

ITI0209: User Interfaces

12. Accessibility

Martin Verrev

Spring 2026

When UX doesn't consider ALL users, shouldn't it be known as "Some User Experience" or... SUX?"

Billy Gregory, Senior Accessibility Engineer, Ubisoft.

<https://x.com/thebillygregory/status/552466012713783297>

Accessibility (a11y) is the practice of making your websites usable by as many people as possible. We usually think of this as being about people with disabilities. But making sites accessible also benefits others - those using mobile devices, or those with slow network connections.

Accessible design helps users with different levels of ability. Many users struggle to interact with digital products because of visual, auditory, or motor impairments.

Inclusive design aims to welcome users from diverse backgrounds and circumstances, ensuring that factors such as race, gender, and religion do not hinder their digital experiences.





Your Name Wasn't Approved

It looks like that name violates our name standards.

You can enter an updated name again in 1 minute. To make sure the updated name complies with our policies, please [read more about what names are allowed on Facebook](#).

Try Again

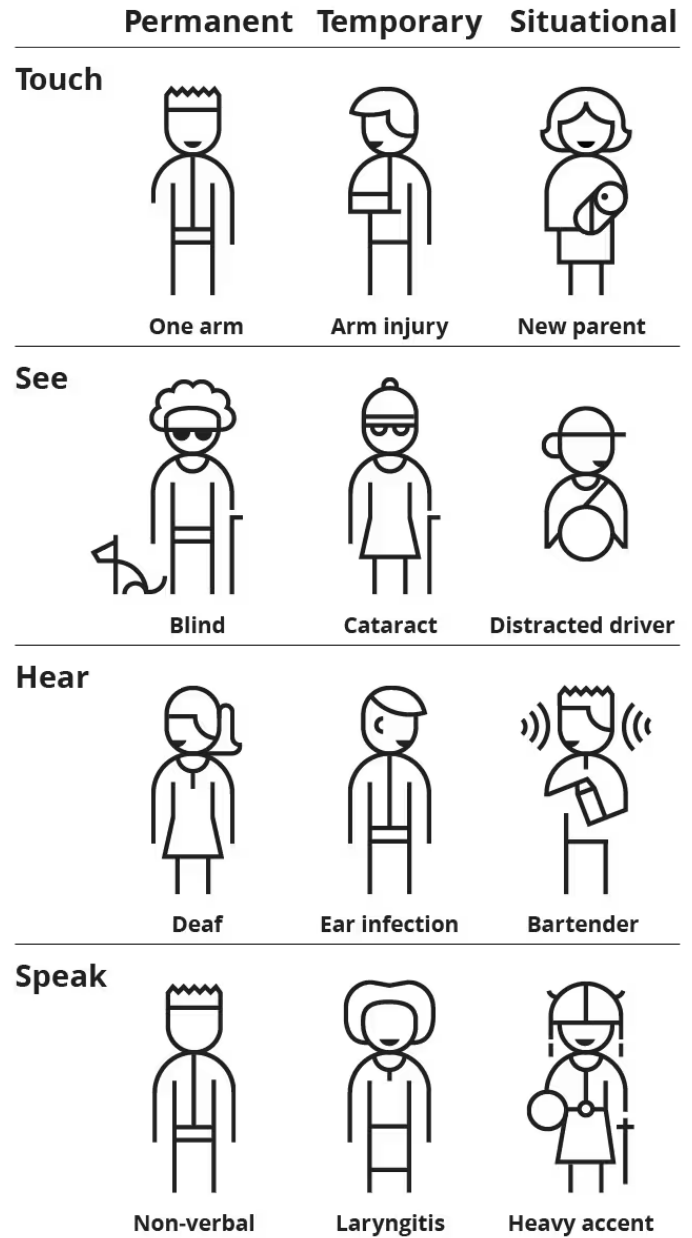
In October 2014, an individual named *Shane Creepingbear*, a member of the Kiowa Tribe of Oklahoma, tweeted that he was kicked off of Facebook due to claims that he was using a fake name, even though this was not the case.

See: <https://www.vice.com/en/article/facebook-is-still-making-it-difficult-for-native-americans-to-use-their-real-names-602/>

Building accessible sites benefit everyone

- **Semantic HTML** improves accessibility, and also improves SEO, making your site more findable.
- **Caring** about accessibility demonstrates good ethics and morals, which improves your public image.
- Other **good practices** that improve accessibility also make your site more usable by others - such as mobile phone users on low network speed. In fact, everyone can benefit from many such improvements.
- Accessibility is also the **law** in EU and UK

All of us become temporarily or situationally disabled sometimes



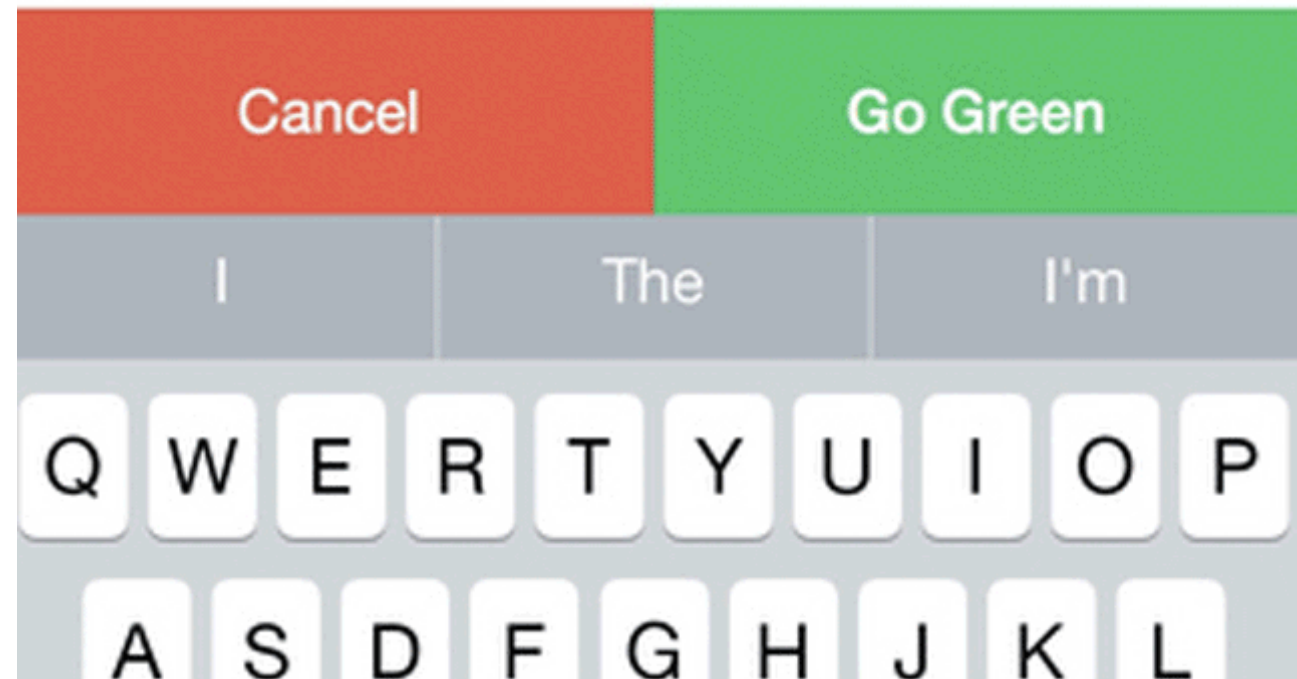
Coping: Visual Impairments

Blindness, low-level vision, and color blindness.

Use: screen magnifiers or software zoom. Some users rely on screen readers.

See also: <https://www.smashingmagazine.com/2017/10/nailing-accessibility-design/>

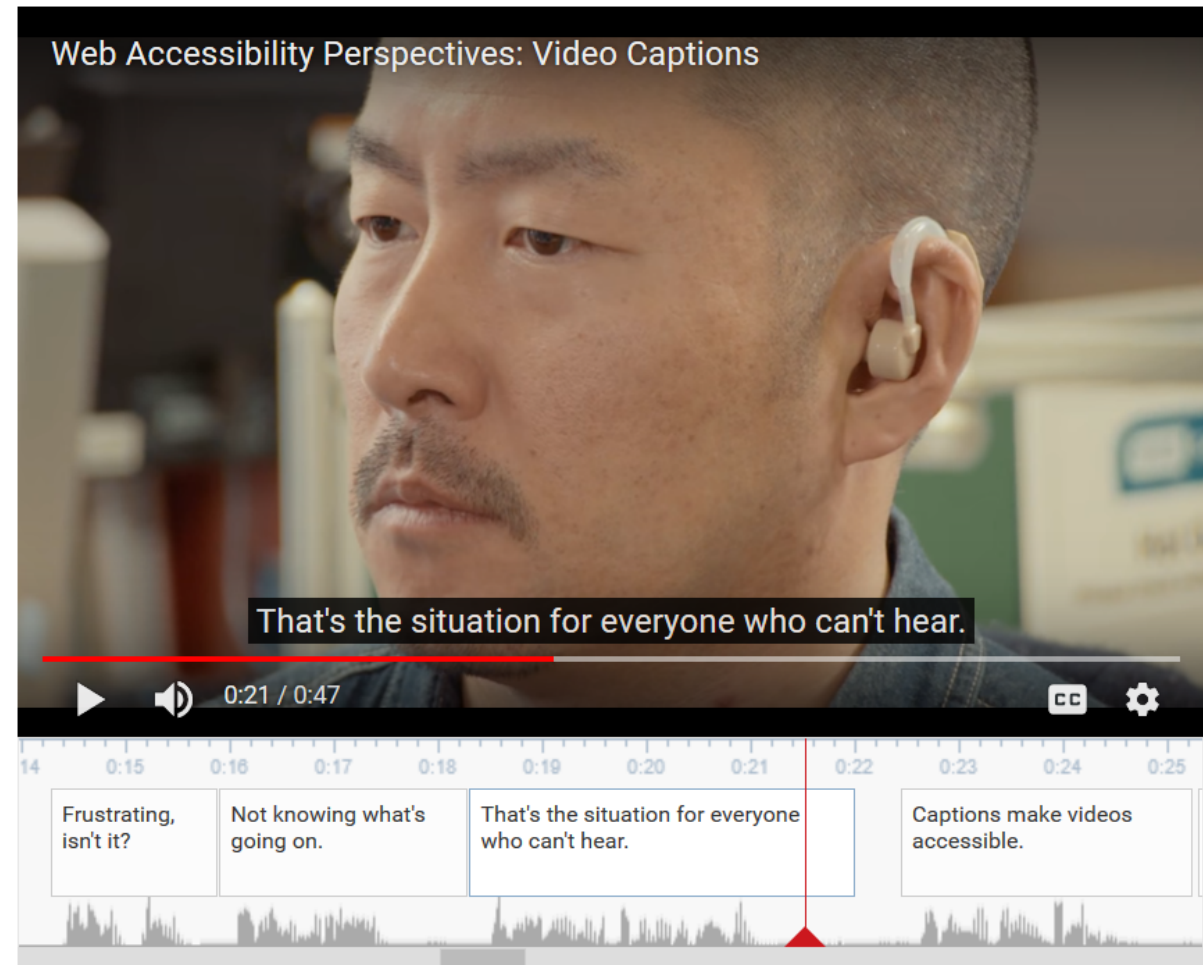
ramen 🍜 🙌



Coping: Hearing Impairments

Low hearing levels or no hearing at all. No specific equipment exists for web use. There are, however, specific techniques for providing textual alternatives to audio content which range from simple text transcripts to video captions.

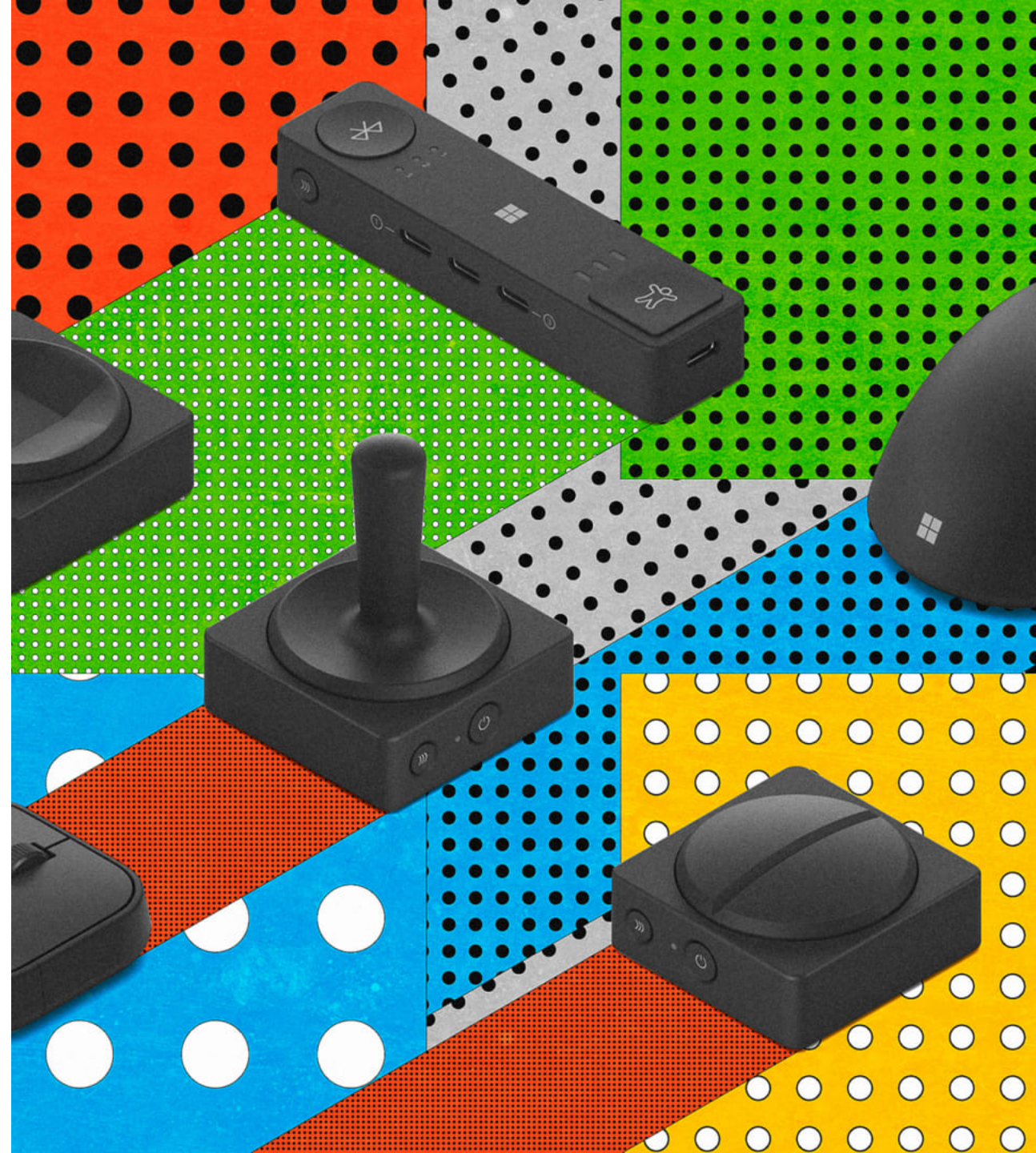
See also: <https://app.uxcel.com/courses/design-accessibility/designing-for-hearing-impairments-463>



Coping: Mobility Impairments

Such users may not have a mouse
The way this usually affects web development work is the requirement that controls be accessible by the keyboard

See also: <https://blinkux.com/ideas/designing-for-mobility-impairments>



Coping: Cognitive Impairments

Cognitive impairment refers to a broad range of disabilities, from people with intellectual disabilities to everyone that has sometimes hard time thinking and remembering.

A good foundation of accessibility for people with cognitive impairments includes:

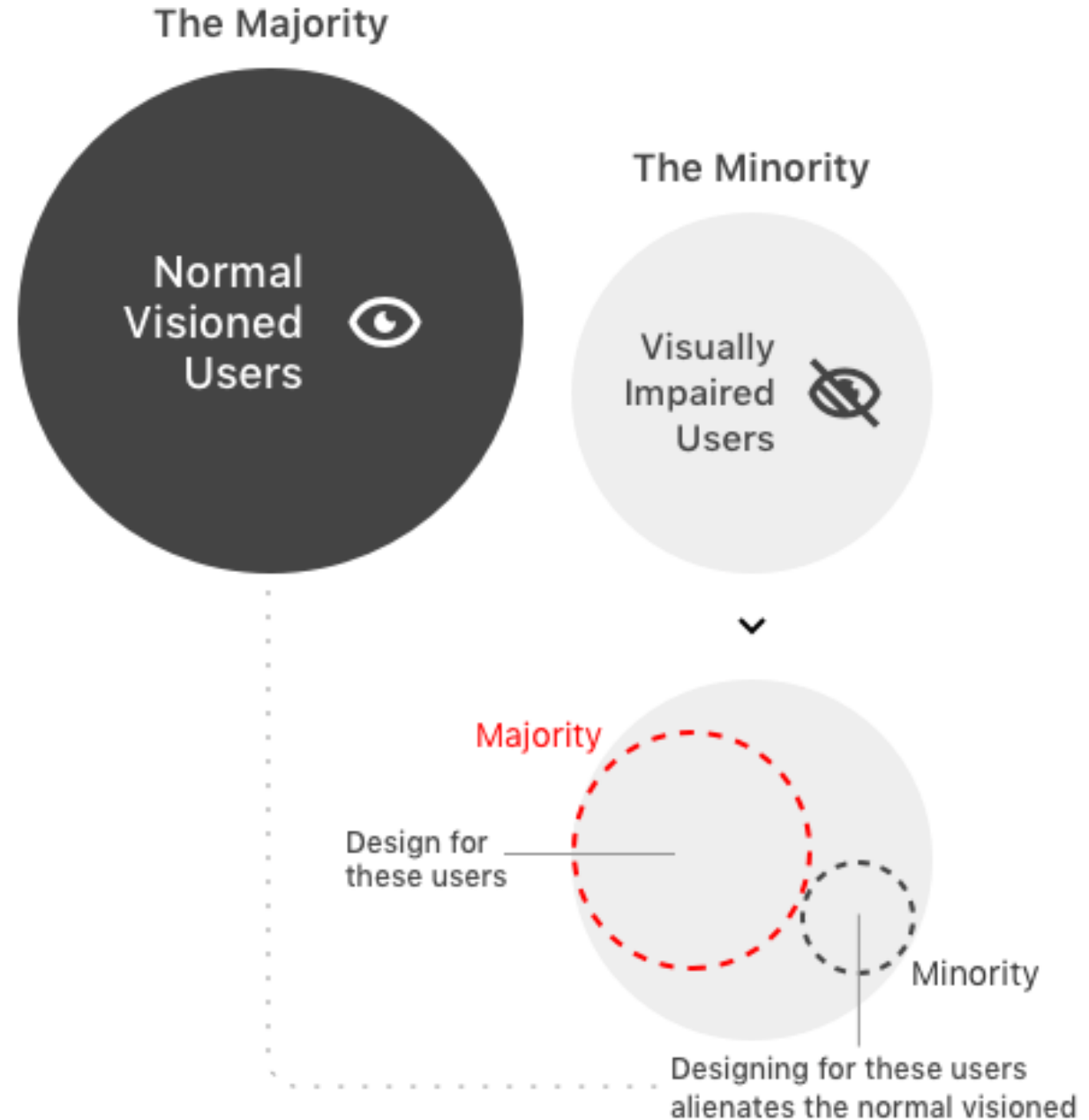
- Easily understood content
- Focusing attention on important content.
- Consistent webpage layout and navigation.
- Dividing processes into logical, essential steps with progress indicators.
- Website authentication as easy as possible without compromising security.
- Making forms easy to complete, such as with clear error messages and simple error recovery.

Working Towards Accessibility

- By default, **HTML is accessible**, if used correctly. Web accessibility involves ensuring that content remains accessible, regardless of who and how the web is accessed.
- **Use common sense.** If you are working on a gallery website showing interesting 3D art, it would be unreasonable to expect every piece of art to be perfectly accessible to visually impaired people, given that it is an entirely visual medium.
- **Be realistic.** "100% accessibility" is an unobtainable ideal

Aesthetic Accessibility Paradox

In general, the more accessible an interface is, the less aesthetic appeal it will have. Highly accessible interfaces are easier on the visually impaired, but harsher on the eyes of the normal-visioned. On the flip side, highly aesthetic interfaces are easier on the eyes of the normal-visioned, but harsher on the visually impaired.



Sign Up

First name

Last name

Email

Password

Sign up

Sign Up

First name

Last name

Email

Password

Sign up

Sign Up


First name

Last name

Email

Password

Sign up

 Too Accessible

Accessible  Aesthetic

Too Aesthetic 

High accessibility/ high aesthetic is possible

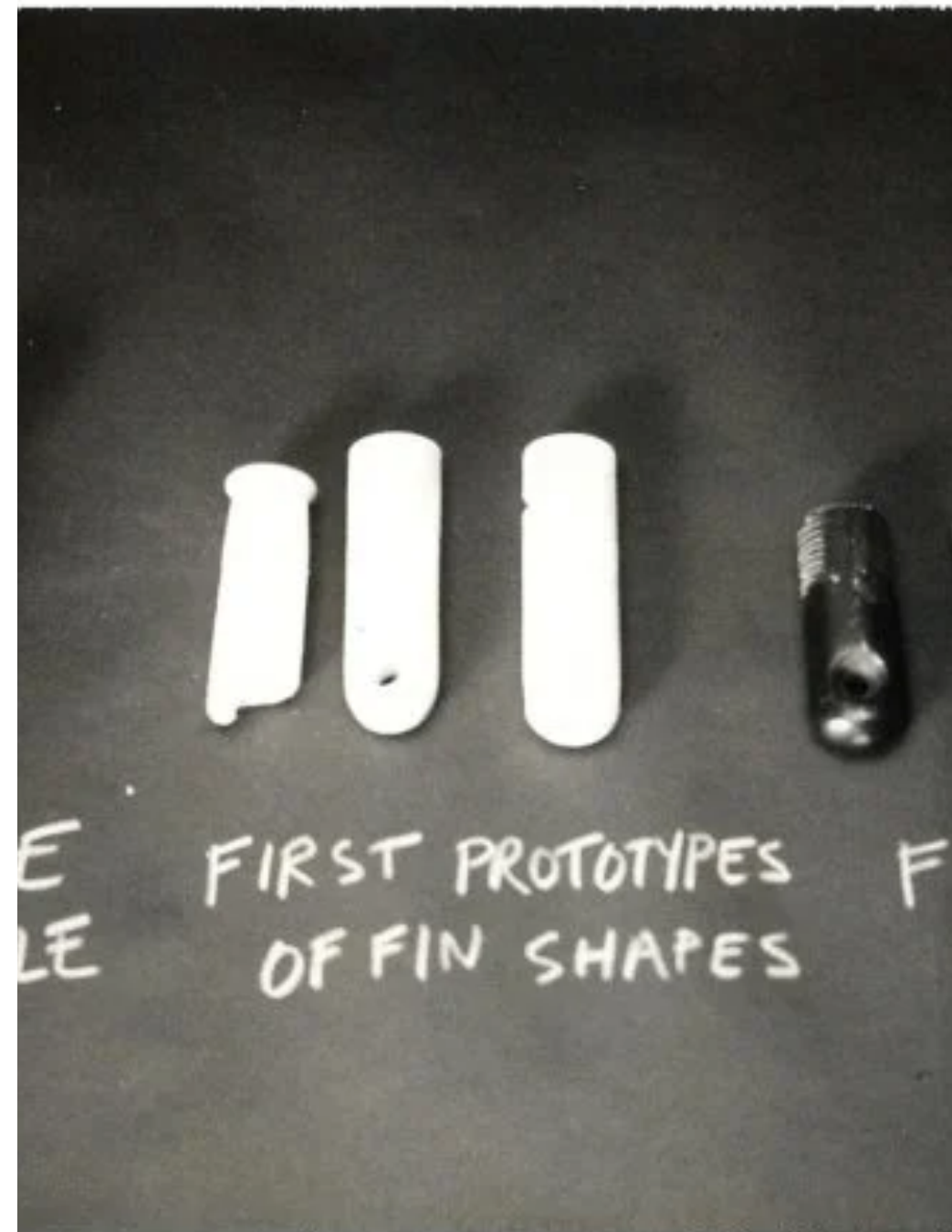
In fact, a well-designed website can be both visually appealing and highly accessible.

It does not mean accessibility has to come at the expense of aesthetics. Rather, the designers must be creative and innovative in finding ways to make their products both visually stunning and highly accessible.

See:

<https://www.rew.works/logbook/utility-empathy>

See: OXO Good Grips Peeler: <https://www.moma.org/collection/works/3758>



Common-Sense Accessibility

Semantic structure

The most important quick win in semantic HTML is to use a structure of headings and paragraphs for your content; this is because screen reader users tend to use the headings of a document as signposts to find the content they need more quickly.

Using native keyboard accessibility

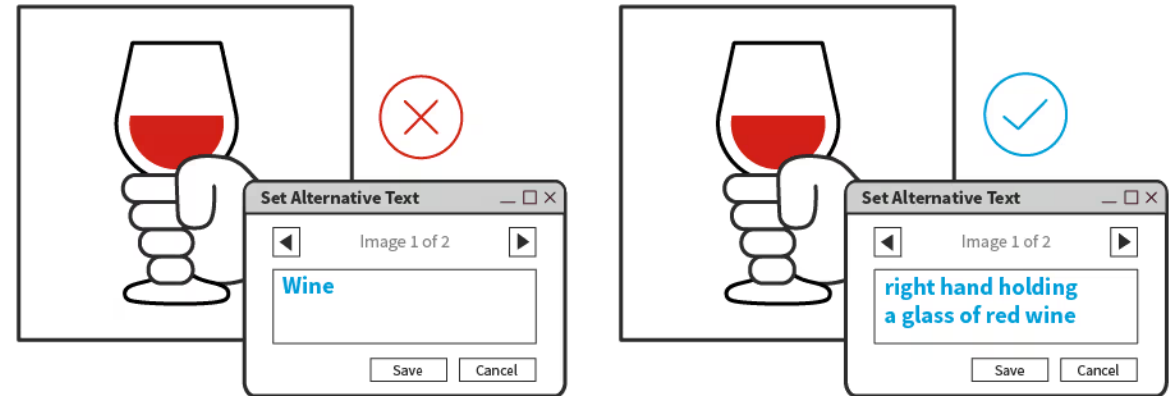
Certain HTML features can be selected using only the keyboard — this is default behavior, available since the early days of the web. The elements that have this capability are the common ones that allow user to interact with web pages, namely links, `<button>` s, and form elements like `<input>` .

Common-Sense Accessibility

Text Alternatives

They are very important. The simplest text alternative available is the humble `alt` attribute, which we should include on all images. This should contain a description of the image that successfully conveys its meaning and content - to be picked up by a screenreader and read out to the user.

Alt Text for SEO



Interaction Design Foundation
interaction-design.org



Common-Sense Accessibility

Element relationships and context

There are certain features and best practices in HTML designed to provide context and relationships between elements where none otherwise exists. The three most common examples are links, form labels, and data tables.

Common-Sense Accessibility

Color and color contrast

When choosing a color scheme for your website, you should make sure that the text (foreground) color contrasts well with the background color. Your design might look cool, but it is no good if people with visual impairments like color blindness can't read your content.

- Use the contrast of at least 4.5:1 for normal text and 3:1 for large text
- Don't rely on color exclusively - some color combinations are easily confused by colorblind users.

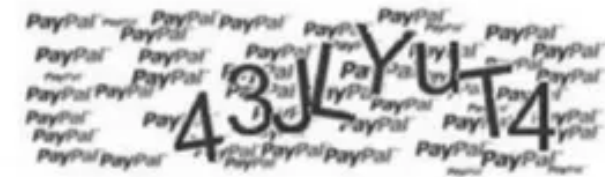
Use color to highlight or complement what is already visible.

How many fields are in an error state?

Sign up for PayPal, it's free.

Personal Account

Shop, receive money, or just pay someone back for lunch. All without sharing your payment info.



How many fields are in an error state?

Sign up for PayPal, it's free.

Personal Account

Shop, receive money, or just pay someone back for lunch. All without sharing your payment info.

Common-Sense Accessibility

Simple functionality

Generally simple functionality should work with just the HTML in place — JavaScript should only be used to enhance functionality, not build it in entirely. Good uses of JavaScript include client-side form validation or custom controls for HTML5 `<video>`.

Common-Sense Accessibility

Using clear language

In general, you should use clear language that is not overly complex and doesn't use unnecessary jargon or slang terms. This not only benefits people with cognitive or other disabilities; it benefits everyone!

- Don't use dashes if you can avoid it. Instead of writing `5–7`, write `5 to 7`.
- Expand abbreviations: instead of writing `Jan`, write `January`.
- Expand acronyms (or at least once or twice). Instead of writing `HTML` in the first instance, write `Hypertext Markup Language`.

Common-Sense Accessibility

Meaningful Text Labels

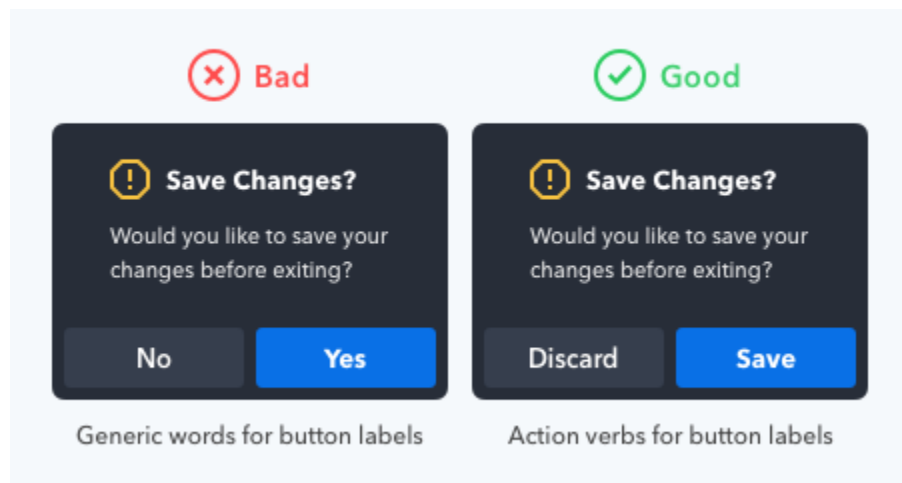
UI control text labels are very useful to all users, but getting them right is particularly important to users with disabilities.

You should make sure that your button and link text labels are understandable and distinctive. Don't just use "Click here" for your labels, as screen reader users sometimes get up a list of buttons and form controls.

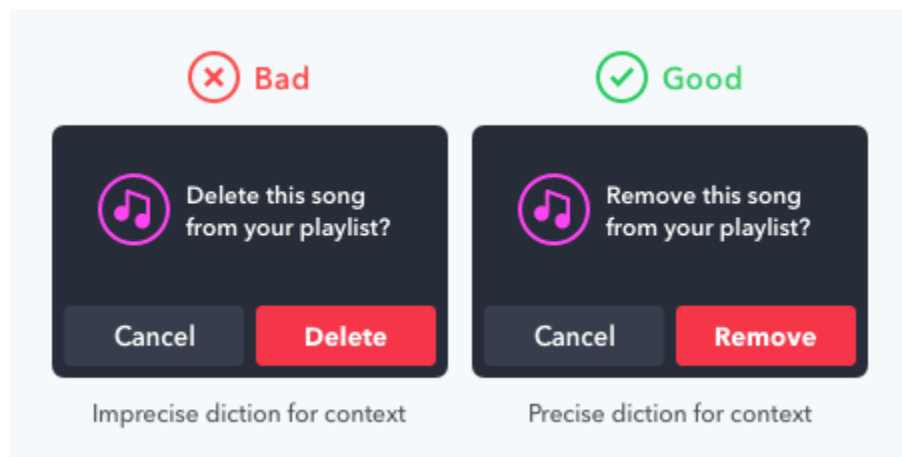
Make sure your labels make sense out of context, read on their own, as well as in the context of the paragraph they are in.

Common-Sense Accessibility: Meaningful Text Labels

1. Use Action Verbs

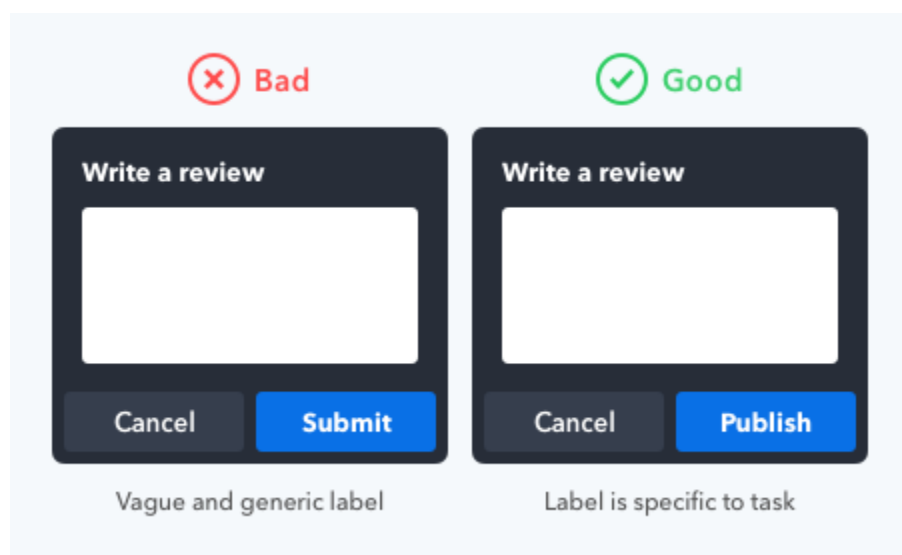


2. Use Precise Diction.

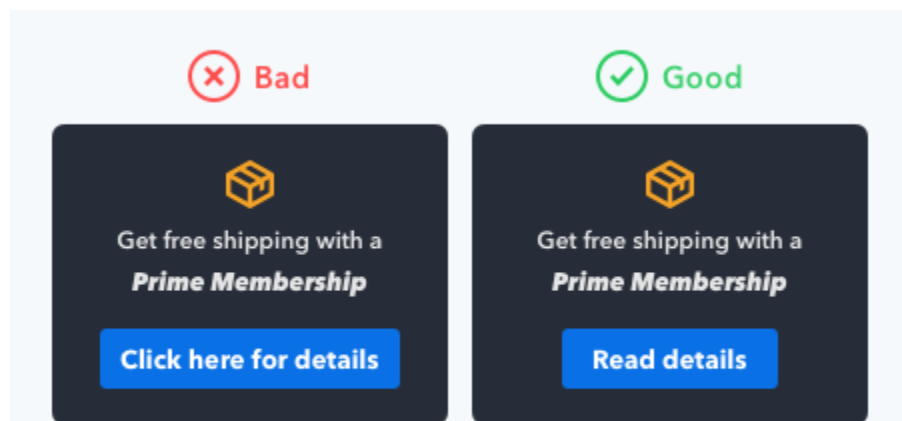


Common-Sense Accessibility: Meaningful Text Labels

3. Use Task-Specific Language



4. Use the Active Imperative Form



Extra Considerations for Mobile

There are some exceptions that need special consideration for mobile; the main ones are:

- **Control mechanisms** — Make sure interface controls such as buttons are accessible on mobiles (i.e., mainly touchscreen), as well as desktops/laptops (mainly mouse/keyboard).
- **User input** — Make user input requirements as painless as possible on mobile (e.g., in forms, keep typing to a minimum).
- **Responsive design** — Make sure layouts work on mobile, conserve image download sizes, and think about the provision of images for high-resolution screens.

The Web Content Accessibility Guidelines (WCAG) are part of a series of web accessibility guidelines published by the Web Accessibility Initiative (WAI) of the World Wide Web Consortium

<https://www.w3.org/TR/WCAG22/>

Levels of WAI Guidelines

A: Essential

If this isn't met, assistive technology may not be able to read, understand, or fully operate the page or view.

AA: Ideal Support

Required for multiple government and public body websites.

AAA: Specialized Support

Typically reserved for parts of websites and web apps that serve a specialized audience.

Going Further: WAI-ARIA

WAI-ARIA (Web Accessibility Initiative - Accessible Rich Internet Applications) is a set of attributes that define ways to make web content and web applications - especially those developed with JavaScript, more accessible.

See also: <https://www.davidmacd.com/blog/wai-aria-accessibility-for-average-web-developers.html>

Although ARIA allows us to modify the accessibility tree for any element on the page, that is the only thing it changes. ARIA **doesn't change** any of the element's inherent behavior.

Features WAI-ARIA

- **Roles** - Define what an element is or does. any of these are so-called landmark roles, which largely duplicate the semantic value of HTML5 structural elements e.g. `role="navigation"`
- **Properties** - These define properties of elements, which can be used to give them extra meaning or semantics. For example, `aria-required="true"` specifies that a form input needs to be filled in order to be valid. Properties do not change throughout the lifecycle of an app.
- **States** — Special properties that define the current conditions of elements, such as `aria-disabled="true"`, which specifies to a screenreader that a form input is currently disabled. States can change, generally programmatically via JavaScript.

When To Use

Only use when necessary, you should *always* use native HTML features. See: <https://www.w3.org/TR/using-aria/#rule1>

1. **Signposts/Landmarks:** ARIA's role attribute values act as landmarks that either replicate the semantics of HTML5 elements (e.g. `<nav>`), or go beyond to provide signposts to different functional areas, e.g search, tabgroup, tab, listbox, etc.
2. **Dynamic content updates:** We can use `aria-live` to inform screenreader users when an area of content is updated via AJAX.
3. **Enhancing keyboard accessibility:** for JS generated elements (using `tabindex`).
4. **Accessibility of non-semantic controls:** for instance `role="button"` for JS generated non-semantic controls.

Common Mistakes

- Labelling mismatching elements ``
- Using An `aria-labelledby` attribute that references an ID that does not exist,
- Not moving the focus to modal when it opens.
- Adding roles that duplicate HTML, e.g. `<button role="button">`
- Adding `tabindex=0` to every element.
- Using `role="menu"` for navigation.

Validating Accessibility



web accessibility evaluation tool

<https://wave.webaim.org/>

Tools

- **Screen Reader** for your browser.
- **Colorblinding**. Extension that simulates the website as a color vision impaired person would see.
- **Web Developer Toolbar**. Universal developer utilities and checks.
- **Color Contrast Checker**. <https://colors.co/contrast-checker>
- **Accessibility Insights** browser extension. <https://accessibilityinsights.io>
- **Pa11y**. <https://pa11y.org>. Automated accessibility checker.
- **Whocanuse.com**. <https://www.whocanuse.com>. Different types of vision impairment checking.

See also: Start Testing for Web Accessibility. <https://a11y.coffee/start-testing/>

A Quick Summary: Dos and don'ts on designing for accessibility

<h3>Designing for users on the autistic spectrum</h3> <table border="1"> <thead> <tr> <th>Do...</th> <th>Don't...</th> </tr> </thead> <tbody> <tr> <td>use simple colours</td> <td>use bright contrasting colours</td> </tr> <tr> <td>write in plain English</td> <td>use figures of speech and idioms</td> </tr> <tr> <td>use simple sentences and bullets</td> <td>create a wall of text</td> </tr> <tr> <td>make buttons descriptive</td> <td>make buttons vague and unpredictable</td> </tr> <tr> <td>build simple and consistent layouts</td> <td>build complex and cluttered layouts</td> </tr> </tbody> </table>	Do...	Don't...	use simple colours	use bright contrasting colours	write in plain English	use figures of speech and idioms	use simple sentences and bullets	create a wall of text	make buttons descriptive	make buttons vague and unpredictable	build simple and consistent layouts	build complex and cluttered layouts	<h3>Designing for users of screen readers</h3> <table border="1"> <thead> <tr> <th>Do...</th> <th>Don't...</th> </tr> </thead> <tbody> <tr> <td>describe images and provide transcripts for video</td> <td>only show information in an image or video</td> </tr> <tr> <td>follow a linear, logical layout</td> <td>spread content all over a page</td> </tr> <tr> <td>structure content using HTML5</td> <td>rely on text size and placement for structure</td> </tr> <tr> <td>build for keyboard use only</td> <td>force mouse or screen use</td> </tr> <tr> <td>write descriptive links and headings</td> <td>write uninformative links and headings</td> </tr> </tbody> </table>	Do...	Don't...	describe images and provide transcripts for video	only show information in an image or video	follow a linear, logical layout	spread content all over a page	structure content using HTML5	rely on text size and placement for structure	build for keyboard use only	force mouse or screen use	write descriptive links and headings	write uninformative links and headings	<h3>Designing for users with low vision</h3> <table border="1"> <thead> <tr> <th>Do...</th> <th>Don't...</th> </tr> </thead> <tbody> <tr> <td>use good colour contrasts and a readable font size</td> <td>use low colour contrasts and small font size</td> </tr> <tr> <td>publish all information on web pages</td> <td>bury information in downloads</td> </tr> <tr> <td>use a combination of colour, shapes and text</td> <td>only use colour to convey meaning</td> </tr> <tr> <td>follow a linear, logical layout</td> <td>spread content all over a page</td> </tr> <tr> <td>put buttons and notifications in context</td> <td>separate actions from their context</td> </tr> </tbody> </table>	Do...	Don't...	use good colour contrasts and a readable font size	use low colour contrasts and small font size	publish all information on web pages	bury information in downloads	use a combination of colour, shapes and text	only use colour to convey meaning	follow a linear, logical layout	spread content all over a page	put buttons and notifications in context	separate actions from their context
Do...	Don't...																																					
use simple colours	use bright contrasting colours																																					
write in plain English	use figures of speech and idioms																																					
use simple sentences and bullets	create a wall of text																																					
make buttons descriptive	make buttons vague and unpredictable																																					
build simple and consistent layouts	build complex and cluttered layouts																																					
Do...	Don't...																																					
describe images and provide transcripts for video	only show information in an image or video																																					
follow a linear, logical layout	spread content all over a page																																					
structure content using HTML5	rely on text size and placement for structure																																					
build for keyboard use only	force mouse or screen use																																					
write descriptive links and headings	write uninformative links and headings																																					
Do...	Don't...																																					
use good colour contrasts and a readable font size	use low colour contrasts and small font size																																					
publish all information on web pages	bury information in downloads																																					
use a combination of colour, shapes and text	only use colour to convey meaning																																					
follow a linear, logical layout	spread content all over a page																																					
put buttons and notifications in context	separate actions from their context																																					
<h3>Designing for users with physical or motor disabilities</h3> <table border="1"> <thead> <tr> <th>Do...</th> <th>Don't...</th> </tr> </thead> <tbody> <tr> <td>make large clickable actions</td> <td>demand precision</td> </tr> <tr> <td>give form fields space</td> <td>bunch interactions together</td> </tr> <tr> <td>design for keyboard or speech only use</td> <td>make dynamic content that requires a lot of mouse movement</td> </tr> <tr> <td>design with mobile and touchscreen in mind</td> <td>have short time out windows</td> </tr> <tr> <td>provide shortcuts</td> <td>ire users with lots of typing and scrolling</td> </tr> </tbody> </table>	Do...	Don't...	make large clickable actions	demand precision	give form fields space	bunch interactions together	design for keyboard or speech only use	make dynamic content that requires a lot of mouse movement	design with mobile and touchscreen in mind	have short time out windows	provide shortcuts	ire users with lots of typing and scrolling	<h3>Designing for users who are Deaf or hard of hearing</h3> <table border="1"> <thead> <tr> <th>Do...</th> <th>Don't...</th> </tr> </thead> <tbody> <tr> <td>write in plain English</td> <td>use complicated words or figures of speech</td> </tr> <tr> <td>use subtitles or provide transcripts for videos</td> <td>put content in audio or video only</td> </tr> <tr> <td>use a linear, logical layout</td> <td>make complex layouts and menus</td> </tr> <tr> <td>break up content with sub-headings, images and videos</td> <td>make users read long blocks of content</td> </tr> <tr> <td>let users request an interpreter for appointments</td> <td>don't make telephone the only means of contact with users</td> </tr> </tbody> </table>	Do...	Don't...	write in plain English	use complicated words or figures of speech	use subtitles or provide transcripts for videos	put content in audio or video only	use a linear, logical layout	make complex layouts and menus	break up content with sub-headings, images and videos	make users read long blocks of content	let users request an interpreter for appointments	don't make telephone the only means of contact with users	<h3>Designing for users with dyslexia</h3> <table border="1"> <thead> <tr> <th>Do...</th> <th>Don't...</th> </tr> </thead> <tbody> <tr> <td>use images and diagrams to support text</td> <td>use large blocks of heavy text</td> </tr> <tr> <td>align text to the left and keep a consistent layout</td> <td>underline words, use italics or write in capitals</td> </tr> <tr> <td>consider producing materials in other formats (for example, audio or video)</td> <td>force users to remember things from previous pages - give reminders and prompts</td> </tr> <tr> <td>keep content short, clear and simple</td> <td>rely on accurate spelling - use autocorrect or provide suggestions</td> </tr> <tr> <td>let users change the contrast between background and text</td> <td>put too much information in one place</td> </tr> </tbody> </table>	Do...	Don't...	use images and diagrams to support text	use large blocks of heavy text	align text to the left and keep a consistent layout	underline words, use italics or write in capitals	consider producing materials in other formats (for example, audio or video)	force users to remember things from previous pages - give reminders and prompts	keep content short, clear and simple	rely on accurate spelling - use autocorrect or provide suggestions	let users change the contrast between background and text	put too much information in one place
Do...	Don't...																																					
make large clickable actions	demand precision																																					
give form fields space	bunch interactions together																																					
design for keyboard or speech only use	make dynamic content that requires a lot of mouse movement																																					
design with mobile and touchscreen in mind	have short time out windows																																					
provide shortcuts	ire users with lots of typing and scrolling																																					
Do...	Don't...																																					
write in plain English	use complicated words or figures of speech																																					
use subtitles or provide transcripts for videos	put content in audio or video only																																					
use a linear, logical layout	make complex layouts and menus																																					
break up content with sub-headings, images and videos	make users read long blocks of content																																					
let users request an interpreter for appointments	don't make telephone the only means of contact with users																																					
Do...	Don't...																																					
use images and diagrams to support text	use large blocks of heavy text																																					
align text to the left and keep a consistent layout	underline words, use italics or write in capitals																																					
consider producing materials in other formats (for example, audio or video)	force users to remember things from previous pages - give reminders and prompts																																					
keep content short, clear and simple	rely on accurate spelling - use autocorrect or provide suggestions																																					
let users change the contrast between background and text	put too much information in one place																																					

Links

- What is Accessibility. https://developer.mozilla.org/en-US/docs/Learn/Accessibility/What_is_accessibility
- Web Content Accessibility Guidelines (WCAG) Overview. <https://www.w3.org/WAI/WCAG21/quickref/>
- A11Y Checklist. <https://www.a11yproject.com/checklist/>
- Bootstrap Accessibility. <https://getbootstrap.com/docs/5.0/getting-started/accessibility/>
- Making Sense Of WAI-ARIA: A Comprehensive Guide. <https://www.smashingmagazine.com/2022/09/wai-aria-guide/>
- Accessibility Review Checklist. <https://sarahabd.com/blog/my-accessibility-review-checklist>

Links

- The Aesthetic-Accessibility Paradox. <https://uxmovement.com/thinking/the-aesthetic-accessibility-paradox/>
- Accessibility Is So Much More Than Contrast. <https://www.netguru.com/blog/accessibility-more-than-contrast>
- Näide riigiasutuste juurdepääsetavusest. <https://harno.ee/juurdepaasetavus>
- Embracing Accessibility in UX Design: Beyond Compliance to Inclusion. <https://medium.com/design-bootcamp/embracing-accessibility-in-ux-design-beyond-compliance-to-inclusion-d60a5bb2b091>

Thank you!