

Tidepool Resources on Accessibility

Common Accessibility Controls

[Much more information about accessibility features for your iPhone is available here.](#)

Vision

- **Speak screen:** Read a screen's text content to you with a single swipe
 - Set it up (one time only; can return to make adjustments later):
 - In iOS Settings, tap *Accessibility*, then *Speech*
 - Enable "Speak Screen"
 - Make any changes you wish to the voice and speed
 - Use it where you'd find it helpful:
 - Swipe from the top of the screen downward with two fingers to have the phone read the screen's text to you
 - [Read more from Apple on how to use Speak Screen](#)
- **Zoom:** Magnify a certain portion of the screen to be able to see it better
 - Set it up (one time only; can return to make adjustments later):
 - In iOS Settings, tap *Accessibility*, then *Zoom*
 - Enable "Zoom"
 - Make any changes you wish to how Zoom works; the control for magnification size is at the bottom
 - Use it where you'd find it helpful:
 - Double-tap three fingers to activate Zoom
 - Drag three fingers to move around the screen
 - Double-tap three fingers and drag to change the magnification level
 - [Read more from Apple on how to use Zoom](#)
- **Increase text size while using an app:**
 - Set it up (one time only):
 - In iOS Settings, tap *Control Center*
 - Scroll down to the "More Controls" section
 - Tap the green + next to *Text Size*
 - Use it where you'd find it helpful:
 - Swipe down from the upper right corner with one finger to open the Control Center
 - Tap on the AA icon and select the text size increase for the app (you can also set the default size for all apps if you prefer)

- [Read more from Apple on making text easier to read](#)
- **Make text bigger:**
 - Set it up (one time only; can return to make adjustments later):
 - In iOS Settings, tap *Accessibility*, then *Display & Text Size*
 - Tap *Larger Text*
 - Move the slider to change the default text size
 - [Read more from Apple on making text easier to read](#)

Hearing

- **Flash the phone light for notifications:**
 - Set it up (one time only; can return to make adjustments later):
 - In iOS Settings, tap *Accessibility*, then *Audio & Visual*
 - Scroll down and tap *LED Flash for Alerts*
 - Enable *LED Flash for Alerts*
 - You can also select whether you wish Flash to occur when the phone is unlocked and/or in Silent Mode
 - [Read more from Apple on LED Flash](#)

Dexterity

- **Decrease touch sensitivity to prevent accidental taps:**
 - Set it up (one time only; can return to make adjustments later):
 - In iOS Settings, tap *Accessibility*, then *Touch*, then *Touch Accommodations*
 - Enable *Hold Duration*
 - Increase or decrease the hold duration; this is the amount of time your finger must connect with the touch target to activate it
 - You can also select whether you wish this to apply to Swipe Gestures, such as swiping right on an app
 - [Read more from Apple on adjusting Touch Accommodations](#)

Writing great image descriptions

It's been said that image description (otherwise known as alt text) is more art than science. It's also the most frequently found barrier in web accessibility.

Who uses image description?

The most obvious answer to this is people with vision impairments, but it actually goes further than that. Anyone who uses a screenreader benefits from image description. Some use cases:

- Those who use screenreaders for preference, to better focus on the material.
- People with learning disabilities that interfere with reading and/or visual processing.
- People unfamiliar with the mainstream culture in which the images are situated; words can help clarify or explain the visual shortcuts and assumptions that are made in image interpretation.

What are the best practices or conventions around creating image descriptions?

- First, the *context and purpose of the image are everything*. An example: the same image of a painting of George Washington will have different descriptions depending on how it's used. A museum with an exhibit in which the painting is featured will have a significantly different description than a site debunking the myth about his wooden teeth.
- Not everything needs description. If it's got a purpose that adds to or provides further detail to the text, it needs description. If it's decorative, it doesn't.
- What's decorative? Think about the image's purpose! Examples of decorative images include:
 - Lines, boxes, or other frames that draw a reader's attention or help them visually navigate
 - Images designed to break up a wall of text or to keep the reader engaged
- What's not decorative?
 - Brand logos
 - Symbols that identify warnings or other critical information (if it's defined and described in the Instructions for Use, it darn well better have an image description)
 - Images of text; these are not otherwise reliably readable by a screenreader
 - Anything that conveys content to your sighted user
- Don't bury the lede! When describing a complex image, start with a summary of the most important information and then drill down into details. If the user wants that detail, they can continue listening or skip out of it once they have the gist.

- Skip the words “image description”, “alt text”, “picture of”, “image of”, “diagram of”, and the like. The screenreader has already notified the user that it’s an image.
- If, however, the type of image is important to the content in its context, include that information. Line graph of increasing glucose values over six hours is different than chart with glucose values; you need to determine why you are using the image in the first place to decide which is appropriate.
- Be concise! Don’t burden your user with unnecessary length.
 - Google uses the first 70 characters for SEO
 - Tooltips and other hover actions generally will pull the description; these are often limited to 125 characters
 - SwiftUI and ReactNative do not have limits on descriptions
- Don't substitute intended emotional impacts for specific descriptions (it's *purple cow*, not *happy cow*), but don't shy away from using emotive descriptors as well when they are appropriate to the context (*purple, smiling and winking cow*)

Let’s talk about data viz

- SVG is the preferred format
 - Scales cleanly for zooming (different topic but also helpful to many of the same populations)
 - Allows individual data points to be identified and read
- Think about how to summarize the data, then how to allow your user to drill down. Consider adding a single summary sentence to a caption. This can also be done with dynamic data viz, with a little forethought. For example: Your glucose has been trending x over the last y hours/minutes and is predicted to z over the next hour to #.
- Where mouse/keyboard/touch actions to access the data are not immediately clear, provide instructions or hints that can be accessed at any time (not just in the initial onboarding). For the ultimate user control, allow the user to toggle whether or not to see hints.
- If it’s truly not possible to make the same level of detail available to screenreader users that is available to nonscreenreader users, add a table of the data that the user can access.

Tools

Check out the Accessibility Tools below for some nifty things to help you remember to include image descriptions!

Further reading

- [ACES Society for Editing guide to great alt text](#)
- Sarah Fossheim's [Writing alt text for data visualization](#)
- [SiteImprove's guide to alt text](#)
- [WebAIM Guide to writing alternative text](#)
- [W3C's guide for designers on how to handle non-textual content](#)

Accessibility Tools

A list of tools that can be used at various places in the software development life cycle to improve disability. The goal is to shift left; move identifying and resolving accessibility challenges to earlier and earlier points in the development cycle, where they are easier (and less expensive!) to handle.

Screenreaders and how to use them

[Voiceover \(MacOS and iOS screenreader\)](#)

- [How to use VoiceOver to evaluate web accessibility](#)
- [Using VoiceOver to evaluate iOS app accessibility](#)

[TalkBack \(Android screenreader\)](#)

- [How to use TalkBack to evaluate accessibility](#)

[NVDA \(Windows screenreader\)](#)

- [How to use NVDA to evaluate web accessibility](#)

[JAWS \(Windows screenreader\)](#)

- [How to use JAWS to evaluate web accessibility](#)

Design tools

Alt-text checkers

[Adee Alternative Text Generator](#) (Figma plugin)

(the manual and mixed development tools will also include checking for image descriptions)

Color contrast checkers

- [Accessible Color Generator](#) (Standalone)
- [Color Contrast Grid](#) (Figma plugin)
- [Contrast Checker](#) (WebAIM standalone)
- [Colour Contrast Checker](#) (Chrome extension)
- TGPi Color Contrast Checker (Standalone; eyedropper tool to pick colors from anywhere on your screen)

Colorblind-friendly checkers

- [Able](#) (Figma plugin)
- [Sim Daltonism](#) (Mac/iOS standalone)

ARIA tools

- [React Aria](#) (library of React Hooks that provides accessible UI primitives)

Development tools

Linters

- Axe-linter (coming soon)
- [ESLint Plugin for React](#)
- [ESLint plugin for React Native](#)
- [Webhint](#) (vscode extension)

Android

- [Accessibility Insights for Android](#)

Continuous integration

- [pa11y-ci](#) (command line tools for pa11y)
- [lighthouse-ci](#) (command line tools for Google's Lighthouse)
- [A11yUITests library](#) for xcuitest

Unit Testing

- [axe-core](#)
- [axe-jest](#) (aXe-core integration for Jest)

Manual testing – tools

[Web Accessibility Evaluation Tool \(WAVE\)](#)

- [WAVE Firefox extension](#)
- [WAVE Chrome extension](#)

Automated testing

- [pa11y](#)

- [axe-core webdriverio](#) (axe-core integration)
- [axe-core nightwatch](#) (An aXe-core integration for Nightwatch)

Mixed automated and manual

- [Accessibility Insights for Web](#)