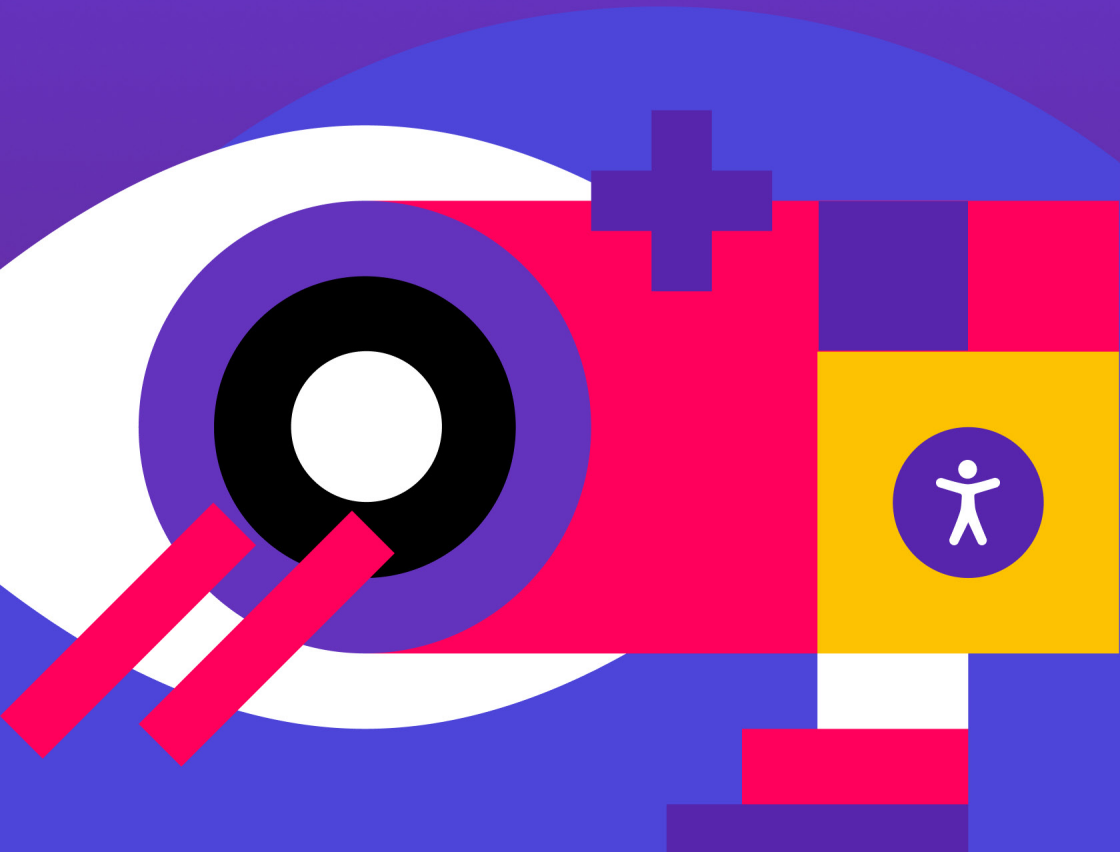


Best practices for implementing great web accessibility

SIMON WATERS





About the author

Before joining Silktide as Head of Marketing, Simon knew very little about accessibility. Over the last couple of years, he's gained a working knowledge of the key principles, and so wrote this book to help other beginners gain an overview of the importance of accessibility.

Table of Contents

What is web accessibility?	05
How does poor web accessibility affect people?	06
Considering situational accessibility.....	07
Accessibility is a scale, not a switch.....	09
The ambiguity in WCAG.....	11
How effective is automated web accessibility testing?.....	13
Best practices for great web accessibility	15
Keyboard navigation and assistive technologies.....	16
Be consistent in page layout and semantics.....	20

Add alternative text to images (where appropriate).....	21
What is alt text for?.....	22
Ensure subtitles, captions, transcripts and audio descriptions are present.....	31
Do not automatically play media without allowing users to stop them.....	35
Begin with the Design.....	39
Color contrast of text and its background.....	42
Page headings and semantic structure.....	43
Form styling and labeling.....	44
Use descriptive and unique URLs.....	55
Minimize the use of PDFs or, if unavoidable, make sure they're accessible.....	58
Check the readability.....	60

The ROI of Web Accessibility.....63

Summary.....	63
Prevalence.....	64
Business case.....	65
Costs and solutions.....	71

Conclusion.....73

Accessibility and the law.....	73
Finding and fixing web accessibility issues.....	74
Next steps.....	75
About Silktime.....	77





Chapter 1

What is web accessibility?

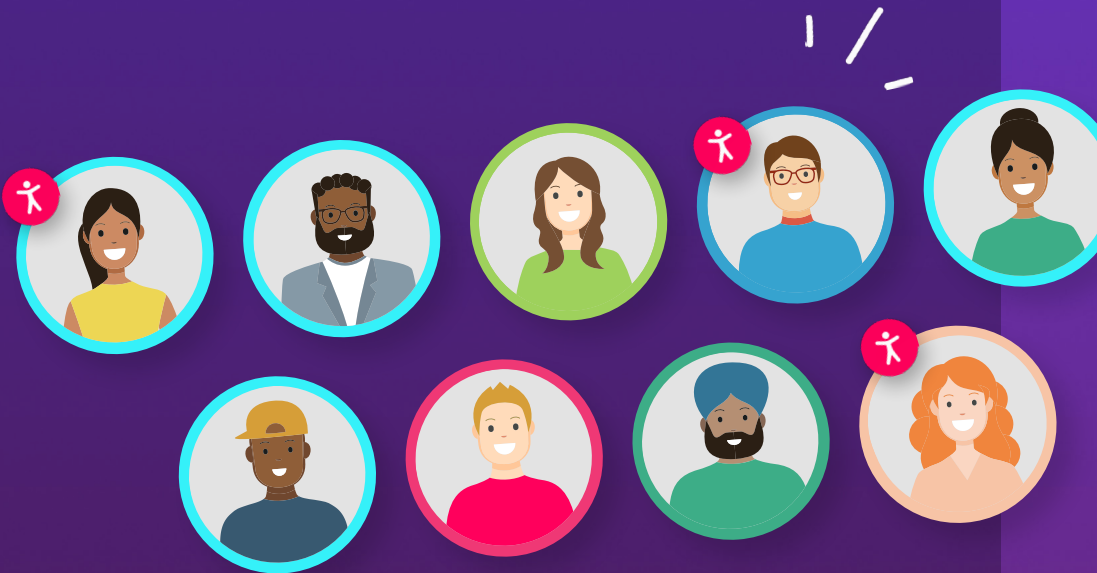
What is web accessibility?



Web accessibility is the process of ensuring that your website caters to a wide range of users' needs. It's not always about disability, although allowing people with disabilities to use your website is the first thing most people think of.

Accessibility is often reduced to checklists, compliance, and legislation but it's really about people and improving their online experience.

This book aims to help beginners understand how they can improve the accessibility of their websites. In the majority of websites that Silktide tests we see the same common accessibility issues. In this book, we'll aim to let you know what these issues are, how you can identify them, and how you can fix them.





How does poor web accessibility affect people?

Let's say your user is blind or has impaired vision. They use a screen reader, which is a piece of technology that reads the content on their screen out to them. They can browse your website, fill in forms, and generally accomplish everything a sighted user can.

Unless of course, your website isn't accessible, then it becomes difficult or completely impossible.

For example, a screen reader will read out everything on your page, one word at a time. That mega menu you have at the top of your site might take a couple of minutes of painstaking work for a blind user to step through. Because this is such a common problem, accessibility standards require you to add an invisible "skip to content" link, which allows screen reader users to jump

past the repeated parts of every page. It's a small thing, but it can make a world of difference.

What if you hide critical information inside images, and not in text? Or you include a form with no labels, so there's no way to know where to put your name, address, or phone number?

Or how about intrusive popup banners, modals, or chatbots that steal focus from screen readers?

These are common accessibility issues that are relatively easily fixed.

For an overview of how accessibility affects people with disabilities, with real-world examples, we'd recommend watching Silktide's short [webinar](#) on the topic.



What is a screen reader?

A screen reader is computer software designed to help people with little or no vision by reading device content aloud. It makes web pages, icons, forms, dialogue boxes, documents, and many other elements, accessible.

Considering situational accessibility

Let's consider "situational accessibility". If a user has low vision they need text contrast to be high enough that it's readable.

The same applies to a person with excellent vision who happens to be using their device in bright sunlight.

A user might not be able to hear the audio in a piece of your content because they are either deaf, have an ear infection, or are in a loud room. [Some people may prefer content to be subtitled.](#) These situations are all fixed with the same solution – adding subtitles, captions, and audio descriptions to visual media.

Indeed, many accessibility issues affect all users to some degree, and often a single solution improves the experience for everyone. Accessibility, then, goes hand in hand with usability.



Permanent

Temporary

Situational

Touch



One arm



Arm injury



New parent

See



Blind

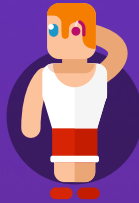


Cataract



Distracted driver

Hear



Deaf

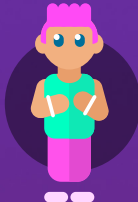


Ear infection



Bartender

Speak



Non-verbal



Laryngitis



Heavy accent

Accessibility is a scale, not a switch.



Silktide is an automated accessibility testing platform, and our technology is used to power the [Index](#), the global web accessibility league table.

We test and then grade website accessibility using our proprietary scoring system.

People often ask us whether there is a score or a threshold they can attain which means their website is accessible. They also ask at what point they can become 100% accessible.

However, it's not as simple as that.

There is some misunderstanding. But this is ok because it's easy to see how, given the law pretty much states “you must be accessible”, people can assume there is some kind of passing grade. The truth is, accessibility is more of a scale rather than a switch. It's not something that is on or off.

You can think of the accessibility of your site as a range, from “totally inaccessible” to “a brilliant experience for everyone”. You want to be as near to “brilliant” as possible on this scale.



**Totally
inaccessible**



Website



**Brilliant
experience**

To achieve this, you need to employ several methods, including both automated and manual web accessibility testing. More on that a little later.



What is WCAG?



The Web Content Accessibility Guidelines (WCAG) aim to be the standard for web content accessibility that individuals, organizations, and governments should follow to ensure more accessible web experiences.

Developed by W3C, WCAG is a set of criteria and tests that help web developers and content creators meet the needs of people with disabilities, by providing clear guidance for making web content more accessible.

WCAG is made up of several “success criteria”, each of which can be passed or failed during accessibility tests. WCAG is made up of three levels, A, AA, and AAA, which you may consider as an indication of their importance (Level A is considered to be mandatory).



The ambiguity in WCAG

Silktide will grade you against WCAG for all the unambiguous things (i.e., those that a computer can test). There are certain levels of accessibility failings that would be unambiguously interpreted as totally inaccessible.

A simple example would be the image below. You're asked to indicate your favorite fruit by selecting one of the three pictures.

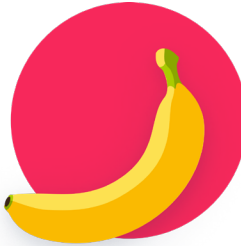
Without alternative text for each image (which describes the contents to screen readers), you couldn't know what you were clicking on.

You'd be asked to select "image", "image", or "image" by a screen reader. If you were, say, blind, that would be a classic unambiguous example, and a WCAG failure.

Please select from the following options



"Image"



"Image"



"Image"



What is alternative text

Alternative text ("alt text") may be added to images so that a screen reader user can hear an audio description of the information contained in that image.





There are many cases however where it's more subjective. There is a [requirement in WCAG](#) that content is easy to understand for as many people as possible. It states that you should write for an audience with a reading age of around 12-13 years.

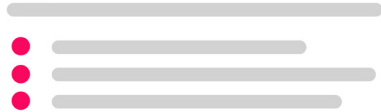
Below are two examples of the same written content. It's quite clear that what you see on the right is easier to understand than what is on the left. Indeed, a computer can determine this as well. We have all sorts of math and formulae that are designed to assess this.



Unnecessarily complex



Easier to understand



But how would you actually grade that? Would you grade one as a pass and one as a failure? Technically, the material on the left is accessible in that you can at least hear it or read it. It's just a bad experience. A pretty strong case in this example could

be made that this is failing a WCAG AAA requirement, although it isn't as simple as yes or no. What if we take the example on the right and add one sentence that fails that criteria. Would that make the entire page a failure? Would it make the entire website a failure? Probably not.

How effective is automated web accessibility testing?

Let's start by getting the elephant in the room out of the way. Automated accessibility testing alone will not make your website 100% accessible.

Always combine automated testing with manual auditing. The reason for this is that some WCAG criteria are ambiguous. For example, automated accessibility tests can tell you if alternative text exists, but what they can't do is understand the context.

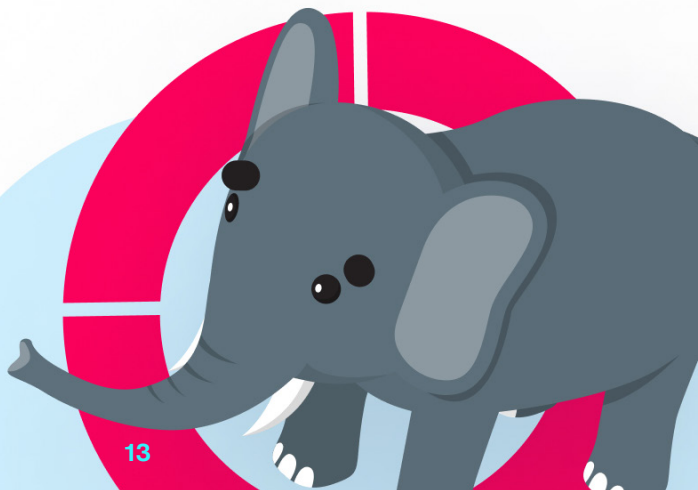
If you display an image of a banana, but your alt text says, "apple", a computer cannot determine its context and

therefore, whether it passes or fails.

That said, a good platform will show you a list of all your alt text and allow you to manually review them quickly.

What automated accessibility testing can do for you is find a broad range of issues quickly, across many thousands of web pages. It's scalable, saves you time and money, and helps you find obvious accessibility issues before your users do.

Automated testing also allows for consistency, makes your team more efficient, and allows you to report on progress towards your accessibility goals.





Chapter 2

Best practices for great web accessibility

Best practices for great web accessibility

There are several areas you can focus on that will help improve accessibility

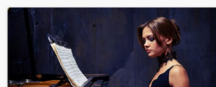
Following is a non-exhaustive list of common accessibility failures found across all sectors and industries by Silktide. We'll approach this in a non-technical way, but there may be some specific code examples to illustrate certain points. An understanding of HTML would be advantageous in these situations.

We'll include data from the Silktide Index where possible that illustrates which percentage of websites we frequently test have the issues outlined.

What is the Silktide Index?

The Index is the league table showing how organizations in various industries compare for web accessibility. Index includes a wide range of both public and private sector websites, across many industries and locations.

Index uses Silktide's automated web accessibility testing software to test and score websites against WCAG 2.1, the globally-recognized standard for web accessibility.



Keyboard navigation and assistive technologies

Not everyone uses a mouse to navigate your website. Whether through motor impairments or not, some users can only navigate using a keyboard.

Other assistive technologies like buttons and switches, which act in a similar way to a keyboard but with fewer buttons, are also used by disabled users.

Blind users also navigate with a

keyboard, typically using a screen reader. Some users simply use a keyboard by preference, for speed.

Of all the potential accessibility issues you could create, preventing users from navigating with a keyboard is the most fundamental to get right because it affects the most users.





Typical keyboard controls

Typically, the TAB key is used to navigate through elements on your web page.

These include links, buttons, form fields, radio buttons, and more. Pressing TAB skips to the next interactive element.

Pressing SHIFT and TAB skips backward to the previous interactive element.

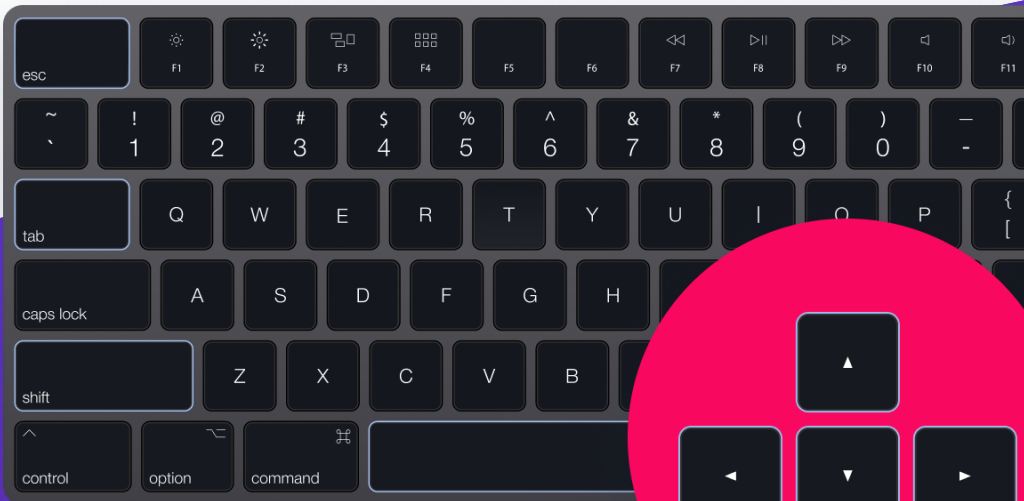
The escape (ESC) key is used to close dialog popups. SPACEBAR is used to activate buttons, to check and uncheck checkboxes, or to expand menus.

ENTER is also used to expand a menu, activate a button or link, or, additionally,

select an option from a list of autocomplete suggestions.

ARROW KEYS are used to navigate menus - up/down to move up and down, and left/right to expand and collapse sub-menus. They're also used to interact with sliders, and scroll

These are just a few examples. For these to work correctly, your website must be coded in an accessible way. If you don't, users will face the following potential problems when navigating with a keyboard.



Arrow keys



Focus states

Once focused, an interactive element can be manipulated or activated. It's mandatory to give users visual feedback on which interactive element is currently highlighted on your page. If a user navigates to a link using a keyboard, [it must be highlighted](#).

WCAG 2.1 specifies how the focus element should appear in terms of the contrast between text and its background, its size and shape, and that it should not be obscure by any other elements. Making focused elements visible is a Level A requirement in [WCAG A \(2.4.7\)](#)

84% of websites in Index do not ensure controls change appearance when they are selected (WCAG A 2.4.7)

Focus order (or Navigation order)

The order in which elements receive focus when using keyboard navigation is also important. The structure of your code dictates this, not how the elements look on the screen.

Generally speaking, [you must ensure that navigation is logical and intuitive](#). That means organizing your page to follow the visual flow of left to right and top to bottom.



Poor website structure also impacts screen readers' ability to read the contents of the page in a logical order and it also impacts navigation.

Therefore, to create a more keyboard-accessible experience you should:

- Pay attention to the semantic structure of your web page by ensuring the reading and navigation order is correct
- Use CSS to control the visual presentation of elements if necessary

Skipping to content

Lengthy navigation can cause problems for users who need to get to the main content of your web page quickly.

Let's say your menu has 50 items, ten in each of five columns. Because the navigation appears semantically high on your page, keyboard users must tab through every one of these items, in turn, just to get to the main content.

That is a fundamentally terrible user experience for everyone using keyboard navigation, but far more so for users with motor impairments or blind people.

The solution is to ensure you have a "Skip to content" link on your website. It

shouldn't be made visible. Generally, this should be the first focused link on the page on pressing TAB. Once selected, the user should be sent straight to the main content (sometimes coded as a #main element in your website) - that is, everything after the navigation bar.

Multiple "skip to content" links are also possible, depending on how complex and information-rich your website is. For example, you might want to include a "skip to search" link as a second option.

WCAG lists this Level A success criterion as "[Bypass Blocks](#)" in their documentation(2.4.1).



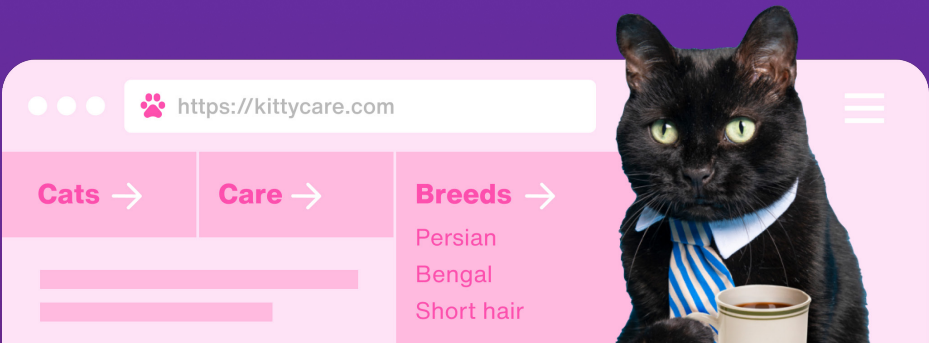
Tab stops

Let's imagine a user who needs to scroll through the five menu lists in the example above.

Tab stops allow users to step through the top-level menu items in turn. Pressing ENTER then opens the menu and allows you to tab through the lower-level options.

Let's imagine a user who needs to scroll through the three menu lists in the example below

That would be a poor experience for everyone using keyboard navigation.



Be consistent in page layout and semantics

Ensure that any navigable elements found on multiple pages are placed consistently on those pages.

That means if your website contains a navigation menu, a search box, or a login button, [you need to ensure that these remain in the same order on every page.](#)

Keeping the same semantic structure helps keyboard navigation and screen readers. Maintaining the same visual structure reduces the cognitive burden of trying to find a moved menu item between different pages.

Our webinar [“Accessible Web Navigation”](#) explains these concepts in more detail.

Add alternative text to images (where appropriate)

Alternative text (or “alt text”) is used by screen readers to describe the contents of an image to a user who may have difficulty seeing it. It should convey meaning where possible.

Content editors, not developers, are generally responsible for the bulk of the website material. This section is focused on helping them understand the implications of implementing alt text incorrectly.

Alt text is covered in WCAG success criteria 1.1.1, [“non-text content”](#).

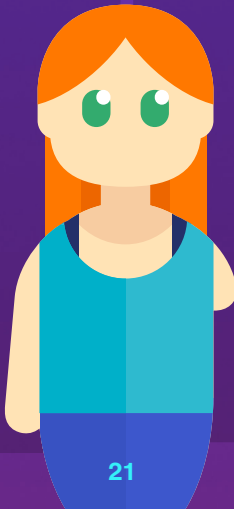
In code, alt text is added using the alt=“your text goes here” attribute. You must use this attribute on all images on your website.

However, you don’t need to include text inside the quotation marks all of the time. I’ll cover this a little later in the “Always think of the context” part of this section.

But for now, just remember that alt=“” is required. Period.

63%

of all websites do not specify alt text for all media.



60%

of all websites do not specify alt text for images inside links



What is alt text for?

Consider this image, which depicts a photograph of a pear, a banana, and an apple. Let's say this appears on an online survey with a message asking you to choose your favorite fruit.

Please select from the following options



Sighted users can easily see each photo. They can use a mouse or keyboard to navigate to one of the pictures and make their selection. Now, consider a blind or visually-impaired user navigating with a screen reader. With incorrect or omitted alt text, these images might be read out as “Image”, “Image”, and “Image”.

Please select from the following options



“Image”



“Image”



“Image”



With correct alt text, the image would be read out as “Pear”, “Banana”, and “Apple”.

Please select from the following options



“pear”



“banana”



“apple”

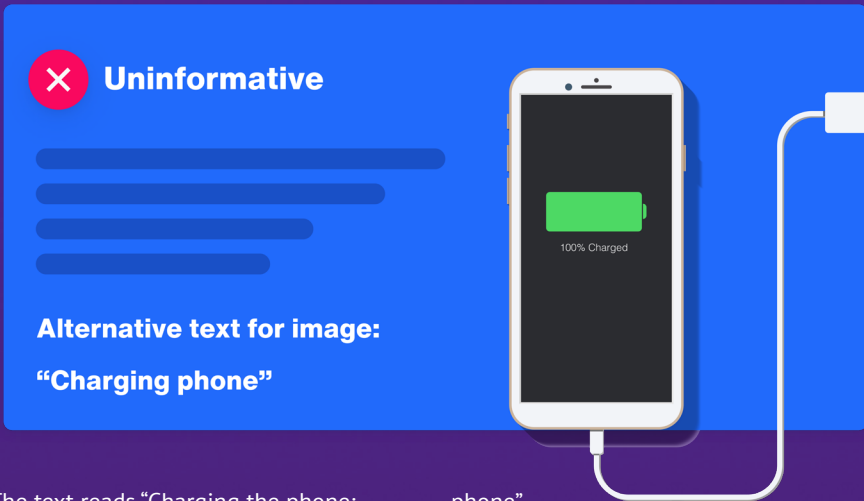
This is the reason we add alt text. Whatever you add to your images will be read aloud.

This example shows how incredibly useful alt text is to screen reader users. In fact, in this example, it’s essential. Without alt text, screen reader users simply wouldn’t be able to complete the action of choosing a fruit.

So just add alt text to all images and we're done?

Not quite. You might think that just adding alt text with abandon solves the problem. You would, however, be wrong, unless you consider the context of your

alt text. Let's take this example. It's an image that shows instructions on how to charge your phone.



The text reads “Charging the phone: Connect the phone to a power outlet using the cable and power adaptor provided”.

The image next to the text shows a picture of a phone and charger, illustrating where to put the power cable both at the phone end and at the power outlet end.

The alternative text reads “Charging

phone”.

Now, this is an example of an image that conveys useful, in fact in this case, critical information, but the alt text does not. In this case, a screen reader user will not get specific instructions on where to plug the power cables.

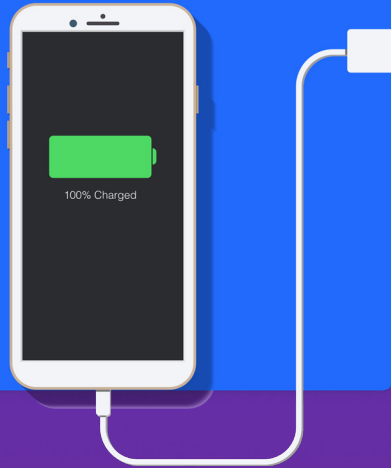
This next image is identical, except the alt text in the image reads “Plug cable into the bottom edge of the phone”.



Informative

Alternative text for image:

**“Plug cable in the
bottom edge of the
phone”**



So now, a screen reader user knows they must plug the cable into their phone to charge it.

Sighted users can see enough information in the image that they probably wouldn't even need to read the text to be able to work out what they need to do.

You can understand that it's not quite as simple as just adding text. You need to make a judgment call as to whether you think the image conveys information that is not otherwise present in the surrounding text.

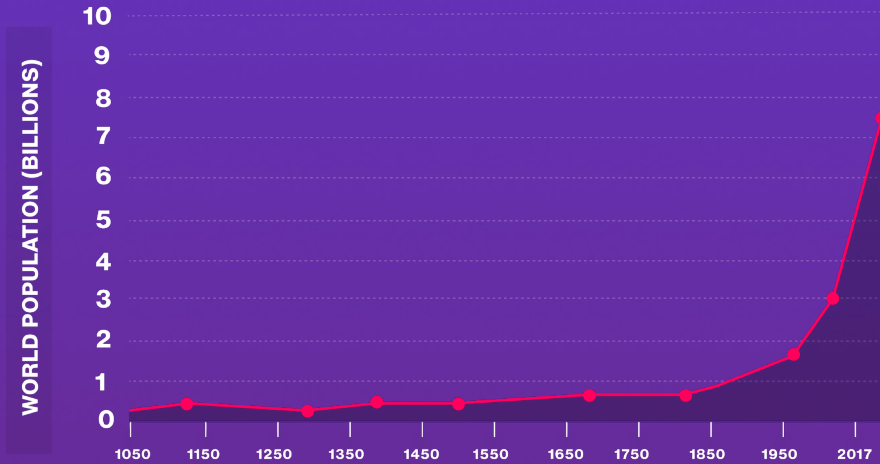
Information presented in a graph

Let's take another example of crucial information presented in an image. One of the most popular forms of visual communication of data is the graph. Imagine you have a picture of a graph showing human population growth.





Human population growth



If you do not add either body text or alternative text describing what the information shows, then this becomes useless to a screen reader user.

If this image appears without an explanation, and the alternative text is set to something like “Human population growth”, then this is not only unhelpful but creates a barrier to accessing this information.

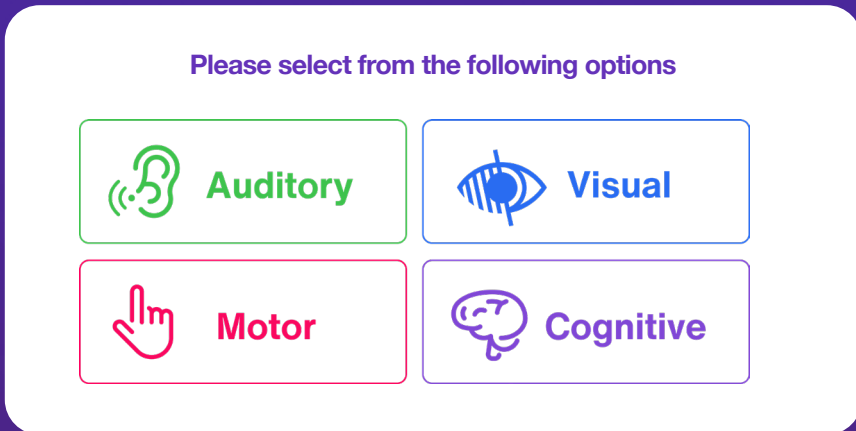
Instead, the alt text should include the information the graph is trying to convey. In this example, “Human population growth increased from under 1 billion in 1000 to 7.5 billion in 2017” would be more useful alt text.

Again, context is key here. You need to consider what is happening in the surrounding text of the article or blog post, or whatever you are writing.



Too much, or unnecessary, alt text?

Let's take a common example of a menu containing both text and an image or icon for each item.



The menu contains four text links, Auditory, Visual, Motor, and Cognitive. There is an icon for each option next to each link.

What's the correct way to present alt text in this situation?

Well, if your idea is to add alt text to the icons explaining what they are, then you'd actually be making the experience worse for screen readers.

In this case, each menu item would be read out something like "Ear receiving sound – Image" followed by "Auditory – Link".





Please select from the following options



Auditory - link



Visual - link



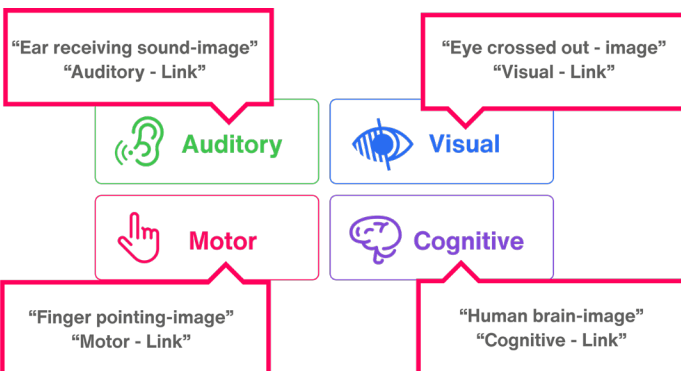
Motor - link



Cognitive - link

Consider the context here. Is a description of the icons in this menu useful? They don't add to the utility of this menu. Rather, they take away from it. Instead, you should specify blank alternative text for each icon in the list.

Remember the mandatory code attribute `alt=""` from earlier? Use that, but don't fill anything in between the quotation marks. This alerts screen readers to ignore the image entirely, and mark it as purely decorative, which is an important accessibility concept. By marking these icons as decorative, a much more pleasant experience for the user results, as only the link text is read aloud.





The icons themselves still serve their purpose to sighted and even dyslexic users, so they should remain of course. But for a visually impaired user, alt text would be a hindrance if added.

In this new example, the images are ignored by screen readers, and the menu would be read out as “Auditory - Link”, “Visual - Link”, “Motor - Link”, “Cognitive - Link”

What happens if I don't have an alt attribute on my images?

There is a big difference between specifying blank alt text and omitting the alt text attribute (`alt=""`) entirely.

If you do not include the alt attribute at all then a screen reader will do its best to read out what it thinks should go in there, often with terrible results.

The best guess is usually just to read the filename. If that's “Screenshot 2020-11-06 at 16.21.25”, that's a big problem for your screen reader users.

Your CMS or website theme should ensure that the alt attribute is always present on every image, with the option for your content editors to add the alt text on a per-image basis.

If they forget (and they will) then the image will be ignored entirely by screen readers. This is arguably better than having a filename read aloud, but still not a great experience if the information contains critical information.



Always think of the context

You should consider the following questions when deciding whether to add alternative text.

- Is this image purely decorative?
- Will important meaning be lost if no alt text is added?
- Does text content include important information present in this image?
- Will adding alt text to this image make the user experience worse?

And if you decide that alt text should be added, consider:

- How can I describe this image's information in context?

Finally, remember, there is a big difference between specifying blank alt text and omitting alt text tags entirely. By omitting, the screen reader will probably read the filename of the image.



Pay attention to spelling in alt text

Make sure you're spelling alt text correctly. Even though it's not visible on the page, spelling errors may confuse screen reader users.

Search engines also look at your alt text and use it when presenting the results of an image search, which can be quite useful for SEO. Misspellings here may impact your SEO to some extent.

Context and automated accessibility testing

Computers are not (yet) capable of understanding whether the alt text you add describes the information in that image in a useful way. They can alert you if alt text is missing, and they can even tell you if your alt text seems nonsensical (e.g. if it is a filename).

Silktide, for example, gives you this information but also prompts you for a manual review of whether alt text is required or is appropriate, for all images on your websites.

Ensure subtitles, captions, transcripts and audio descriptions are present

Everyone should be able to access video content. Accessible videos include the following features:

- Subtitles
- Audio descriptions
- Transcriptions
- Captions

The accessibility requirements for video and audio media are described in [WCAG 1.2.1](#) and [WCAG 1.2.2](#).





- how's your day been so far?

Subtitles

Did you know that 70% of videos on Facebook are watched without sound?

From an accessibility standpoint, subtitles are the text version of spoken dialogue and narration. They are synced with the content on the screen.

Subtitles are arguably the absolute bare minimum you should include in all video content for better accessibility. However as you'll see shortly, captions, transcriptions, and audio descriptions are just as important. You should include all of them where appropriate.

Users in loud rooms or in other locations where volume cannot be increased (like trains, or the library) also benefit from subtitles.

They also help non-English speakers (or

indeed, non-native speakers of whichever language the audio is recorded in) understand content more clearly, especially if, for example, the words are spoken quickly or in a regional accent.

There are several tools to help transcribe audio into subtitles. If you're uploading videos to YouTube, for example, it will do it for you (with varying degrees of success - you will likely want to review the file once completed and perform any manual changes).

There are a few different file formats available, but they all accomplish the same thing in the same way. Spoken words are split up and timestamps are added, and this is how media players know when to show the text on the screen.

Captions

Captions are designed for deaf people and are used to describe audio content that is not narrated. Without captions, the context of what a deaf person sees on screen can dramatically change.

Take this image. The protagonist, Anna, walks along the train tracks facing away from the camera. She says “What a beautiful day”.

The caption below reads “[Train approaches]”

You can understand how the context of the image changes dramatically with the new knowledge that there is the sound of an approaching train off-camera.



Transcriptions

These differ from subtitles in that they are not time-coded snippets of text, but rather, the full text of spoken dialogue written out in complete sentences. Transcripts are designed so that users can read the spoken video content, or use a screen reader to read it to them.

They include only the dialogue. They do not include descriptions of what is happening on screen.

Audio descriptions

Subtitles and transcriptions can't tell the whole story, as often what is happening on the screen is not narrated by the characters.

Audio descriptions introduce another voice that describes what is happening on the screen. This is incredibly useful for blind users who otherwise would only be able to hear the dialogue. But what if there is no dialogue?

A great example of how audio descriptions work in video content is [the Frozen 2 Disney trailer on YouTube](#).

25%

of all websites fail to
provide captions for all
prerecorded audio and video



Do not automatically play media without allowing users to stop them

In code, it's possible to set HTML media (video or audio) to play using the autoplay attribute, and further, it's possible to set the media to loop with the loop attribute.

However, it's bad practice to do this without also allowing user control. It's also considered good practice to mute the audio by default.

For any audio over 3 seconds long,

and video over 5 seconds long, autoplay without adding user controls is a [WCAG accessibility failure](#).

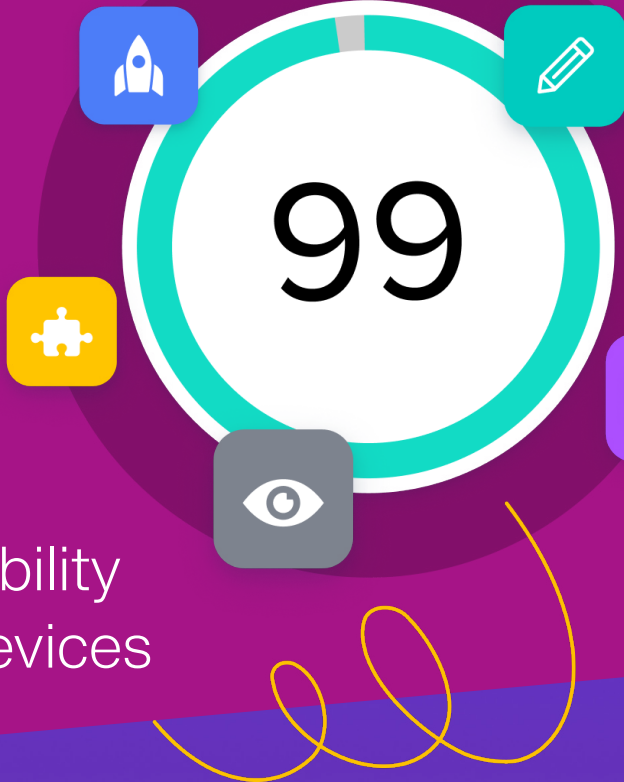
Animations over 5 seconds long should also contain controls to pause or stop them. All automatically looping media must contain controls, regardless of their length.

Flashing content

Warnings must be given for any content which includes flashes, strobos, sudden loud sounds, or other possible triggers for seizures.

[Flashing content with more than 3 flashes per second is not allowed](#), unless below certain thresholds for size and color contrast.





Test accessibility on mobile devices

When the first iteration of WCAG 2.0 was released in 2008, mobile phones capable of giving rich Internet experience barely existed (the first iPhone was released in 2007).

Accessibility fell short as mobile device technology improved over the next ten years, so WCAG 2.1 introduced a range of new testing criteria to address this.

Now, mobile device usage has overtaken desktop usage worldwide.

The main accessibility issues faced by mobile users include:

- being unable to view content in different orientations

- the content on the screen moving in all directions when trying to scroll
- being unable to zoom in and out of a website and making fonts larger, without breaking the layout

There are others, including ensuring you have sufficient button sizes and that you've implemented autocomplete on form fields (this is not exclusively a mobile issue but certainly more useful for mobile users). There's more on autocomplete in the "Form styles and labeling" section a little later.

Let's consider these issues, and why they are important, in turn.



Device orientation

People need to be able to view your content on mobile devices in any orientation. That means your content should work for them whether their device is in [portrait](#) or [landscape mode](#).

Consider a user with cerebral palsy who uses a wheelchair. They may have a tablet attached to their wheelchair. Your application or web page should work regardless of whether the device is attached vertically or horizontally.

If you've employed a responsive design when building your site, then it should already be viewable in both orientations.

Two - dimensional scrolling



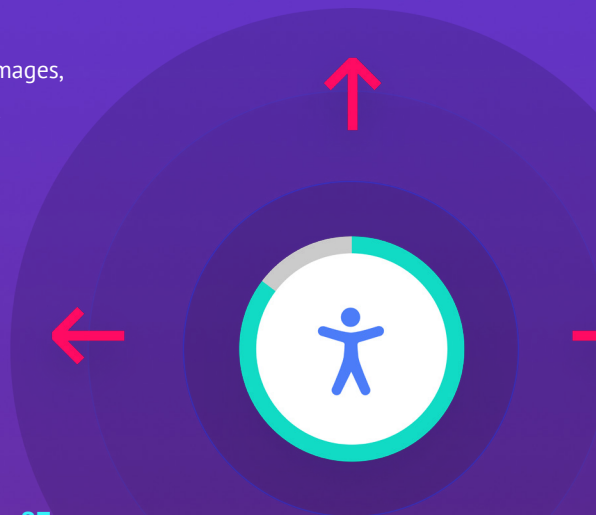
[Web pages should only scroll in one dimension](#). This means you should be able to scroll either up and down, or left and right, but never both at the same time.

There are two main reasons that two-dimensional scrolling fails WCAG tests:

- 1) People with motor issues may not be able to scroll in two dimensions
- 2) Cognitive burden increases when scanning text across multiple screen widths

Two-dimensional scrolling occurs when images, tables, or other on-page elements are not constrained to their container width (i.e. they are too wide for the current screen width).

Having to remember what line of text you were on when scrolling to the next line of text increases the cognitive burden.





61%

of websites fail to ensure pages don't scroll in two dimensions on small screens

Disabling pinch-to-zoom

Users must be allowed to zoom into a website using their mobile device to increase the size of the text without breaking the layout.

Text and other on-page elements must flow correctly once zoomed in.

[Pinch to zoom](#) is enabled in web browsers by default, so in failing this accessibility check in WCAG you would need to go out of your way to turn it off.



25%

of all websites disable pinch-to-zoom





Begin with the design

Designers and UX teams should be aware of the need for accessibility in the early stages of a new web project.

Specific examples of design impacts on accessibility include:

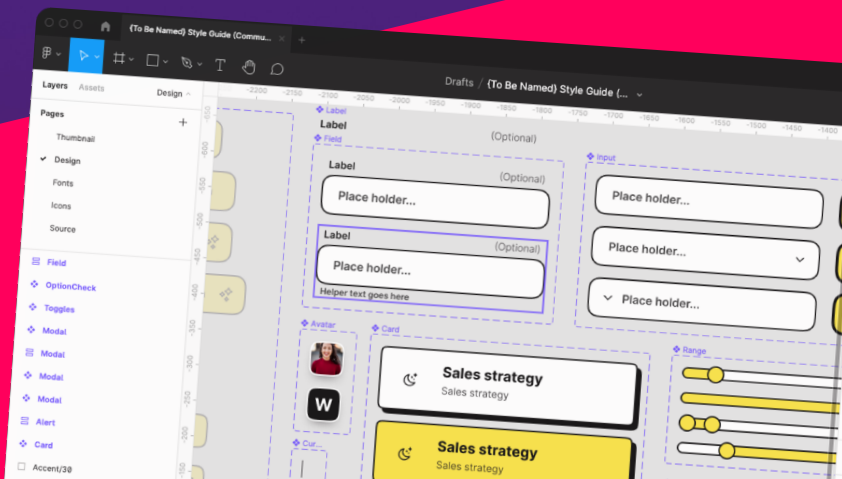
- Typography and font choice
- Letter spacing
- Color contrast of text and its background
- Minimize use of carousels
- Hierarchy and semantic heading structure
- Form styling and labeling
- Focus styling
- URL structure

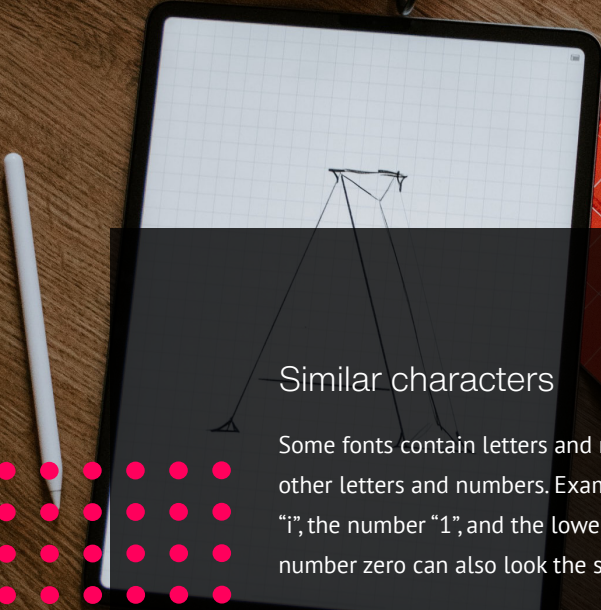
There's a lot to consider, but laying out a style guide at the beginning will help your developers implement these when coding.

Typography and font choice

By choosing the correct font, you'll improve legibility and readability for people with moderate visual impairments, dyslexia, or a learning disability. But how do you choose a “correct font”?

Ideally, you should consider the following issues.





Similar characters

Some fonts contain letters and numbers that look the same as other letters and numbers. Examples often include the upper case “i”, the number “1”, and the lower case “l”. The uppercase “o” and the number zero can also look the same.

Ensure that additional strokes are included to distinguish these characters from one another.



Uppercase O



Zero



Uppercase i



Lower case L

Mirroring

You should also ensure that mirroring is reduced. The lowercase letters “b” and “d”, or “p” and “q” are often confused by young children and, sometimes, in later life by adults.

To avoid issues here, use a font that distinguishes these letters from one another by including extra strokes, for example above the “p” and below the “b”.

bd pq

Text spacing

[Increasing space between and around text](#) helps people with:

- Low vision
- Dyslexia

To ensure good readability, you should apply the following minimum attributes to website text:

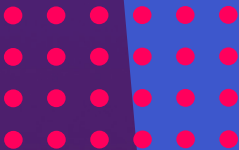
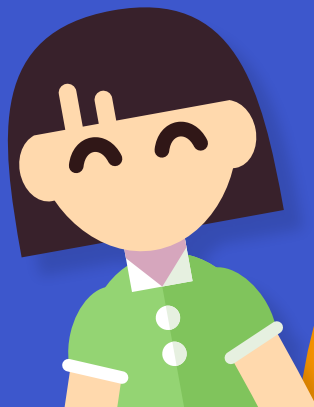
- Line height of at least 1.5 times the font size
- Spacing following paragraphs to be at least 2 times the font size
- Letter spacing to be at least 0.12 times the font size
- Word spacing to be at least 0.16 times the font size



The quick brown fox



The quick brown fox



Color contrast of text and its background

It's important to maintain readability by ensuring [the contrast between text color and its background is sufficient](#).

Insufficient contrast affects people with low vision, and people using devices in bright sunlight. For example, this yellow text on a red background fails all levels of WCAG.

AA
AAA

2.82:1

#FD0101 #FFFA7B

Normal text

The quick fox

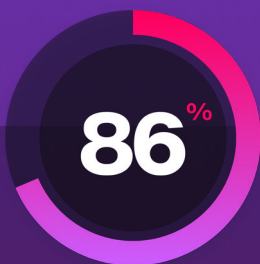
AA Fail AAA Fail

Large text

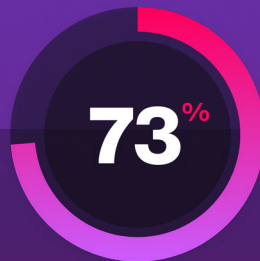
Large text

AA Fail AAA Fail

Normal text requires a contrast ratio of at least 4.5:1, and large text must be at least 3:1.



of websites do not ensure text
has sufficient contrast



of links are indistinguishable by
more than just color

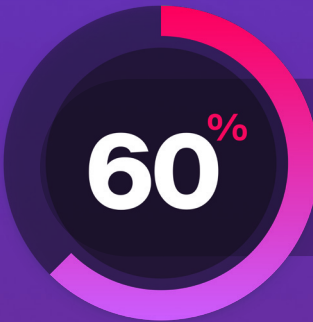


Page headings and semantic structure

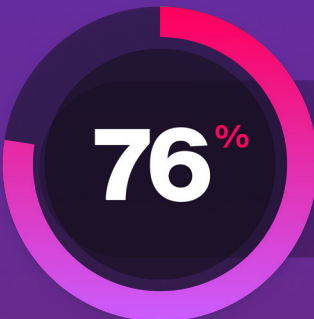
We briefly mentioned earlier that screen readers rely on the [semantic structure](#) of headings and content. Your design must ensure:

- Only one h1 per page
- Subheadings are nested and do not skip
- Paragraphs are nesting within headings
- Navigation, main content, sidebars, and footers appear in code in the correct order, regardless of where they appear on the page

Developers are responsible for implementing your design but you must ensure you give them the correct information to work from.



of websites fail to ensure every page contains a heading



of websites fail to ensure every heading contains text



Form styling and labeling

Accessible forms are not just about design, they're also about user experience.

Creating accessible forms is a huge topic in itself. There are, however, a few fundamentals to get right in the design process.

You should pay attention to their appearance, the semantics, and the user flow.

Appearance

There are several elements available to you when creating forms, including fields, dropdowns, checkboxes, star ratings, and buttons. It's important to ensure you adhere to the [WCAG non-text color contrast specifications](#) when creating these elements.

Boundaries

This is a relatively new concept in WCAG 2.1 that refers to the contrast of form elements. Specifically, whether or not you can determine where the element is and what it is, visually.



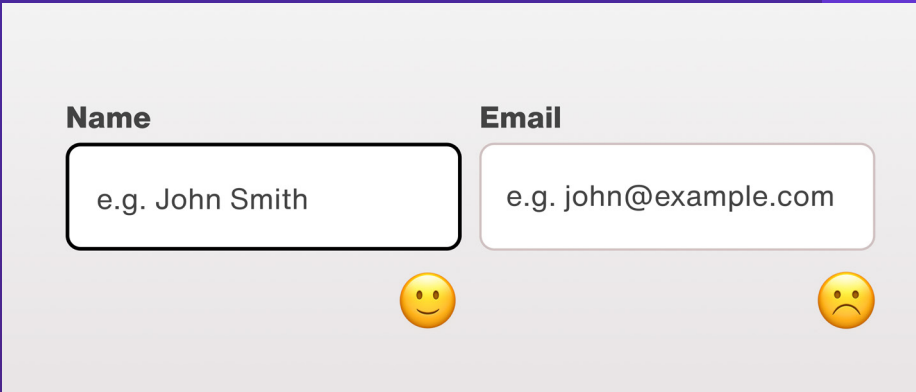
Submit

mes@company.com



You should ensure that there is relatively high contrast between the form element and its surroundings.

Take these two examples:



On the left, the input field has a thicker, high contrast border that easily distinguishes the form field from the background.

On the right, there is a light grey border against a white background, creating a less obvious distinction between the two.

The example on the right would not pass the WCAG success criterion for boundaries (which in

this instance is [1.4.11: Non-text contrast](#)).

A key aspect of this requirement is not that you must add thick black borders to everything, but that you must make the elements clear.

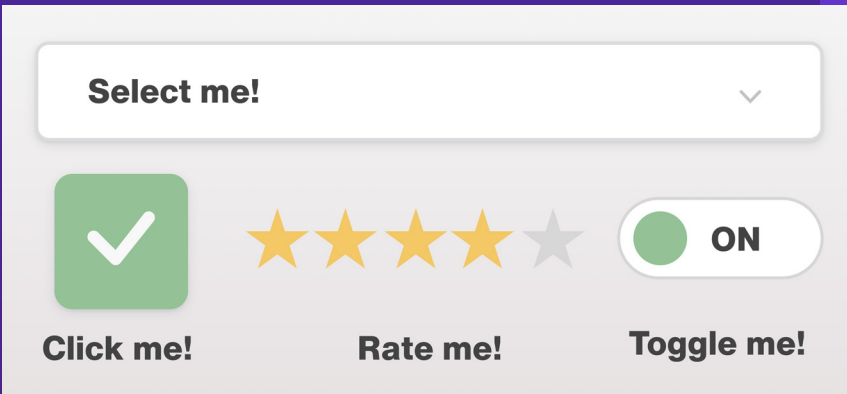
Another way to achieve this is to simply change the background color accordingly. However, you must ensure that this difference has a contrast ratio of at least 4.5:1 to meet WCAG Level A.



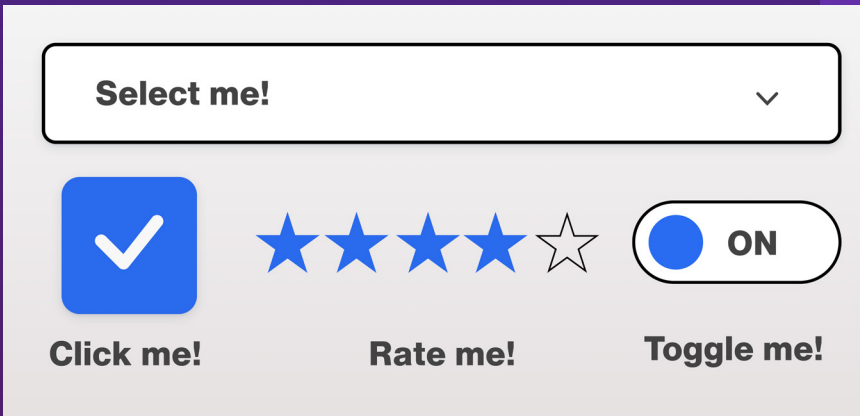
Further examples of insufficient contrast

This image depicts a form containing a dropdown menu, a checkbox, a rating widget, and a toggle button.

None of these elements would pass WCAG as at least one aspect of their color contrast is too low.



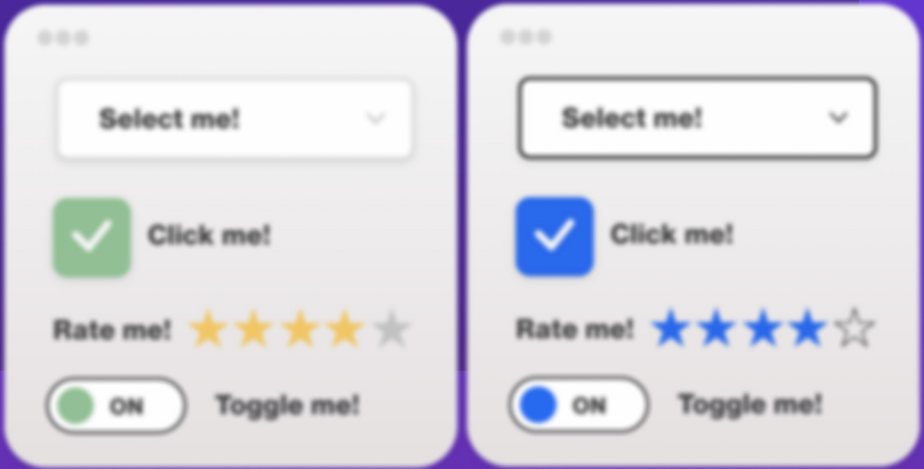
If you compare this with a more accessible variant, you can see the difference between the two.



The borders around elements are darker, the text is more contrasting with its background, and elements like the reviews widget are clearer.

So, why is this important?

Well, people with moderate visual impairments might view your form like this:



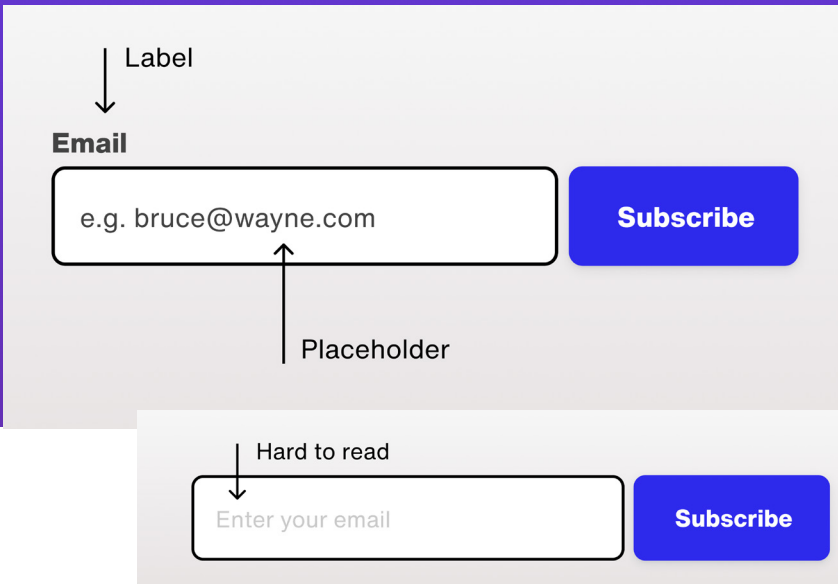
This is a simulated view of moderately blurred vision. Notice how the borders around the dropdown and the toggle button are invisible in the version on the left?

A lot of important information is completely lost. Insufficient contrast prevents people with low vision from using forms.

It's therefore very important you ensure you adhere to the WCAG color contrast specifications when creating your form. You should also ensure that focused elements (the highlighted input field) are sufficiently differentiated from other fields.

Labels and placeholders

Labels and placeholders are not the same things. Labels are necessary and should sit outside of the form field as a permanent reminder of the purpose of that field. Labels are usually placed above the field they are related to.



Placeholders can be used to illustrate examples of the content that needs to be input into a particular field. However, placeholders are not substitutes for labels. This is because when a user starts typing into a box, the placeholder text disappears. This adds an unnecessary cognitive burden, as people can forget the purpose of the

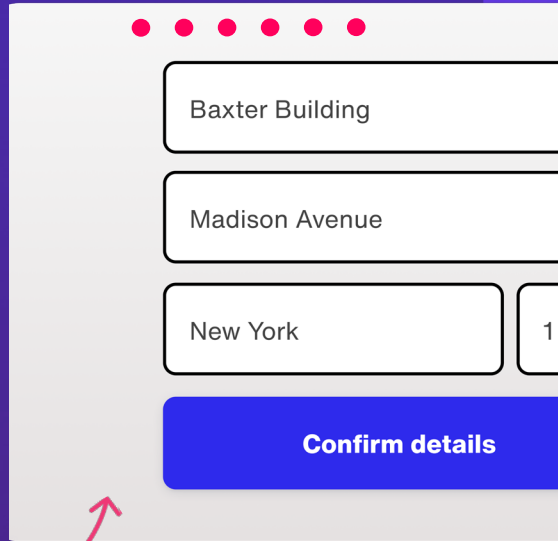
field once the placeholder text is no longer present. So, where do people get this wrong?

In this example, the developer has created this form field and used the placeholder as a label. There are a few problems with this approach.

Firstly, the grey placeholder text on this white background causes issues for people with moderate visual impairments as outlined above - they may not even see this text.

Secondly, when you start typing in this field, you no longer know what information you are being asked to provide. This is because placeholder text disappears when you begin to type.

This example may seem slightly simplistic, as there is only one field. But consider a multi-field form with only placeholder text, like this.

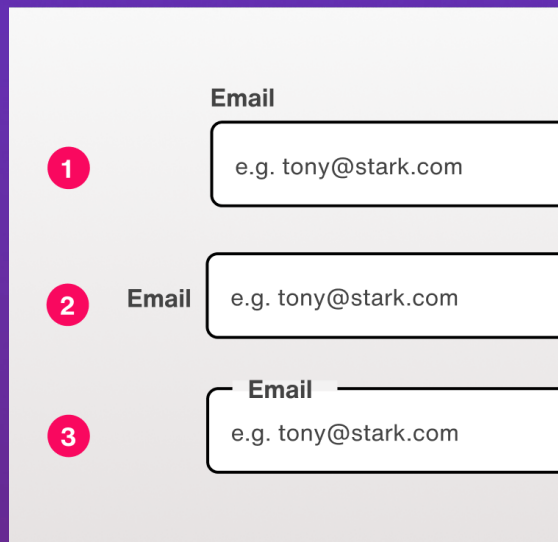


A multi-field form with a light grey background. At the top, there are two rows of red dots. The form contains four input fields: 'Baxter Building', 'Madison Avenue', 'New York', and a field with '1'. Below these fields is a blue button labeled 'Confirm details'. A red arrow points from the text 'like this.' to the form.



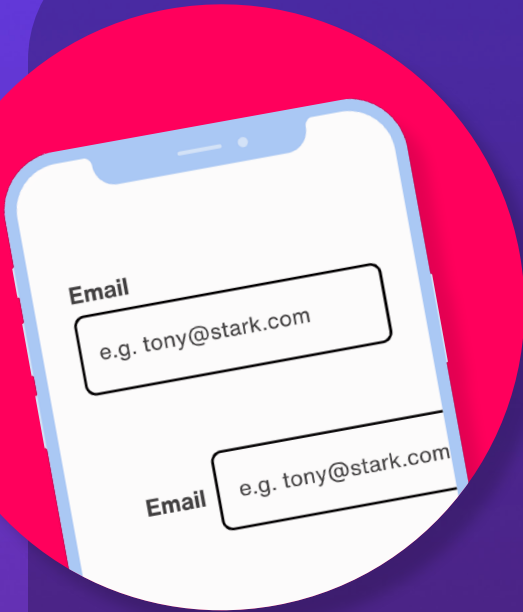
Because the user has already filled in similar forms, parts of this form have already been autocompleted by the browser, you'd have to take an educated guess as to what each field is supposed to be.

Here are three common ways labels are implemented in form design.



Three examples of email field labels. Each example shows a red circle with a number (1, 2, or 3) next to the label 'Email' and an input field containing 'e.g. tony@stark.com'.

- 1 Email e.g. tony@stark.com
- 2 Email e.g. tony@stark.com
- 3 Email e.g. tony@stark.com

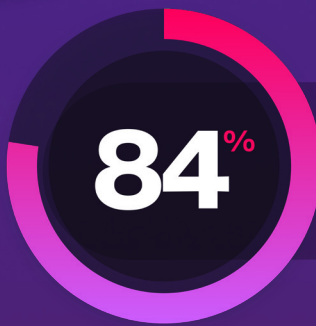


The first approach has the obvious advantage of being able to fit on a small display.

The second is perhaps not so good for mobile devices, as there's less room available to type in the box.

The third, while quite trendy, may or may not be better for accessibility. However, none of these are strictly covered under WCAG so you should use your best judgment when choosing.

The importance of form labels cannot be understated. They are therefore mandatory under [WCAG 3.3.2](#).



websites fail to ensure form controls have labels

Focus

Each form control has a 'focus state'. This is implemented as a way to clearly see which element is currently selected, usually by changing its color.

In this example, the email input box has been selected and is currently highlighted with an obvious blue border.



Email

e.g. bruce@wayne.com



I agree to all the things

Sign me up



While most people are familiar with this, something that is less obvious to many developers is that checkboxes and buttons also need to be highlighted when focused.

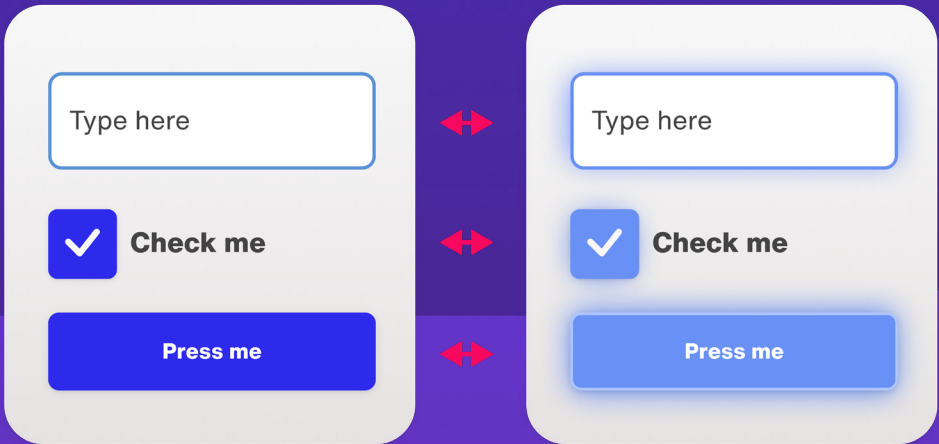
The reason it's less obvious is that most users would select these controls using a mouse because they can easily navigate to them and interact with them visually. Alternatively, they might be used to tapping on them if using a mobile mouse

But what about users who select form controls differently?

People who are unable to use a mouse or who otherwise use keyboard navigation may instead tab through form controls. They rely on those controls to change visually so they know which is currently selected.

WCAG 2.1 does not state specifics as to how you should implement this, only that you should.

In the upcoming WCAG 2.2 specification, however, clear guidelines have been written to state just how contrasting the focus state should be for these form controls.



Without focus, forms are unusable via keyboard. This corresponds to many assistive technologies, like switches.

Focus state is required under [WCAG 2.4.7](#) and [1.4.1](#)



Valid and invalid states

Commonly, when a user enters incorrect information into a form, they cannot proceed. There's usually some feedback as to why this is the case, and the user is given the opportunity to correct their information before proceeding.

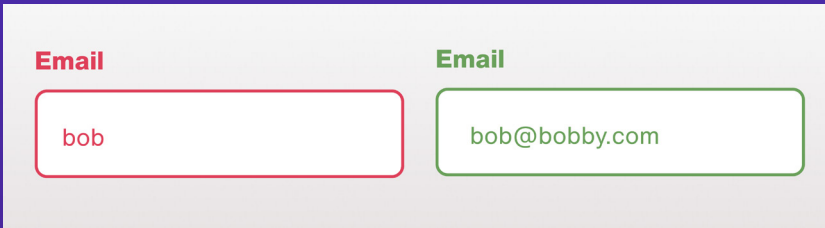
Obvious examples include the mistyping of an email address or an incorrectly-formatted zip code.

The mechanism by which a user is

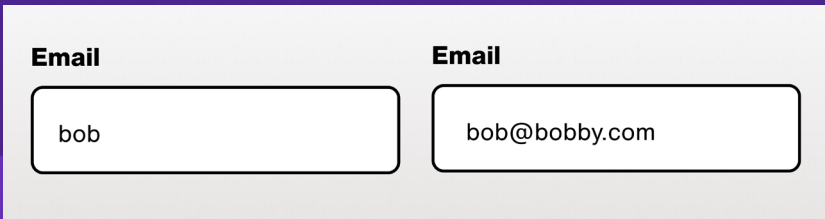
given this feedback is important. Often, developers implement color to indicate that a field has incorrect information inside.

In the below example the user is made aware that the information in the field on the left is incorrect, as it does not contain a complete email address, by adding a red color to the form control.

The field on the right is indicated as correct by being colored in green.

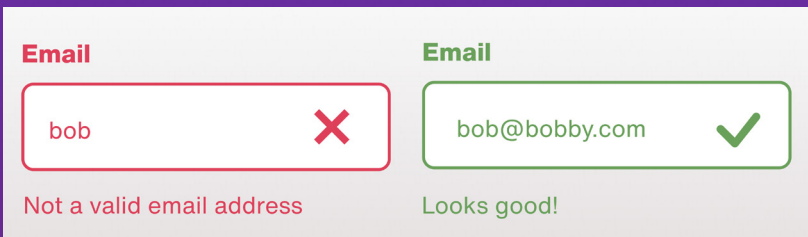


However, imagine that your user is color blind or has some other impairment meaning that distinguishing between these two colors is difficult. They may not see color at all, such as in this example.



In this case, the form does not work and the user doesn't know why. Using color alone is not an accessible way to convey information.

The solution is to either add a visual marker, like a tick or cross symbol or add additional explanatory text. Ideally, you should do both for the most accessible experience.





▶▶▶▶▶ Even without color, this example still works.

Email <input type="text" value="bob"/> ✘ Not a valid email address	Email <input type="text" value="bob@bobby.com"/> ✔ Looks good!
--	--

So, why is this important?

Not all users can see color, and without an alternative method of conveying information, this becomes inaccessible.

Conveying meaning without using color alone is a requirement under [WCAG 1.4.1](#)

Autocomplete

Autocomplete is an essential requirement of WCAG 2.1 ([WCAG 1.3.5 Identify Input Purpose](#)).

The key thing to remember is that autocomplete is a way for you to say what a field is supposed to contain in a standard way, so that if possible, the browser will fill

this information in automatically for the user.

[There are many standardised autocomplete attributes available](#). The most common are things like **name**, **email**, and **tel**. In code, the input type is set to one of these attributes.

```
<input type="tel" name="home-phone" autocomplete="home tel">
<input type="tel" name="work-phone" autocomplete="work tel">
<input type="email" name="home-email" autocomplete="home email">

<input type="text" name="shipping-city" autocomplete="shipping address-level2">
<input type="text" name="shipping-state" autocomplete="shipping address-level1">
<input type="text" name="shipping-country" autocomplete="shipping country-name">
<input type="text" name="shipping-postal-code" autocomplete="shipping postal-code">
```



Browsers then attempt to fill this information automatically from the user's device.

Why is autocomplete important? Well, WCAG 2.1 [aims to minimize typing](#) where possible.

If information is pre-filled, then people with language and memory-related disabilities benefit by not having to remember complex information.

Any users with moderate to severe motor impairments may have limited use of a

keyboard, or not use a keyboard at all.

Autocomplete greatly enhanced the experience of these users by reducing the burden of inputting characters with some assistive technologies (for example, a keyboard switch).

Finally, autocomplete benefits all users by simply speeding up the process of filling in forms. Typing in postal addresses into online checkout forms can be avoided completely, as can pre-filling email addresses into product signup forms.

Use descriptive and unique URLs

[Links](#) are the most important way for users to navigate through your website.

People use links to:

- Navigate your content
- Download or view documents online
- Open their email program automatically
- and many more

All links should:

- Avoid duplicating link text unless the destination is the same.
- Link text should be meaningful when read out of context
- Help users understand where the link goes

In terms of accessibility, these items are important, because blind users or those with moderate to severe visual impairments may use a screen reader when visiting your website.



Due to the nature of how screen readers work, each link is read aloud. It's possible to navigate through your page by tabbing to each link, in turn, skipping the content in between.

Therefore, if all your link text says "Read more" or "Click here", the user will hear:

"Read more" "Read more" "Read more"

This is not a great user experience. Would you know where the link goes by listening to these options?



Undergraduate

We offer the finest degrees
in all the land

[Read more](#)



Postgraduate

That piece of paper looks nice,
you should get another

[Read more](#)



Another example is a website that has these two sentences, linking to news about the company and news about its products.

Read more about our company [news](#).

Read more about our product [news](#).

If a user is skipping through links on your page, the text in the sentences will not be read, and all they will hear is:

"News - Link" "News - Link"

In this instance, a solution would be to change the link text so that they are a) unique, and b) understandable out of context. For example:

- [Read more about our company news.](#)
- [Read more about our product news.](#)

Now, the screen reader will say:

“Company news - Link”
“Product news - Link”

It’s a simple change, but not something the average content editor will be aware of. Most aren’t even aware they aren’t aware of it.

How to ensure better-quality links

When you add links, think how they would look and sound if there was no surrounding text. Do they still make sense out of context?

Even the examples above could be made better. Change the entire sentence to a link, like this:

- [Read more about our company news.](#)
- [Read more about our product news.](#)

Screen readers would say:

“Read more about our company news - Link”

“Read more about our product news - Link”

Other best practices are:

- When linking to an email address, encompass the whole name, for



of websites do not explain the purpose of their links.

example, [Dave Smith](#).

- When linking to a document, including the file type and size. For example, [Download our guide to accessibility \(PDF, 3MB\)](#). This is so the user can know in advance how big the file is and approximately how long it will take to download.

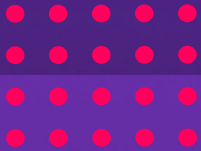
Avoid adding a full url in the text of the link. Instead of writing “[Visit https://silktide.com/](#)”, write “[Visit the Silktide website](#)”.

Minimize the use of PDFs or, if unavoidable, make sure they're accessible.

PDFs are generally the bane of any web content manager's life, especially those in sectors that must adhere to accessibility legislation. This is because they are difficult to make accessible and staff don't understand why they need to. It's also difficult to retrospectively make PDFs accessible without the source document available.

Best practice is simply to avoid PDFs entirely. Create HTML (web) versions of PDFs. You can add information to web pages and build accessible forms with less effort than creating PDFs, plus electronic form submissions are easier to track.

For more information [visit the UK Government Digital Service website](#).



“

Staff don't understand
why they need to make
PDFs accessible ”



PDFs are easy to make look consistent, but they are far harder to make accessible. By default, the information you need to make them accessible is simply not there.

So in summary, PDFs are great for:

- Portability
- Permanence
- Printed media

Unfortunately, these same qualities make them terrible for:

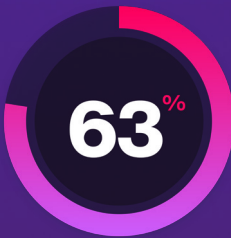
- Accessibility

- Maintainability
- Mobile experience

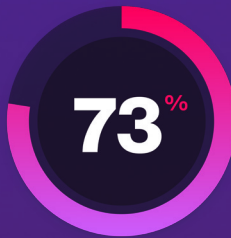
If you've read all this and are still adamant that you want to create accessible PDFs, then you should [watch the Silktide webinar "How to make PDFs accessible"](#).

In that webinar we teach you how to:

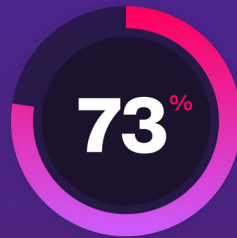
- Build an accessible PDF using Word
- Make an existing PDF accessible without the source document
- Make an existing PDF accessible with the source document



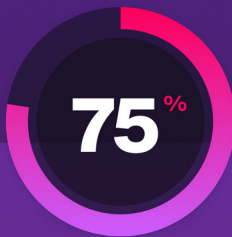
of websites fail to add bookmarks to long PDFs



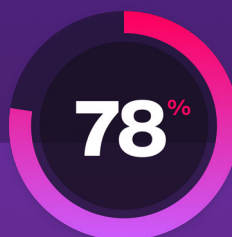
of websites fail to define a title for PDFs



of websites fail to specify a default language



of websites do not tag PDFs



of websites fail to specify a heading on PDFs



Check the readability

Generally speaking, you should aim for your content to be readable by a wide audience.

Easier to read text is more accessible, more usable, and gives a greater sense of fluency to the end-user.

Even websites with a highly educated audience benefit from easy-to-understand content. You want to make your readers feel smart, not stupid.

For sites concerned with accessibility, hard-to-read text can be a serious barrier to understanding. Readability has also been mentioned as a possible ranking variable for search engines, although evidence for this is mixed.

[WCAG 3.1.5](#) states that if complex information requiring a minimum of

a lower secondary level of education (7-9 years of schooling) is present, then supplementary content should be available.

Supplementary content may include:

- Summary paragraphs aimed at a lower reading age
- Illustrations of complex concepts in a simpler form
- An audio version of web content

Of course, it's not possible to get complex topics across to someone with only a few years of schooling, but that's not the point, nor is it realistic. WCAG simply suggests that you summarise complex information in a more readable way, alongside the technical content.





Specifying languages

On a more technical note and somewhat related, [you should set the default language of a page](#) so that screen readers can convert text into the correct speech patterns and inflections. It's a Level A requirement so you should consider this mandatory.

Specifying the language attribute also enables Google Translate to automatically offer suggested translation on page load.

32% of websites fail to specify a language on every page





Chapter 3

The ROI of Web Accessibility

The ROI of Web Accessibility

We've talked a lot about why you should create a better experience for everyone by making a more accessible website.

The benefits to most users are clear, but businesses often look for some indication of the ROI of investing in web accessibility.

Following are some statistics to illustrate.

Summary

- There are 14.1 million disabled people in the UK and this number is increasing ([source](#)).
- The spending power of UK families with at least one disabled person is an estimated £274 billion / year ([source](#)).
- 71% of disabled people abandon websites which present access barriers ([source](#)).
- Almost everyone experiences temporary or situational impairments (e.g. noise, poor lighting, distractions); these people benefit from web accessibility ([source](#)).
- 97.4% of the top 1 million websites have clear accessibility issues ([source](#)).
- 93% of disabled users don't report inaccessible websites to their owners ([source](#)).

Prevalence

In 2020, there were 14.1 million disabled people in the UK ([source: GOV.UK](#)).

- 8% of children are disabled
- 19% of working-age adults are disabled
- 46% of pension age adults are disabled

However, these figures drastically understate the number of people with a

temporary or situational impairment.

Consider that In the United States, about 26,000 people a year suffer from loss of upper extremities. But the number of people with similar temporary and situational impairments exceeds 20 million (source: Microsoft).



Other examples:

- Distracted users watching TV benefit from websites that impose a lower cognitive burden on them
- Users with mobility impairments, say riding a train, who benefit from decently sized buttons, and forgiving controls.
- Users unable to experience sound,

due to their environment: at work, or at a bar

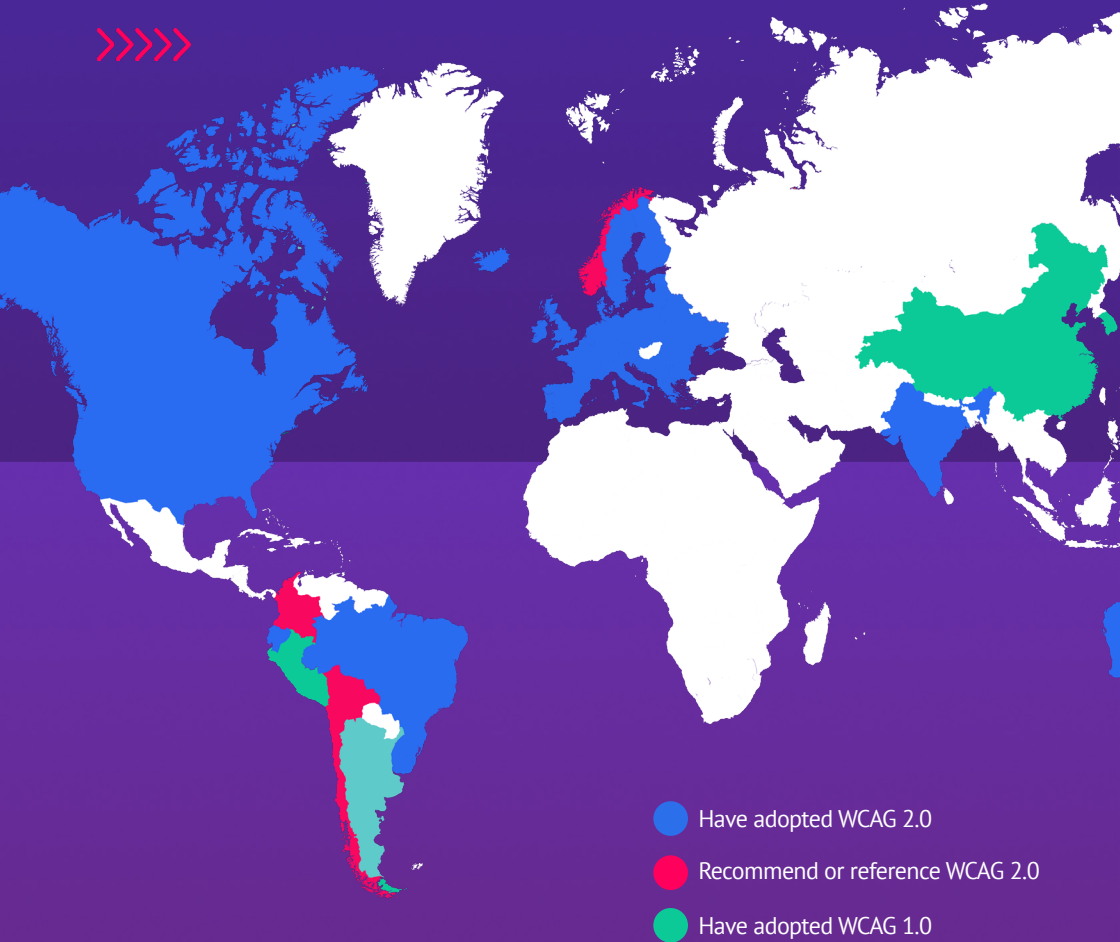
- Most users are now on mobile devices, and the idea of a user sitting down to experience your website on a computer, in a quiet room is now the exception, not the rule.



Business case

Legal risk

Legal risk varies enormously by country and between private and public sector organizations. Most countries have adopted legislation build around the [WCAG standard](#).



In the UK, the Disability Discrimination Act (1995) and Equality Act (2010) make it unlawful for a service provider to discriminate against a disabled person by refusing to provide the service or providing a worse service because of their disability.

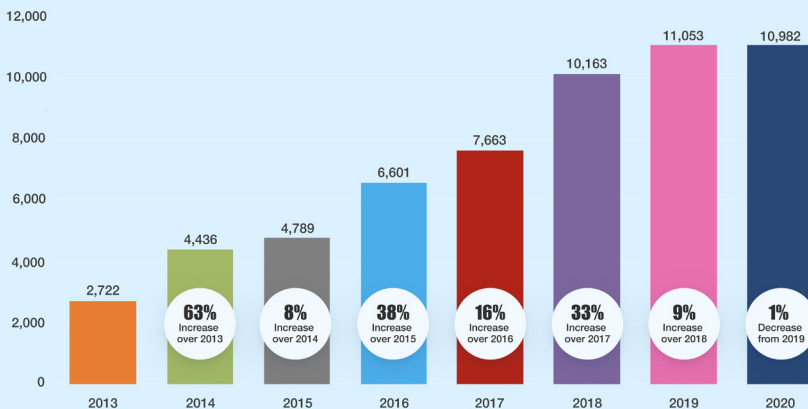
At the time of writing, although the UK public sector organizations have [clearly enforced standards](#), the legal risk for private organizations is relatively low. A customer would need to demonstrate a breach of the Equality Act in a County or

High court, which is difficult and expensive; although several cases have been initiated, all of these were settled out of court.

Conversely in the US, the [Americans with Disabilities Act \(ADA\)](#) and [Section 508 of the Rehabilitation Act \(1973\)](#) are widely used, and present a substantial risk of high compensation. In 2020, [10,982 ADA Title III lawsuits were filed in federal court](#), a number that was barely slowed by the COVID-19 pandemic.

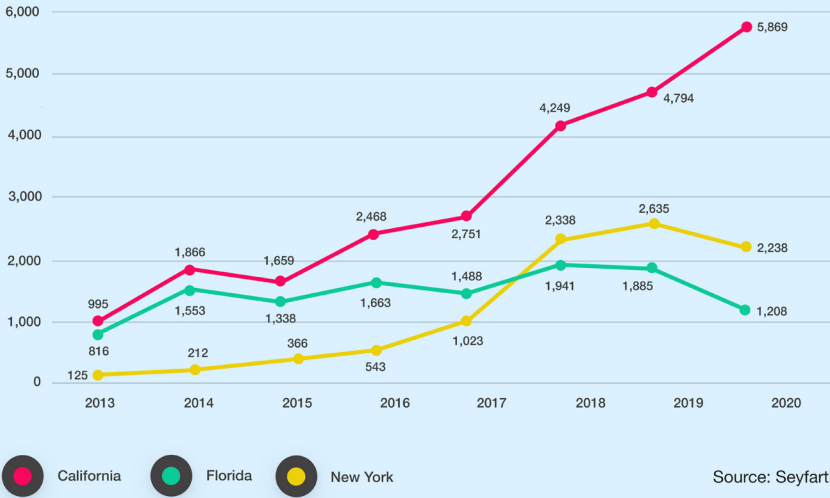
Total number of ADA title III federal lawsuits filed each year

January 1, 2013 - December 31, 2020



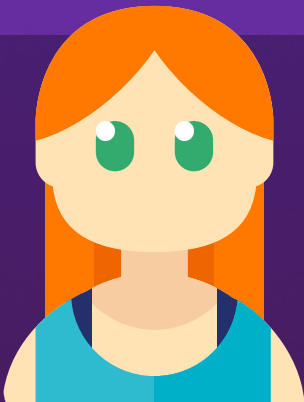
Source: Seyfarth.

California, New York, and Florida ADA Title III Federal Lawsuits (2013-2020)



These trends are accelerating drastically. During January 2021, 1,108 cases were filed – the most ever in a single month ([source](#)).

Globally, more legislation is being introduced over time, and although laws rarely change, the trend is pointing towards more regulation with stricter enforcement.



“Globally, more legislation is being introduced over time”

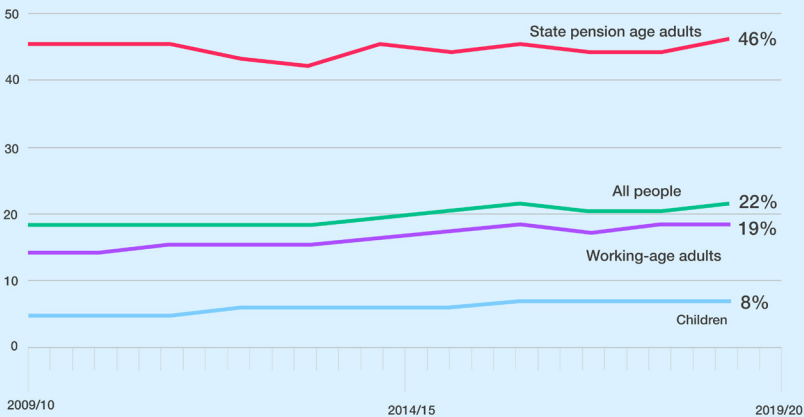


Commercial opportunity

Modern organizations are seeing a transition similar to what happened with diversity and inclusion: what was initially seen as a frustrating new challenge, is increasingly being recognized as an incredible opportunity.

In the UK, 22% of all people report a disability ([source: National Statistics](#)); for people of state pension age, this is 46%. These numbers are increasing over time:

Percentage of people



Source: National Statistics.



The total spending power of families with at least one disabled person in the UK is estimated at £274 billion a year ([source](#)). 71% of people with a disability will abandon a website that presents them with access barriers ([source](#)).

Add to this the much higher number of people who have a temporary or situational disability, who benefit from the improved user experience of an accessible website.

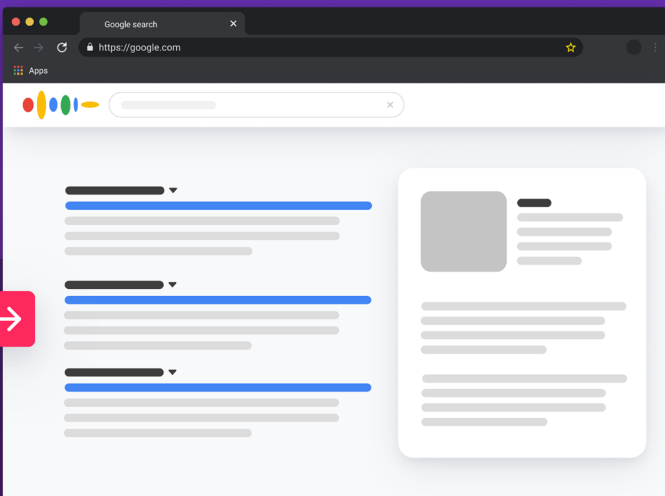
Considered as a facet of website usability, the commercial opportunity of improving accessibility is enormous. On average, e-commerce websites that redesign for better usability [double their sales](#); accessibility is a component of these gains.

Many components of accessibility are closely aligned with Search Engine Optimization (SEO), including the use of alternative text, headings, landmarks, semantics, captions, page titles, languages, code quality, and link size.

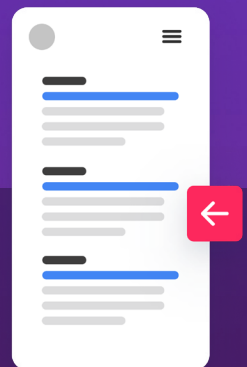
Improving accessibility can therefore improve search engine placement and traffic ([source: W3C](#)).

An accessible mobile experience is both a key ranking factor ([source: Google](#)) and a requirement of modern accessibility standards (WCAG 2.1 and later). Websites rank differently in Google on mobile and desktop devices, depending on how accessible they are on mobile:

Desktop



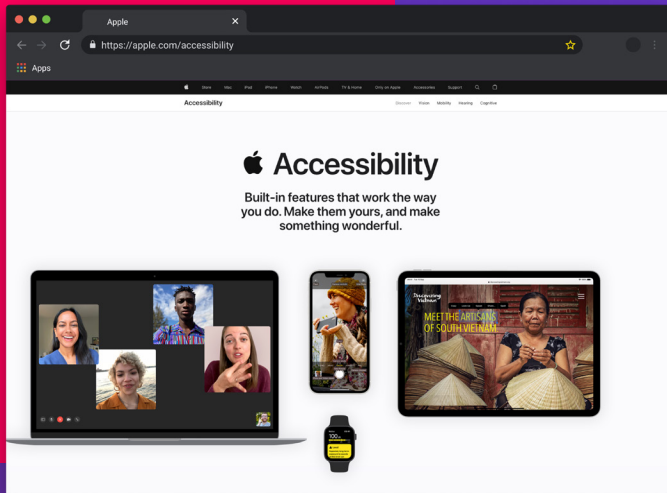
Mobile



Brand and reputation

Organizations who embrace accessibility can benefit from the positive press that accessibility attracts, such as [BBC's coverage of Ticketmaster](#).

The reputational damage from being inaccessible is hard to quantify but is present. Inaccessible websites are by definition, delivering an inferior user experience, which means lower customer retention, and fewer recommendations.



Consider that the largest two companies in the world, as measured by market capitalization at the time of writing, are [Microsoft](#) and [Apple](#). Both invest heavily into world-leading accessibility, which is prevalent throughout their products, services, training, brands, and marketing. Apple has had public accessibility guidelines available for its operating system since 1987.

Costs and solutions

The cost of making a website accessible is generally a small fraction of the overall cost of the website as a whole. This fixed cost is almost always a negligible fraction of the expected returns.

Once the correct processes, tools, and training are in place, the overhead for keeping a website accessible drops significantly, so long as it's done properly.

A further concern can be how accessibility can reduce the responsiveness of your team. By introducing additional processes, accessibility can slow down the rate at which websites can be developed or updated.

Silktide is designed to help organizations [identify and remediate accessibility issues quickly](#), and at scale. Wherever possible, issues are identified automatically, and Silktide helps teams share, triage, and understand how to fix them in the most efficient way.

By providing clear and transparent scoring of the different areas of your web estate, Silktide helps to align your teams. They can view their progress easily, regularly, and across business departments. Together, scores help incentivize team members to care about and excel at accessibility.

Further reading

- [The Business Case for Digital Accessibility](#) (W3C)
- [Accessibility of top 1 million websites](#) (WebAIM)
- [Assessing the online shopping experience of customers with disabilities, and the costs to business of ignoring them](#) (Enterprise rent-a-car)
- [We analyzed 6,554 websites for accessibility issues and here are the results](#) (Silktide)





Chapter 4

Conclusion

Conclusion

Web accessibility problems affect everybody, and often, the solution to a single issue creates a better experience for all users.

Fixing poor web accessibility should be seen not as an expensive, time-consuming endeavour, but as an opportunity to improve.

Not only will you reduce your risk of

litigation, you'll improve your SEO, create a better experience for all your users, and have a positive impact on your brand perception.

While the topic of web accessibility can seem overwhelming, it's possible to make good headway in fixing common issues relatively quickly, as long as you know where to look.

Accessibility and the law

Depending on your location and your sector, you may be subject to laws requiring that your website is sufficiently accessible to certain recognised standards.

The consequences for non-compliance may be severe.

In Europe, the EU Web Accessibility Directive applies to all public sector websites, and states that these websites must be accessible to WCAG 2.1 standards.

In the U.S., the Americans with

Disabilities Act extends to 'places of public accommodation', and this includes (depending on your location and sector) your website.

U.S. organizations with poor web accessibility leave themselves vulnerable to litigation, with hefty fines levied against large companies in recent years. The number of ADA web accessibility lawsuits filed [is increasing](#).



Finding and fixing web accessibility issues

A single issue may affect every page on your website (a good example of this is a missing Skip to Content link), so by fixing that you'll have a big impact on both usability and accessibility.

We identified many common accessibility issues in websites around the world in the [Silktide Index](#). The data clearly shows that no sector or location is immune to web accessibility issues.

While creating and managing an accessible website might seem like a monumental challenge, given the general lack of awareness, lack of resources, or inability to find current accessibility problems, with the right tools it's possible to create a more accessible website relatively quickly.

For example, in just five weeks the Royal Cornwall Hospitals NHS Trust, a UK-based healthcare provider, [increased their Silktide](#)

[accessibility score from 35 to 99](#).

They were able to do this for a number of reasons. Firstly, Silktide was able to find accessibility issues automatically across their websites, including mobile accessibility issues.

Secondly, Silktide helped educate their web team in accessibility by including information and suggestions on how to fix the issues in plain language.

Finally, Silktide's platform improved the team's workflow by listing out all the issues in one place, and allowed the team to segment their websites and assign work to individuals.

This created a much more efficient way of working and allowed the team to rapidly improve accessibility by working on the higher priority issues (those identified as having the most impact) first.





Next steps

The best place to begin is by starting a conversation about web accessibility in your organization.

We've supported many organizations with accessibility, and work closely with the public sector.

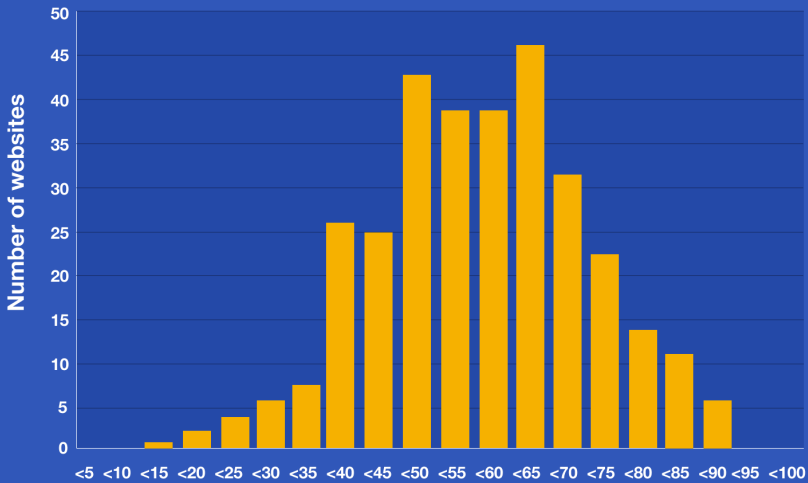
Over the last few years, government agencies and healthcare providers have

worked closely with SilkTide to improve their accessibility. The figures prove that over time, these sectors have massively improved.

Conversely, the private sector is still lagging behind somewhat. You can see a comparison in average accessibility scores across two sample sectors here.

How accessibility scores vary across FTSE

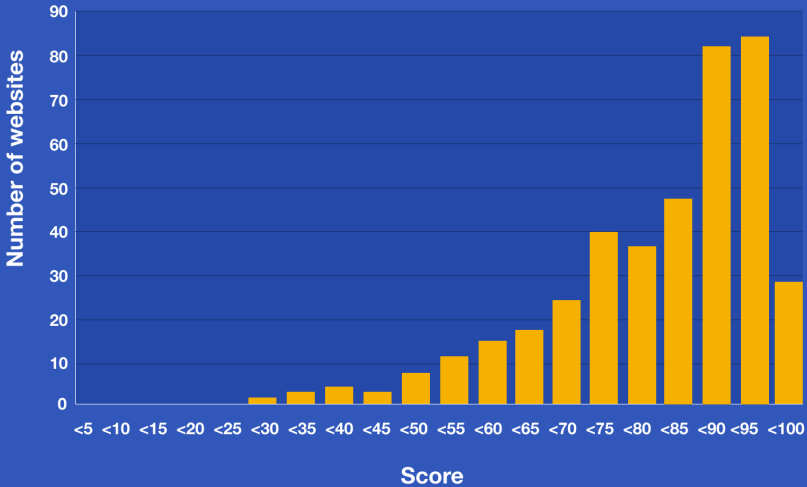
This is the number of FTSE websites with each accessibility score.





How accessibility scores vary across UK Councils

This is the number of UK Council websites with each accessibility score.



Our (January 2022) data shows a marked difference in average accessibility scores between the public and private sector.

Most of the UK FTSE websites score between 40 and 70, while most UK Councils score between 70 and 100.

This is likely to be more of a case of ignorance on the part of FTSE businesses rather than wilfully neglecting web accessibility.

Indeed, in February 2020 when Silktide first tested UK Councils the average score was ten points lower across the board.

Over time, we hope to see average scores improve across all sectors, whether private or public, as more organizations start their accessibility conversations.



About Silktide

We're the web intelligence company helping businesses and organizations make their websites better.

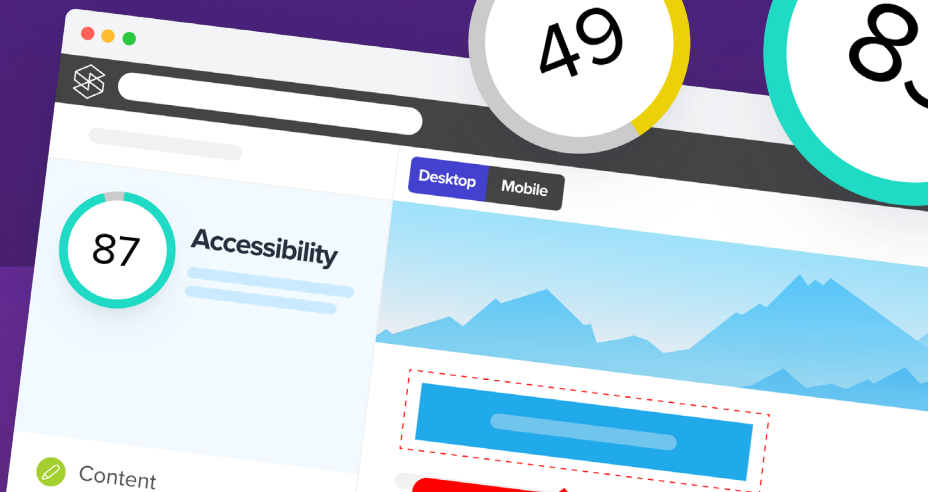
Founded in 2001, Silktide helps thousands of customers analyze tens of millions of websites every year. Our users are found in over 900 cities around the globe.

We cover almost every aspect of an organization's web presence, including content, accessibility, marketing, mobile,

social, speed, and legal compliance.

Our clients are located around the world, and we help large organizations in both the public and private sectors improve their websites.

Our mission is to make the web a better place.



silktide

silktide.com