Inclusive and Accessible Web Content Guide

Introduction (and Work in Progress)

This guide is intended to give a multi-faceted view on how one may go about designing and making an inclusive website. If you are developing for a platform that uses themes (like Wordpress, or Drupal), or using a front-end framework (like Bootstrap), this guide will also give you insight into what features are desirable (or undesirable) in choosing a solution.This is very much still work in progress. Please feel free to add to this page, or leave comments.

Web inclusion and accessibility is very much an art, and the objective is to provide an excellent experience to everyone, regardless of ability and methods of access.

Table of Contents

Good, clean semantic HTML markup

This will get you 70% closer to an accessible website.

Why use semantic markup? Many assistive technologies, like screen readers, can leverage semantics of a webpage to provide better experiences. For example, instead of putting top level navigation in a , a <nav> element will provide a landmark to a screen reader which will allow users to quickly navigate your site.

Why does HTML need to be clean? Simpler the markup, easier it will be for someone to use your site, and it will load faster too! Some typical scenarios are nested containers which serve no real purpose except to provide a named selector for styling or scripting. If you don't need or use these selectors, remove them from your markup.

Resources

- Quick-and-Dirty Website Accessibility Tests - and Fixes (wiki.fluidproject.org)
- Code Project HTML and CSS basics (includes some semantic markup)
- How to Use Headings on your Site (yoast.com)
- W3School - HTML 5 Semantics (w2school.com)
- Mozilla Developer Network - HTML elements (dev.mozilla.org)

Let users bring their own tools

Todo: write about accessibility tools and the role of the content creator.

Be Cautious with Dynamic Content and Widgets

Many websites use dynamic content (i.e. content that changes or appears after the initial page load). While these effects may be desirable, it poses challenges for many users. Some common examples are image carousels, login "modal" dialogs, or status messages and notifications.

How dynamic content can cause problems:

- Individuals who process information slowly, or have challenges processing information may find it confusing or disorienting if parts of the site changes. Some dynamic content can interrupt user focus.
- Many assistive technologies access web sites via a copy of the HTML created on load time. A possible problem occurs when the original content dynamically changes - somehow those changes need to get communicated to the copy used by the assistive technology. To help communicate changes, ARIA markup is used.

Possible solutions

- Give accessible controls to the user to control movement and presentation of changing content.
- Use alternative methods of presenting content. For example, instead of a content carousel, consider using divided content sections to display the same information.

Resources

- Using ARIA - W3C
- WAI-ARIA Authoring Practices
- Open Ajax Examples - Examples of dynamic web components with ARIA.
- Mozilla Developer Network - ARIA
- "What is WAI-ARIA, what does it do for me, and what not?" - an older resource (published in 2014) dealing with ARIA 1.0. The principles are still applicable.
- W3C WAI-ARIA 1.1

Use Root Font Relative Sizing in CSS

In CSS sizing and positioning content and containers can be done in many ways such as using px, em, or rem. We suggest using rem units for CSS (and your CSS and themes should use it too).

Rem units is defined as: "Relative to font-size of the root element". This way you can specify a root font size of 16px, and by using rem units everywhere else, any changes to the root font size will scale your site accordingly.

Why use rem units?

Using rem units makes it possible for users to easily transformations and customization of your site to suit their personal needs. For example, if someone requires larger text because they want better legibility, all they need to do is change the root font size and everything resizes to fit automatically.

This article "Font sizing with rem" on Snook.ca explains why this is a good idea and how it can be done. There have been some counter-arguments to not use rem units recently (like this article "R.I.P. REM, Viva CSS Reference Pixel!"), but because we believe in user agency and personalization, we recommend using root-relative units.

For what it's worth, major CSS frameworks (i.e. Bootstrap, and Zurb Foundation) are using root relative units as it's proven to be the most robust way of sizing and positioning HTML.

Content Robustness, Multiple Modalities, and Access Personalization

Using root font relative sizing is one step toward the larger goal of more flexible, malleable, and robust web content. With the wide variety of devices, computers, and ways of accessing internet served content (i.e. via a browser, a mobile app, short message feeds etc.), the content you have should withstand different presentation modes and transformations.

Multiple Modalities and Formats

A good strategy for ensuring the content you create today is accessible in the future is to have your content stored and presented in alternative modalities and formats. For example, images or video can be accompanied by a caption or text transcript. This allows users with limited bandwidth or older devices access to your content even without the visuals. The text transcript also allows access for users with low or limited vision, and is more malleable for personalization (such as contrast modes, and different text presentation).

When possible avoid captioning by rendering the captions directly into the video. Instead use the platform's captioning functionality so it can be controlled by the user. You can also use Amara.org to create captions and translations of the video. Be sure to check the translation text if using an automatic captioning system.

All-You-Can-Access Content Buffet

By providing your content in multiple formats and modalities, your users can choose how best to access and digest your content. Often by providing multiple modes of access, your content is more easily understood and retained. Care should be taken not to "overload" information. One strategy could be to provide ways to reveal additional content on-demand (such as a "Show Transcript" feature), this way the primary mode of access is presented initially and the secondary modes are accessed as needed.

Content Lost to Flash

The shift away from Flash as a rich web content format has resulted in valuable legacy content that is no longer accessible. A lesson learned from this scenario is to safe guard content by not relying on a single platform or format.

Resources:

- Multimodal Design Patterns for Inclusion and Accessibility (slide presentation)
- Amara.org - crowd-sourced video captioning and transcript platform
- POET image description tool
- Pressbooks - open textbook publishing

Web Standards are a Baseline

Many jurisdictions that have accessibility regulations for online content require WCAG 2.0 AA compliance (Also visit Government accessibility standards and WCAG-2 on PowerMapper) .

We often get caught up on WCAG 2.0 AA compliance and we lose sight of the bigger picture. A website may satisfy WCAG 2.0 AA criteria, but it does not necessarily mean that the website gives an enjoyable or usable experience, or that the website is inclusive, or takes into consideration conditions outside of the specification.

WCAG is baseline web accessibility, and is not the end of the journey for inclusion. For example WCAG does not cover many learner or cognitive differences.

An Analogy

Traffic laws do not teach a person to be a (good) driver. Similarly, WCAG compliance does not mean a website is inclusive or even usable by all persons.

Resources:

- Quick-and-Dirty Website Accessibility Tests - and Fixes
- W3C Validator
- AChecker accessibility checker
- WebAIM Colour Contrast Checker
- Accessibility, News and You (BBC News Accessibility) - comprehensive resource on motivations for web accessibility and how to reach accessibility goals.

Multiple Pathways for Access

Often in web development and design, we create content in the same way *we* would access it and making the assumption that everyone else will do the same. However, this is rarely the case as each individual has their own strategies for accessing, parsing, and consuming information.

Multi-path Navigation

Content on the web should have multiple paths that lead to it. This enhances discoverability and accommodates multiple strategies of access.

For example, a textbook may be accessed different ways even though it is the same material. One reader may choose to refer to the table of contents, another may prefer to skim the pages looking at diagrams and images, while yet another may jump to the index looking for a particular topic.

Examples:

- a blog can have a search function, and a monthly archive.
- an online textbook can have a table of contents, index, and a search function.
- an online quiz can use a breadcrumb, next and previous links, and a summary at the end.

The navigation strategy should be cohesive and predictable so that users can easily get to where they need to every time they visit your site.

Multi-modal Content

This same principle of multiple pathways can be applied to actual content itself. By providing your content in multiple formats and modalities, you can enhance comprehension, retention, and learning.

For example, a table of national census data could be represented multiple ways which compliment each other. A diagram representing the country with statistics, a textual analysis of the data, along with the table can benefit many more people than if you used just a table of data alone.

Great Text

In building a web resource, emphasis is often placed on the visual aesthetics like colours, and the placement and size of a logo. These graphical elements are chosen with care and intention for optimal effect - this should also be true for the text present on your site.

Why Text is Important

For some visitors to your site, text is the only way they will understand what is before them. For other users, text is reassuring and unambiguous. Yet for others, text with visuals is a powerful combination for them to process and retain visual information. Whatever the reason, good text is important.

Text Complexity and Comprehension

Often how we write about a topic is similar to the way we want to read it, and we can forget that some readers are not at the same level. Therefore it's important to balance the substance of the text with the "usability" of the text.

Textual usability:

- Say what you need - not more or less.
- Use terms appropriate for your audience.
- Simpler terms are often better.

Text Labelling for a Mobile Age

- Aria-labelledby / describedby
- Describing buttons and links

Dealing with Custom Interactive Web Content

In this section the term "custom interactive web content" is used to describe situations when a combination of custom JavaScript and generic HTML elements are used to accomplish an interaction that is not possible using standard HTML. Some examples may be rather innocuous like a warning message that appears when a web form is incorrectly filled, to a more complex application like an online calendar.

The challenge with custom interactions is ensuring that someone using an assistive technology can perceive what is happening and have the necessary controls to accomplish the task.

Example: Flow Chart Web Application

A website has a drag-and-drop flow chart tool where the user can draw boxes, and arrows. For someone who is a sighted, mouse user this interaction wouldn't be much of a problem. But what if the user was not able to see the screen, or use a mouse?

- How would you describe all the shifting and rearranging of shapes when the user performs actions?
- How would a user interact with this if they only have access to a keyboard, or maybe a joystick with a single button?
- A sighted user can understand the state of the chart at a glance, what would the equivalent be for a non-sighted user?

Approaches to Dealing with Interactive Web Content

If the complex interactive is central to the experience and information, consider ways that the interactive can be adapted to other forms.

- Be prepared and willing to significantly change, modify, or even abandon ideas.
- Considering and building in multiple modes of access is more easily done early in design, than later in development.

Cognitive Load and Content Density

- front-loading instructions and cognitive burden
- simplification, on-demand information and contextual help
- avoid having the user recall information that is no longer accessible in their current context.

Managing Progress and Time

- Indicate number of steps, time commitment, etc up front
- Provide ways to save progress and resume later

Buttons, Links

Here are some best practices and general rules of thumb when using buttons and links:

- Links go somewhere, and buttons do something. For example, a logo in the site header that takes the user the front page should be a link. The magnifying glass image also in the header should be a button because it executes a search function.
- If you break from this function and have links that behave like buttons, and / or buttons that behave like links, you should also use an ARIA role to describe its function. This gets a lot more complicated and opens up possible confusion for users with assistive technology.

Styling buttons and links:

- Colour alone is not sufficient to indicate a button or a link. For example, regular text with a dark blue link may be hard to identify.
- Use some sort of physical indicator of links and buttons - such as an underline, a small arrow, an outline.

Dealing with hover, active, and focus states

Todo

Colours, Images, and Graphics

Dealing with Images and Captions:

- Use a <figure><caption><details> block for an accessible way to provide captions with an optional "reveal for more details" widget.

Example Figure with captions markup:

Figure caption with details

```
<figure role="group" aria-describedby="caption-01">
    <img src="./example.jpg" alt="Always have alt text.">
    <figcaption id="caption-01">
        <details>
            <summary class="caption">
                Brief caption text goes here. Can be a little longer than traditional Alt text.
            </summary>
            <p>
                A longer description with more details, links, and other elements can go in here.
            </p>
        </details>
    </figcaption>
</figure>
```

Figure with just a caption (and no extra details)

```
<figure role="group" aria-describedby="caption-01">
    <img src="./example.jpg" alt="Always have alt text.">
    <figcaption id="caption-01">
        <span class="caption">
            Brief caption text goes here. Can be a little longer than traditional Alt text.
        </span>
    </figcaption>
</figure>
```

Images without any labels (i.e. icons)
Often images are used without any accompanying text labels, as often done for icons. In these cases, images should still have valid alt text to describe the icon. Where icons are also links, an aria-label should be used on the anchor to describe the purpose of the image link.

Example:

Figure with just a caption (and no extra details)

```
<a href="https://twitter.com" aria-label="Go to Twitter"><img src="" alt="Twitter Icon"></a>
```

Consider using Scalable Vector Graphics (SVG)

For some graphics, it may be appropriate to use SVG image format. SVGs benefit from scaling seamlessly across different devices and resolutions, and can be styled using CSS for further customization.

Colour contrast:

- Check colour contrast with the Colour Contrast Analyzer https://developer.paciellogroup.com/resources/contrastanalyser/

Language Usage

- Simple language
- Avoid using language that implies certain physical or mental ability, or a particular modality. i.e. "click", "tap", "simply", "obviously", "top", "bottom", "red".