

Accessibility Quick Wins

You can implement today

Daniel Yuschick

Senior UI Developer

@DanielYuschick

Table of Contents

1. Relating to Accessibility
2. Accessibility Quick Wins
3. Takeaways
4. Resources



Relating To Accessibility



Web accessibility is exactly like **responsive design**
but for additional devices and people.



Responsive Design

- Requires cross-team collaboration
- Benefits an ever-expanding audience
- Required for a positive user experience
- Directly impacts conversions
- An expectation from the user

A product requirement

Web Accessibility

- Requires cross-team collaboration
- Benefits an ever-expanding audience
- Required for a positive user experience
- Directly impacts conversions
- An expectation from the user

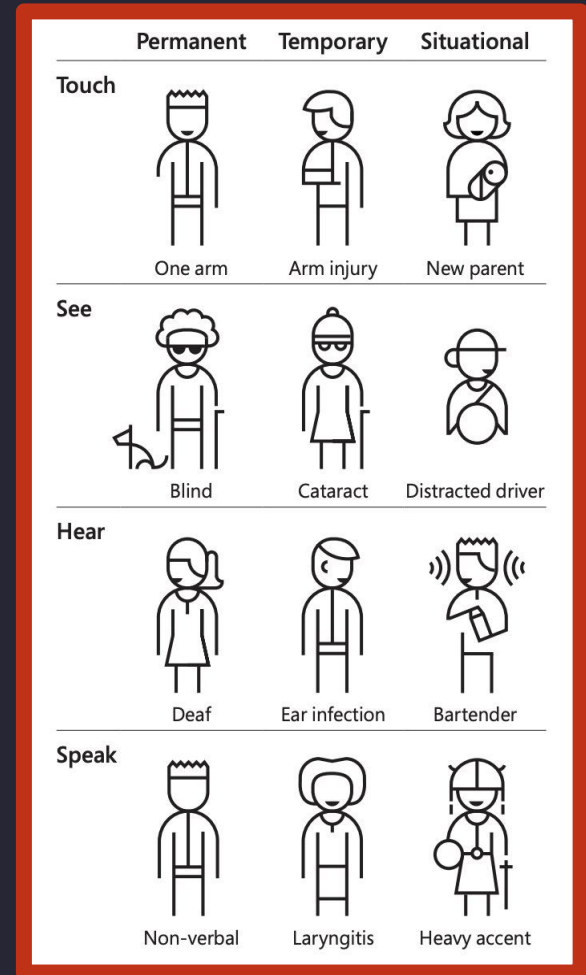
Not a product requirement 🤔





TELL ME WHY

Web accessibility is fundamentally
misunderstood.



An **accessible** product is one with **markup** that communicates as clearly as its **design**.



Accessibility Quick Wins



Indicate The Current Page

Why is this important?

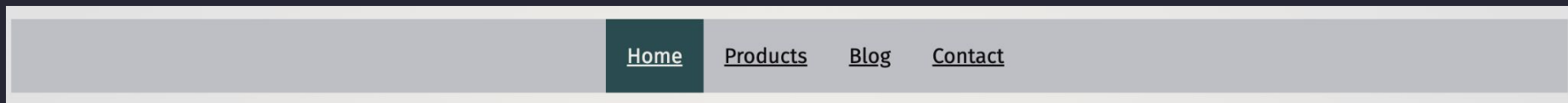
People with low vision may not see an underlined navigation item.

Or people using screen magnifiers may be too zoomed in to see a changed background color or other active navigation item visual cues.



Indicate The Current Page

 [View the demo](#)



A visual indicator of the current page.



Indicate The Current Page

 [View the demo](#)

```
<nav>
  <ul>
    <li><a href="/" class="active">Home</a></li>
    <li><a href="/products">Products</a></li>
    <li><a href="/blog">Blog</a></li>
    <li><a href="/contact">Contact</a></li>
  </ul>
</nav>
```

✗ This markup does two things:

1. It requires creating an `.active` class in CSS
2. It fails to communicate to assistive technologies the same information it does visually

```
<nav>
  <ul>
    <li><a href="/" aria-current="page">Home</a></li>
    <li><a href="/products">Products</a></li>
    <li><a href="/blog">Blog</a></li>
    <li><a href="/contact">Contact</a></li>
  </ul>
</nav>
```

✓ This markup does two things:

1. It prevents the need for creating a new CSS class
2. It uses `aria-current="page"` to communicate with assistive technologies the same information it does visually



Prefers Reduced Motion

Why is this important?

Many people find motion, flashing colors, and abrupt spacial changes disorienting, particularly on screens. Windows and MacOS offer settings to reduce the motion of their systems to improve its experience and safety, and the web must respect that setting as well.



Prefers Reduced Motion

 [View the demo](#)

```
@media (prefers-reduced-motion: reduce) {  
  * {  
    animation-duration: 0.01ms !important;  
    animation-iteration-count: 1 !important;  
    transition-duration: 0.01ms !important;  
    scroll-behavior: auto !important;  
  }  
}
```

- ✓ This markup does two things:
1. The `@media` query respects a person's system settings for reduced motion
 2. It reduces all animation and transition durations to a near-zero value.
 - a. By setting the durations to 0.01ms, any Javascript listeners will still trigger. If no Javascript listeners are present, the durations can be 0.



Prefers Reduced Motion

[View the demo](#)



"Preventing all animations and transitions is a bit heavy handed; don't ya think?"

For more nuance, reduce only motion-related properties, like:

- Translate
- Scale
- Height/Width
- Top/Right/Bottom/Right

Allows:

- Link and button `:hover` color transitions
- Opacity transitions



Skip To Main Content Link

Why is this important?

People using a keyboard to navigate the web will need to tab through every navigation item of each page before reaching the content.

Providing a 'Skip to main content' link allows that repeated section to be skipped.



Skip To Main Content Link

 [View the demo](#)

1.

```
<a  
  class="visually-hidden"  
  href="#main-content"  
>  
  Skip to main content  
</a>  
<header> ... </header>
```

2.

```
<a  
  href="#main-content"  
  class="visually-hidden"  
  id="main-content"  
>  
  Start of main content.  
</a>  
<main> ... </main>
```

1. Skip to content trigger: create an anchor tag at the start of the document (or section to skip).
2. Skip to content target: create an anchor at the start of the main content section.



Skip To Main Content Link

 [View the demo](#)

```
.visually-hidden {  
  -webkit-clip-path: polygon(0 0, 0 0, 0 0, 0 0);  
  clip-path: polygon(0 0, 0 0, 0 0, 0 0);  
  box-sizing: border-box;  
  position: absolute;  
  margin: 0;  
  padding: 0;  
}
```

Create a `.visually-hidden` class in CSS to hide content from the UI while remaining accessible to assistive technologies.



Accessibility Takeaways



Accessibility Takeaways

- Accessibility is a standard, *not* a feature.
- An accessible product has markup that communicates as clearly as its design.
- All teams must collaborate toward accessibility.
- It's easier to re-design a feature than to hack away in development.
- Accessibility isn't "all or nothing."
- 💡 Accessibility is a standard, *not* a feature.



Web Accessibility Resources



Web Accessibility Resources

- [Axe Dev Tools](#) - Deque
- [Practical Accessibility Course](#) - Sara Soueidan
- [Build & Test Accessible Web Apps Course](#) - Marcy Sutton
- [A11yphant](#) teaches web accessibility, one step at a time.
- [Layman's Accessibility Checklist](#)
- [A Deep Dive Into Skipping to Content](#) - Paul Ratcliffe
- [Demos Collection](#) - Codepen
- [5 Accessibility Quick Wins You Can Implement Today](#) - Daniel Yuschick



Thank You

Twitter: @DanielYuschick

