Accessibility

:focus Styles

The goal of :focus is to give the user guidance on where exactly they are in the document and help them navigate through it. To achieve that, we need to avoid a focus that's too subtle or not visible at all. In fact, removing outline is a bad idea as it removes any visible indication of focus for keyboard users. The more obvious the focus is, the better.

Autocomplete

Every time you have to deal with a larger data set, be it a map, a data visualization, or just a country selector in checkout, autocomplete can boost customer's input massively. But just as it helps with the input, it needs to help with announcing the options and the selection to the screen reader users as well.

Buttons and Icons Links

It's not uncommon to have a link or button that visually has no text but consists only of an icon — a compact navbar, for example, or social icons.

Disabled Buttons

It has become quite common for lengthy web forms to keep the "Continue" button disabled until the customer has provided all data correctly. This behavior acts as an indicator that something is wrong with the form, and it can't be completed without reviewing the input. This works if the inline validation for every input field is working well, and it doesn't work at all when it's glitchy or buggy.

Accessible Cards

Cards offer a lot of advantages. They work well on mobile, provide large click areas, and the fact that they can be stacked both horizontally and vertically makes a lot of layout decisions easier. However, there’s no accessibility standard to follow, no <card> element or an ARIA design pattern. Instead, the potential accessibility barriers you might encounter depend on the card’s purpose and content.

Carousels and Content Sliders

An accessible carousel sounds a bit like oxymoron — while there are plenty of scripts that provide the functionality, only few of them are accessible. Now there are, of course, accessible range sliders, but carousels are a slightly different component.

Close Buttons

“Close” buttons are everywhere — in modals, ads, confirmation messages, cookie prompts and any overlays that will appear in your interface. Unfortunately, the functionality is often limited to mouse users, leaving screen reader users and keyboard-users out.

Checkboxes And Radio Buttons

The good ol’ issue: how do we style checkboxes and radio-buttons to ensure that they look, well, at least similar, in most browsers. When hiding an interactive element, we need to make sure we choose a hiding technique that keeps it screen reader-accessible, position it on top of whatever is visually replacing it, so that a user navigating by touch can find it where they expect to, and then make it transparent.

Color Systems

Getting color contrast right is an essential part of making sure that not only people with visual impairments can easily use your product but also everyone else when they are in low-light environments or using older screens. However, if you’ve ever tried to create an accessible color system yourself, you probably know that this can be quite a challenge.

Color Palettes

Finding the perfect tint or shade of a color is not only a matter of taste but also accessibility. After all, if color contrast is lacking, a product could, in the worst case, even become unusable for people with vision impairments. WCAG 2.0 level AA requires a contrast ratio of at least 4.5:1 for normal text.) and 3:1 for large text, and WCAG 2.1 requires a contrast ratio of at least 3:1 for graphics and UI components (such as form input borders). AAA requires a contrast ratio of at least 7:1 for normal text and 4.5:1 for large text.

Automating Accessibility Testing

Maybe it's a missing alt attribute or a heading structure that isn't semantic, often it's little accessibility issues like these that slip our attention and make it into production. The GitHub app AccessLint is here to prevent this from happening by bringing automated web accessibility testing into your development workflow: When you open a pull request or make edits to an existing one, AccessLint is already there, automatically reviewing the changes and commenting with any new accessibility issue before the code goes live.

Understanding Visual Impairments

You've probably heard of protanopia, deuteranopia, or glaucoma before. But how do people with visual impairments like these actually see your color combinations? Corey Ginnivan's tool Who Can Use simulates it for you.

Component Libraries

While many of the component libraries we create are trying to cover all the usual suspects (the accordions, the tables, the carousels, the drop-downs, along with typography, colors and box shadows), No Style Design System by Adam Silver is focused primarily around accessibility and web forms.

Cookie Consent Prompts

Overlays and pop-ups are always problematic. But especially for screen reader users, sometimes those prompts are incredibly difficult to deal with to set any settings or even confirm the usage of cookies on the site. In some cases, users glide past consent prompts without being aware of them, in the others, the prompt are impossible to accept, resulting in an inability to use the site at all.

Current Page Navigation States

Color is an effective way to convey meaning, but it's always a good idea to have a second visual indicator for people with low vision or color vision deficiencies, too. An icon, for example.

A Complete Guide To Dark Mode On The Web

Dark mode is quickly becoming a user preference with Apple, Windows, and Google having it implemented into their operating systems. But what about dark mode on the web? Adhuham wrote a comprehensive guide to dark mode that delves into different options and approaches to implementing a dark mode design on the web.

Data Charts

Data visualizations are a great way to make information stand out. However, they also come with their own accessibility challenges.

Data Visualizations

Data visualizations often contain important information that users have to act upon. While sometimes we can use large numbers with short sentences instead, visualizations can help understand developments and large amounts of information faster. But that means that the information has to be easy to understand, and that refers especially to the selection of colors, the way information is presented, labels, legends as well as patterns and shapes.

A Flexible Data Visualization Library

When Torstein Hønsi was looking for a simple charting tool for updating his homepage with snow depth measurements from the local mountain where his family keeps a cabin, he was frustrated with what he found. And, well, that's when he decided to build his own solution and share it with the world. The result is Highcharts, a flexible charting library that comes with all the tools you need to create reliable and secure data visualizations.

Date Pickers

There are dozens of date picker libraries out there, but it's always great to have reliable workhorses that just work across browsers, don't have heavy dependencies, are written reasonably well, and meet all major accessibility requirements.

Styling Horizontal Dividers

<hr> elements are usually quite boring. Plain, horizontal lines that provide a visual break and divide content. But did you know that they can be styled using CSS and SVG to give your content and designs a nice personal touch?

Cross-Browser Form Styles

Have you ever struggled with hiding and styling custom checkboxes and radio buttons? What about custom select-styles? Or perhaps an accessible dropdown-navigation menu? We tend to build and rebuild the same components all the time, so let's get them right once and for all.

Footnotes And Sidenotes

In their essence, footnotes aren't much more than jump-links — links to the description of a source, either placed at the bottom of the document, or in the sidebar, or appearing inline, a little accordion. However, as footnotes are jump-links, we need to ensure that screen reader users understand when links are references to footnotes — and we can do it with the aria-describedby attribute. The counter for every link would be implemented via a CSS counter. With :target, we then highlight the row which the reader has jumped to, and we provide a back-link back to the actual footnote placed in the content.

Inputs

In 2019, WebAIM analyzed the accessibility of the top one million websites, with a shocking conclusion: the percentage of error-free pages was estimated to be under one percent. To make our sites inclusive and usable for people who rely on assistive technology, we need to get the basics of semantic HTML right.

The Perfect Link

A link is a link is a link, right? Well, there's more to a link than just a clickable word or image. With her article "The perfect link", Rian Rietveld examines how to write, design, and code a link that works for everyone on every device. Rian covers the question if a link should open in a new window or a new tab, how to make link texts understandable, how to handle links to an email address, telephone number or file, what you need to consider when embedding an image in a link, when to underline it and how to deal with hover and focus styles, as well as semantic matters and internal links. A 360-degree look at the topic.

App-Wide Keyboard Navigation

A well-thought-out concept for keyboard navigation benefits everyone: It enables people who can't comfortably use a mouse, assists screen reader users in interacting with an application, and it provides power users with more shortcuts to work as efficiently as possible. Usually, keyboard support is limited to specific shortcuts, but the team at Discord decided to go a step further with their application and expand keyboard support to, well, everything. The case study “How Discord Implemented App-Wide Keyboard Navigation” shares valuable insights into how they tackled the task — and the challenges they faced along the way, of course. One turned out to be particularly difficult: How to consistently indicate where focus is on the page? As existing solutions for Focus Rings didn’t work out, the team had to build their own solution from scratch and made the code open source. If you’re facing a similar challenge, this one’s for you.

Tap/Click Menu

Is it still a good idea to design mega-drop-downs opening on hover? Probably not. Hover menus have plenty of usability and accessibility issues, as they are inconsistent, confusing and of course need an alternative solution for mobile devices.

Accessible

You might have a simple modal or overlay on the page, perhaps to confirm customer’s input, or to show a couple of photos in a gallery, or just to confirm user’s preferences. In all these cases, building an accessible modal will turn out to become quite an adventure, also know as a focus trap.

Password Fields

“Show password” and password hints make form fields more usable. They help users figure out if they mistyped their password as well as what pattern is acceptable when they create a new one. However, as it turns out, accessibility is often lacking when it comes to these things.

Support User Preferences With prefers-reduced-*

Not every user is the same, and while some users love animations, others may have medical issues concerning motion. The prefers-reduced-motion media query lets you toggle animations on and off, but there are even more solutions to manage animations depending on a user’s preference.

“Skip” Links

Especially on pages with a large amount of navigation, moving between sections or around the page can be frustrating and annoying. That’s where “Skip” links can be very helpful. Unfortunately, it’s not uncommon to see “Skip” links being implemented but hidden away with display: none, and as such, unavailable to anybody (including screen reader users and keyboard users).

SVGs

Talking about SVGs: what we can do with SVGs today goes way beyond the basic shapes of yesteryear. Not only can we describe SVG icons, but also style and animate them. If true inclusiveness lies beyond patterns — what other factors should we consider when designing and developing accessible SVGs? That’s exactly the question that Carie Fisher is answering in her piece on Accessible SVGs: Inclusiveness Beyond Patterns. In the article, Carie takes a closer look at SVG color and contrast, light and dark modes, SVG animation, reduced motion and plenty of tools focused all around accessibility. You’ll also find demos and code examples in the articles, along with detailed explanations and pointers for further reading.

Tabs

Your interface might be using tab panels, but to keep the content of these tabs accessible to keyboard-users and screen reader users, we need a very careful and deliberate exposition of visual design and ARIA semantics.

Tables

There are plenty of accessibility issues related to tables, but the biggest challenges is to turn a visual representation into a linear series that will be read aloud meaningfully by a screen reader, without omitting any important information.

Toggle Switches

Whenever our forms provide a binary selection to our customers — on/off, dark/light mode etc. — we could use a toggle switch. The switch needs to serve a couple of purposes: it needs to clearly explain the current selection (and that's clear not that often at all!), it needs to explain that there are two options, and it needs to be obvious enough for customers to understand how to switch between them.

Tooltips And Toggletips

A component that's closely related to icon buttons is a tooltip. Literally "tips for tools", they are little pieces of information that explain the purpose of a control, or a visual, that otherwise could be misunderstood. Every time we want to explain why we need a particular piece of personal information in a checkout, that's where we'll probably be using a good old tooltip.

Video/Audio Players

It's not uncommon to see viewers frequently using captions when watching a short clip or a lengthy movie these days. We might be consuming the video in a noisy environment, or perhaps we can better understand written language, or perhaps we are currently busy with something else and need to look up something quickly without having to resort to headphones. Beyond that, how often do we use keyboard's <space> to prompt a pause, or key arrows to move back and forward? Still, many video players and custom solutions don't provide this functionality out of the box.

Website Features That Annoy Screen Reader Users

A missing alt caption, an auto-playing video, unlabelled buttons, poor use of headings, inaccessible web forms — what might seem like a small issue for sighted users can make the difference between being able to use a website independently or not for blind and visually impaired people.

References

A Complete Guide To Accessible Front-End Components — Smashing Magazine