

Accessibility Best Practices

Context

Accessibility is often overlooked when designing applications. It is an extremely important part of inclusion. A developer must think about accessibility from inception.

In order for us to really understand how to build a rich web experience, we need to know how the web looks like and feels like for those with accessibility needs.

Structure

Time	Topic
5	Warmup
15	Overview
5	Pomodoro
45	Experiment
5	Pomodoro
15	Review

Warmup

With a neighbor, discuss the following:

- What do you know already about how web applications relate to accessibility?
- What have you done already in your projects to make them more accessible?
- What are you most curious about regarding accessibility?

Goals

- Build empathy for people with accessibility needs
- Identify the 4 main accessibility issues
- Build familiarity with accessibility tools

In the news...



3 billion

Internet users

57 million

Americans with a disability (2012)

**Consider the following statistics
from the Census Bureau's
survey taken on July 25, 2012.**

19.9 Million (8.2%) have difficulty lifting or grasping. This could, for example impact their use of a mouse or keyboard.

**15.2 Million (6.3%) have a
cognitive, mental, or emotional
impairment.**

8.1 Million (3.3%) have a vision impairment. These people might rely on a screen magnifier or a screen reader, or might have a form of color blindness.

7.6 Million (3.1%) have a hearing impairment. They might rely on transcripts and / or captions for audio and video media.

According to the Pew Internet Project Survey
conducted by Princeton Survey Research Associates
International,

54% of adults living with a disability go online.

**Web applications that
accommodate all people.**

Internet == for_the_people
skin_color != matter
SES_background != matter
physical_ability != matter

With great power...

These are *NOT* edge cases.

Advocate for all users.

If the internet was meant for everyone then we should make it for everyone.

...this is for everyone...

— Tim Berners-Lee, re: HTTP/HTML

The computer will always do exactly what you tell it to. Maybe its time we start telling them to do the right thing.

Here are the top 4 accessibility things to look out for on the web:

- Visual
- Mobility
- Cognition
- Hearing

Making sure we help accommodate these users allows us to really gear and tune our UI for everyone.

Today, we will talk about the first three on the above list.

Visual

Tools Needed

- chrome_vox, a screen reader
- I want to see like the colour blind, a chrome extension that emulates color blindness

Experiment!

Discussion

1. What was the biggest difference between both experiences?
2. What was the most frustrating thing for you?
3. Could you get the screen reader to read the whole article for you?

Code tip

Radio buttons === hard to click.

Event delegation to the rescue!

Often times it's just a matter of wrapping your radio button in a label tag.

```
<label>  
  <input type="radio" name="rgroup" />  
  Carrots  
</label>
```

Mobility

Mobility is an issue where the user typically has difficulty using a mouse or a trackpad. The first and easiest thing we can do as developers is making sure we are making our application completely usable with just a keyboard.

Video + Experiment

see markdown

Discussion

1. Do you see a benefit to using tabindex?
2. What did you notice about the Auto Trader website while only using tab interaction?

Code Tip

```
<p tabindex="0">Hello there</p>
```

Cognition

read webaim description in markdown

Read + Experiment

Discussion

1. What steps can we take to consider people with cognitive issues?
2. Can you think of any website you know that would be difficult to use for someone with cognitive issues?

Code Tip

Organizing your data in a way that is decreeable and engaging for the user will help combat this. Some things we can do to help draw attention to our document:

— Using proper headings, and lists

- Using proper headings, and lists
- Using more white space in your design

- Using proper headings, and lists
- Using more white space in your design
- “Chunking” content into more manageable pieces

- Using proper headings, and lists
- Using more white space in your design
- “Chunking” content into more manageable pieces
- Making forms manageable by breaking them into multiple, sequential steps

- Using proper headings, and lists
- Using more white space in your design
- “Chunking” content into more manageable pieces
- Making forms manageable by breaking them into multiple, sequential steps
- Providing a logical reading order

- Using proper headings, and lists
- Using more white space in your design
- “Chunking” content into more manageable pieces
- Making forms manageable by breaking them into multiple, sequential steps
- Providing a logical reading order
- Being consistent with fonts, colors, and locations of page elements

- Using proper headings, and lists
- Using more white space in your design
- “Chunking” content into more manageable pieces
- Making forms manageable by breaking them into multiple, sequential steps
- Providing a logical reading order
- Being consistent with fonts, colors, and locations of page elements
- Offering keyboard access

- Using proper headings, and lists
- Using more white space in your design
- “Chunking” content into more manageable pieces
- Making forms manageable by breaking them into multiple, sequential steps
- Providing a logical reading order
- Being consistent with fonts, colors, and locations of page elements
- Offering keyboard access
- Consider offering content in multiple formats

Finding Violations

- Add the axe extension.
- In the developer tools of any website, click on the "aXe" tab and "analyze violations" which will display any violations for a given site.

Wrap up

Now that we are more aware of what is going on in the world of accessibility, what are we going to do to make sure we are committing ourselves to making the web a better place for everyone?

Write down 3 actionable commitments you can do in your next coming projects to accommodate for accessibility issues and submit them [here](#).