of colour blindness. You can resize it to any size and you aren't restricted like you are with the web page. It also has an iOS version where you can use the device's camera.



A11Y CHECKLIST

a11yproject.com

The A11Y project work tirelessly on providing clear advice and tips on web accessibility. It contains its own list of resources, an accessible widget and pattern library, and is worth visiting for its blog on new approaches.



LIGHTHOUSE

bit.ly/2jnLfpG

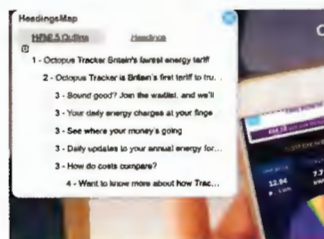
Lighthouse is a tool that audits the accessibility performance, best practices, and PWA standards of any web page. It's built into Chrome's audit panel in the web inspector, is brilliantly detailed, and can be run instantly.



CONTRAST RATIO

leaverou.github.io/contrast-ratio/

Want to test a combination of two colours before using them in a design? Lea Verou has made a neat online checker that will show you an example of how it looks, and displays the contrast ratio level and what level it passes.



HEADINGSMAP

bit.ly/2ER31Nf

The HeadingsMap extension generates a tree, based on the headings on a page and highlights any that are out of place in terms of hierarchy, or have been skipped entirely. It's important for both screen readers and SEO.



VALIDITY

bit.ly/2BLqlx3

A nice and easy Chrome extension to perform inline HTML validation checks on your pages, with the results being outputted to the browser console (yellow for warning, red for error). This extension is also available in Firefox.



VOICEOVER

apple.co/2rUkTl

The most comprehensive of the screen reading software mentioned, VoiceOver is built into every major Apple operating system and gives you a great insight into your web page's performance for those with blindness or low vision.