Basics of accessibility

Also known as a11y

@samikeijonen

Why a11y matters

In short, it leads to a better web

Where to start

Change of mindset is a good start

Accessibility is a mindset where you build the basics = HTML first.

Developer point of view

Then try not to destroy the basics with CSS and JS.

Front-end dev point of view

Would you build a bridge that only a red Ferrari can use?

- Andrea Fercia

Philosophical point of view

Accessibility is the practice of making web usable by as many people as possible

General inclusive point of view

Accessibility, usability, and inclusive design are closely related

UX and design point of view

Okay, but why accessibility matters

See how “accessibility matters” is in it’s own line

It’s about us

Not just about blind people

Diversity of people

Different situations

- We get old
- Low vision / hearing
- Reading difficulties like dyslexia
- Color blind
- Hand broken, new mum / dad
- Sun shine
- Usher syndrome
- Hotel wi-fi
- etc.

Accessibility issues are: permanent, **temporary**, situational

Good for business

$$$

- Many (20%?) rely on web being accessible
- Everyone benefits web being usable

Good for usability

Users like your site or app!

- Intuitive design
- Ease of learning
- Efficiency of use
- Memorability
- Error frequency and severity
- Subjective satisfaction: users like your website!

Good for SEO

Search engines are blind.

- Semantic HTML
- Headings and page structure
- Image alt text
- Link anchor text
- Page titles
- Video and audio transcripts
- Writing accessible content

Required by law

"Breaking the law, breaking the law"

- US section 508
- EU directive
- Rules are inline with Web Content Accessibility Guidelines (WCAG).

1. Perceivable (havaittava):

Information and user interface components must be presentable to users in ways they can perceive.

2. Operable (hallittava):

User interface components and navigation must be operable.

3. Understandable (ymmärrettävä):

Information and the operation of user interface must be understandable.

4. Robust (toimintavarma):

Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

Where to start?

Change of mindset, empathy

Resources

- WP a11y handbook
  - Markup
  - Design
  - Content
  - Testing
- Inclusive components
- WCAG guidelines
- Web Accessibility Initiative (WAI)
- ARIA practises
- Accessibility guidelines in Finnish

Quick tips

Use semantic HTML elements

HTML for the win.

Use semantic HTML elements

HTML for the win.

Use semantic HTML elements

HTML for the win.

Use semantic HTML elements

HTML for the win.

Use semantic HTML elements

HTML for the win.

Use semantic HTML elements

HTML for the win.

Use semantic HTML elements

HTML for the win.

Use semantic HTML elements

HTML for the win.

Use semantic HTML elements

HTML for the win.

Use semantic HTML elements

HTML for the win.

HTML5 sectioning elements

Gives structure of a web page

Defines ARIA landmarks by default

HTML5 sectioning elements

Provides a method for assistive technology users users to navigate these section.

Where is navigation, where is main content?

Most common landmarks

<header>
<nav>
<main>
<aside>
<footer>

```
<html>
 <head>
   <title>Page title</title>
 </head>
 <body>
  <header>
      <nav>Nav list</nav>
   </header>
   <main>Main content</main>
   <aside>Other content</aside>
   <footer>Footer content</footer>
 </body>
</html>
```

Semantic markup

 and doesn't expose any meaning to anything.
Use them only for layout and styling purposes

Use meaningful markup:

<a>
<button>
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>
<p>
<img>

<blockquote>
<figure>
<time>
<article>
<address>
<form>
<fieldset>
...

Unfortunately you can’t blindly trust HTML accessibility

HTML: The Inaccessible Parts
HTML5 Accessibility

Test everything with keyboard only

When interface works with keyboard, most of the work is done.

Most common keyboard controls for navigating are Tab, Shift + Tab, Enter, Space, and arrow keys.

Do not remove focus styles

:focus { outline: none; } ← never do this

Keyboards users needs to see where they are navigating.
That’s why interactive elements like links and form fields needs focus styles.

Focus (state) styles

Designing focus, hover, error etc. styles

Keep it simple

Modals, infinite scroll, sticky elements, complex forms, main point is buried etc.

The WebAIM Million

An annual accessibility analysis of the top 1,000,000 home pages.

Remember color contrast

Between foreground and background colors.
For example between button text and background color.

And use of color

Color should not be the only indicator for information or function.

For example red border alone for error in text field is not enough. Inline error message for the rescue.

Images and Alternative Text

Provide alt text even if it’s empty:
<img src=”decorative.jpg” alt=””>
<img src=”laptop.jpg” alt=”Laptop with basics of accessibility presentation open.”>

An alt Decision Tree

Form labels

```
<label for="name">Name:</label>
<input id="name" type="text">
```

Matching for and id values associate the label with its form control.
User can also click on the label to set focus to the form control.

Headings

Quickly scan and understand page content.

At the same time headings give AT users a way to navigate the content and define page structure.

Headings

One <h1> on a page

Don’t skip heading levels

After <h2> should only come <h2> or <h3> etc.

Link text

Avoid repetitive link text like “Read more”, “Click here”.

Link text should be understandable on it’s own, for example “Learn HTML deeply”.

Learn HTML deeply

Skip links

First focusable element to take user to the main content, and skip all header information like main nav.

Skip links

```
<body>
    <a class=”visually-hidden skip-link”
    href="#content">Skip to content</a>
    ...
    <main id=”content”>...
```

Doctype

<!DOCTYPE html>

Ensure browsers are doing their best rendering HTML document.

Add document language

<html lang="en">

Let assistive technology know the language that your text is in.

That’s it, thanks!

Sami Keijonen