



Common Accessibility Mistakes and How to ~~Make~~ Avoid Them

July 14, 2018

Today's Agenda


- I. About Me
- II. Accessibility Principles
- III. Common Mistakes
- IV. Questions and Opinions



Ben Robertson

Front End Developer

benrobertson.io

 [@banquos_ghost](https://twitter.com/banquos_ghost)

Mediacurrent is a **full-service digital agency** that implements world class **open source software development, strategy and design** to achieve defined goals for **enterprise organizations seeking a better return on investment.**

Ben's Homegrown Web Accessibility Principles

Principle 1: Web Design > Graphic Design

The Three Tasks of Web Design

1. Write Good **Markup**
2. Use **CSS** to **enhance** the existing structure
3. **Layer** interactivity with **JavaScript**

Principle 2:

Be ASAP:
As Semantic As Possible

Every time you start typing
<div>...

Ask yourself:

Could I use a more semantic element?

HTML Elements Organized by Purpose

<http://bit.ly/semantic-html>

(MDN)

Content sectioning

Content sectioning elements allow you to organize the document content into logical pieces. Use the sectioning elements to create a broad outline for your page content, including header and footer navigation, and heading elements to identify sections of content.

Element	Description
<code><address></code>	The HTML <code><address></code> element indicates that the enclosed HTML provides contact information for a person or people, or for an organization.
<code><article></code>	The HTML <code><article></code> element represents a self-contained composition in a document, page, application, or site, which is intended to be independently distributable or reusable (e.g., in syndication). Examples include: a forum post, a magazine or newspaper article, or a blog entry.
<code><aside></code>	The HTML <code><aside></code> element represents a portion of a document whose content is only indirectly related to the document's main content.
<code><footer></code>	The HTML <code><footer></code> element represents a footer for its nearest sectioning content or sectioning root element. A footer typically contains information about the author of the section, copyright data or links to related documents.
<code><header></code>	The HTML <code><header></code> element represents introductory content, typically a group of introductory or navigational aids. It may contain some heading elements but also other elements like a logo, a search form, an author name, and so on.
<code><h1></code> , <code><h2></code> , <code><h3></code> , <code><h4></code> , <code><h5></code> , <code><h6></code>	The HTML <code><h1></code> - <code><h6></code> elements represent six levels of section headings. <code><h1></code> is the highest section level and <code><h6></code> is the lowest.

Alternatives to <div>

- <section>
- <article>
- or
- <header>
- <footer>
- <aside>
- <button>

List of all elements, organized by purpose (MDN):

<http://bit.ly/semantic-html>

Principle 3:

Web Sites Should
Look Good Naked



```
document.head.parentNode.removeChild(document.head);
```


The Naked Test: Bookmarklet

<http://bit.ly/remove-styles>

(StackExchange)



Sign in

with your Google Account

Email or phone

[Forgot email?](#)

Not your computer? Use Guest mode to sign in privately. [Learn more](#)

[Create account](#)

NEXT

English (United States) ▾

[Help](#)

[Privacy](#)

[Terms](#)

Sign in

with your Google Account

Email or phone

Forgot email?



Type the text you hear or see

Not your computer? Use Guest mode to sign in privately.

[Learn more](#)

Next

Create account

Afrikaans

azərbaycan

català

Čeština

Dansk

Deutsch

eesti

English (United Kingdom)

English (United States)

Español (España)

Nice heading.

Where are the buttons?

Label order?

Next / Create Account
Swapped Position

The Naked Test: What to Look For

- Does the structure of the site make sense?
- Is the content organized?
- Do interactive elements look interactive?
- Do inputs have labels?

Principle 4: Talk to Your Computer

ARIA Attributes

- **aria-label:** What is this element?
- **aria-expanded:** Is this element open or closed?
- **aria-describedby:** What element describes this element?
- **aria-live:** Will this element be updated later?

The Principles: A Refresher

Web Design > Graphic Design

Write Good Markup.

Enhance good markup with CSS.

Layer interactivity with JavaScript.

As Semantic As Possible

Avoid <div>s when possible.

Always use <button>.

Websites should look good naked.

Remove CSS to check markup.

Talk to your computer.

Use ARIA attributes to give the browser / screen reader more context.

Common Mistakes

Missing Title

- Every page needs a `<title>Page Title</title>`
- It should be informative
- It should be unique on your site.

Poor Heading structure

- Headings: h1, h2, h3, h4, h5, h6
- Form the outline of a web page
- Should be used in order - don't skip one!
- Assist readability / scannability

Etymology

The word "animal" comes from the Latin *animalis*, meaning *having breath*. In colloquial usage, as a consequence of [anthropocentrism](#), the term *anim*

Characteristics

Animals have several characteristics that set them apart from other living things. Unlike bacteria, which are [prokaryotic](#), and unlike [protists](#), which are eukaryotes, animals [produce their own nutrients](#)^[8] animals are [heterotrophic](#),^{[7][9]} feeding on other organisms. In a few exceptions, animals breathe oxygen and [respire aerobically](#).^[11] All animals (with a few exceptions) breathe (or respire) during at least part of their [life cycle](#), but some animals, such as [sponges](#), become [sessile](#). The [blastula](#) is a stage in [embryonic development](#) that is [differentiated](#) into specialised tissues and organs.

Structure

All animals are composed of cells, surrounded by a characteristic [extracellular matrix](#) and [glycoproteins](#).^[14] During development, the animal extracellular matrix forms a variety of complex structures possible. This may be calcified, forming [skeletons](#) (primarily algae, plants, and fungi) are held in place by cell walls, and so do [gap junctions](#), and [desmosomes](#).^[17]

With few exceptions—in particular, the sponges and [placozoans](#)—animals have [tissues](#), which transmit signals and coordinate the body. Typically, there is a [gut](#) (in [deuterostomes](#)).^[19]

`Read More`

- Read more what?
- Read more where?
- Links text should tell where or what a user is clicking on
- Solution: Add hidden text
- **`{{ title }}`**

Inputs Missing a <label>

- Most common mistake is using a placeholder value instead of a value
- Designers love this:



- Wrong way: `<input placeholder="Search Google or type URL" />`
- You can achieve this using the "float label" pattern (<http://bit.ly/float-labels>)
- Every input needs a real <label> tag, associated with an input
- `<label for="email-field">Email Address</label>`

CSS Grid / Flexbox: { order: ?? }

- `order` lets you reorder flex items or grid items
- Only visual reordering
- Creates a mismatch between logical ordering / visual order
- Go back and update the source order

Images and ALT attributes

- Always include the alt attribute on an image
- ALT = Alternative
- ``
- Is the image content, or merely decorative?
- It may be fine to leave the alt attribute blank

Empty ALT

```
<div class="card">
  <h2>Ben Robertson</h2>
  
  <p>Ben Robertson is a front end developer at Mediacurrent.</p>
  <a href="/ben-robertson">View Ben Robertson's Profile</a>
</div>
```

Empty ALT #2

```
<button>
  
  <span>Close Menu</span>
</button>
```

:focus { outline: none; }

- Focus styles are for people who aren't using a mouse
- They help show a user where they are currently interacting with the interface
- Don't remove them, style them!
- Limit them to keyboard-only interactions with :focus-visible (polyfill: <http://bit.ly/focus-ring>)

Missing keyboard functionality

- Interactivity needs to be triggered via **mouse and keyboard**
- Use the <button> element
- Other common keyboard considerations:
 - Exiting the current component
 - Submitting
 - Moving position / browsing

For other keyboard functionality, listen to the keyup or keydown event.

<http://keycode.info>


```
document.addEventListener('keyup', (event) => {
  switch (event.keyCode) {
    // escape
    case 27:
      // exit
      break;
    // enter || spacebar
    case 13 || 32:
      // submit or something
      break;
    // left arrow
    case 37:
      // move back / previous
      break;
    // right arrow
    case 39:
      // move forward
      break;
    // up arrow
    case 38:
      // move up
      break;
    // down arrow
    case 40:
      // move down
      break;
  }
}
```

Hiding Things

- So many ways to hide things
- The mistake is to hide something visually but not from tabindex or screen readers
- Methods of hiding visually and from tabindex / screen readers:
 - `display: none`
 - HTML hidden attribute: `<p hidden>Hidden Text</p>`
- Hiding from screen-readers (combine with other CSS):
 - `aria-hidden="true"`

Hiding & aria-expanded

- Indicates expanded / collapsed
- `<button aria-expanded="false">A Button</button>`



```
button.setAttribute('aria-expanded', 'false');  
button.setAttribute('aria-expanded', 'true');
```




Thank you!



@Mediacurrent



facebook.com/mediacurrent



Mediacurrent.com