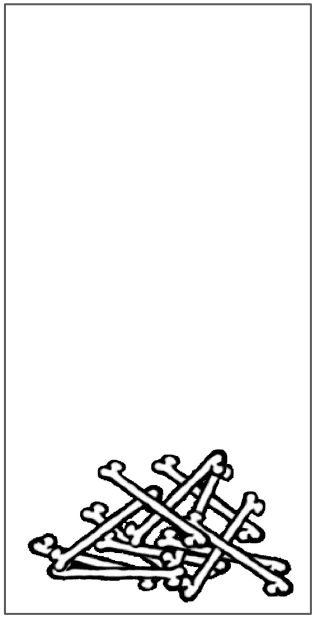
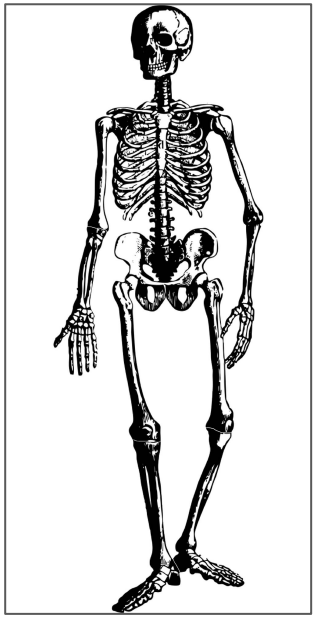
More HTML, Accessible Design, and CSS Preview

Lecture 2

Review: Websites



WORDS + IMAGES



HTML

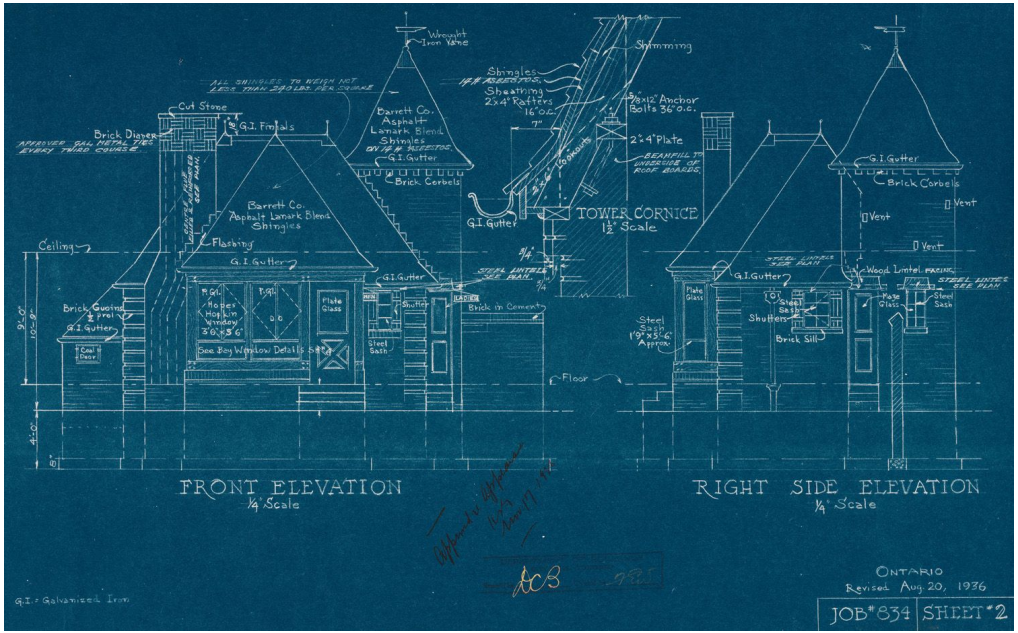
CSS

JAVASCRIPT

Ok, but really… Web? Internet? The same, right?

Nope! An analogy...

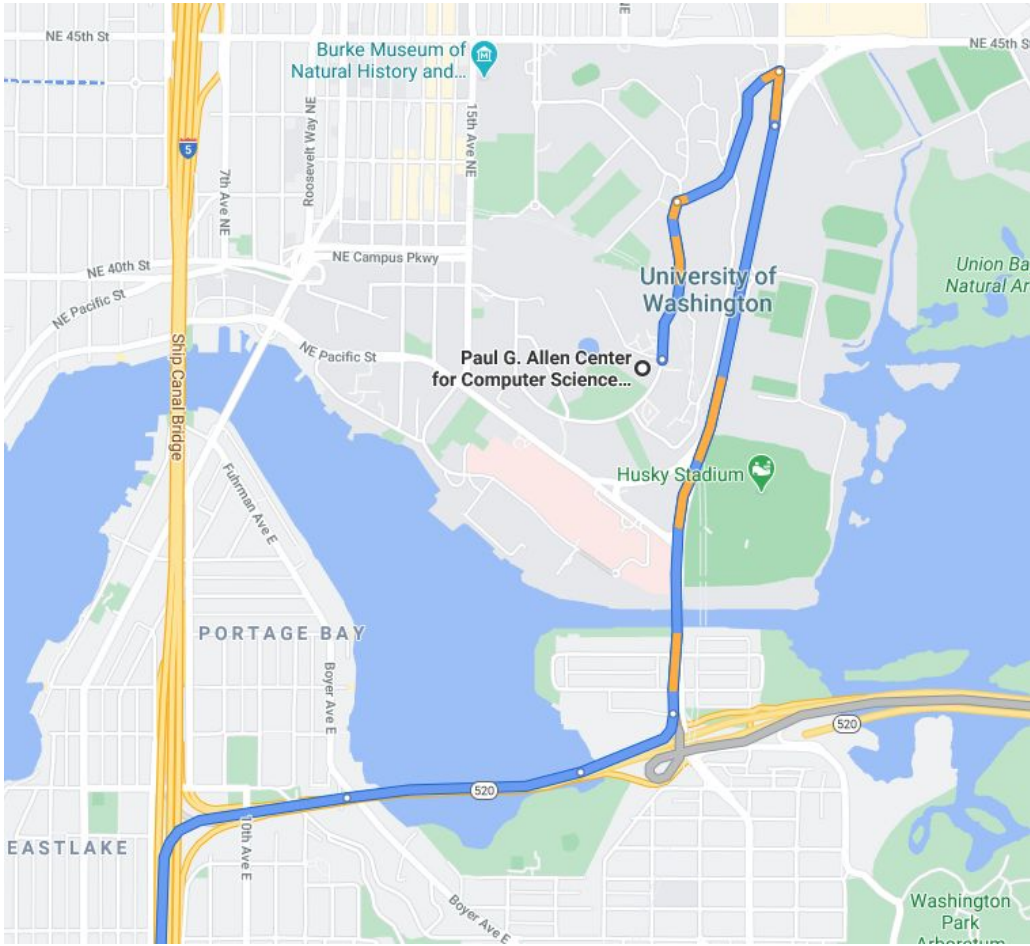
Where does the house come from?



Getting to the House

Usually:

- Get the address
- Ask someone for directions from your address to the House's address
- Series of instructions:
  - Turn here, follow until there
  - Continue until you reach this
  - Look for House
- Follow directions



In this analogy...

The Internet is kind of like:

- The roads you take to get from one place to another
- Plus the related tools to make using those roads possible (cars, signs, traffic control, GPS, etc.)

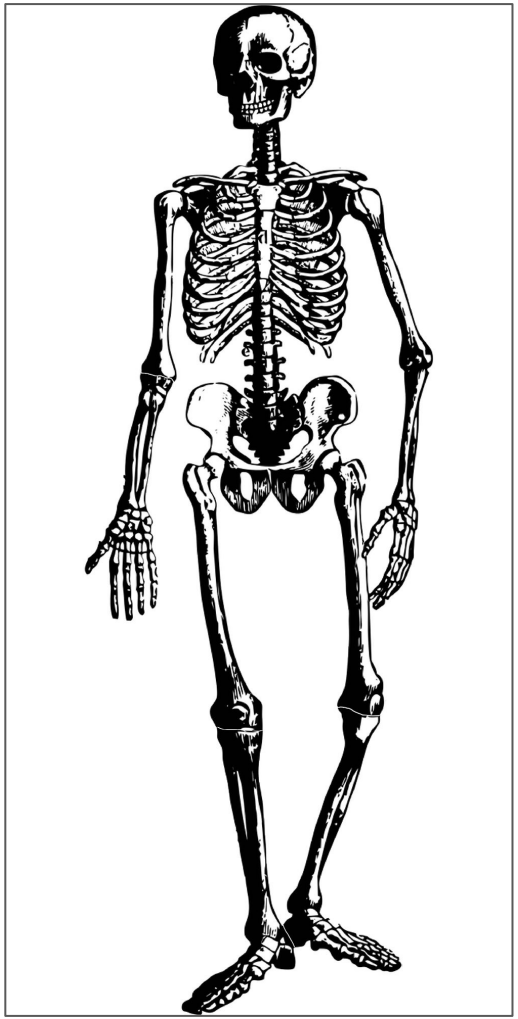
The Web is kind of like:

- References to houses or businesses, and the things you see inside of them.
- "If you want to talk to Alex, you can find them at 123 Street Way."

Refining the analogy

We'd never do this for real houses, but the web is more like:

1. Go to the house's address
2. Ask whoever answers for the current blueprints to the house
3. Go back to your own home/residence
4. Build the thing the blueprints represent
    - (Might need to go back to the other house to pick up some speciality supplies.)
5. After you're done looking at or using it, throw it away.



HTML

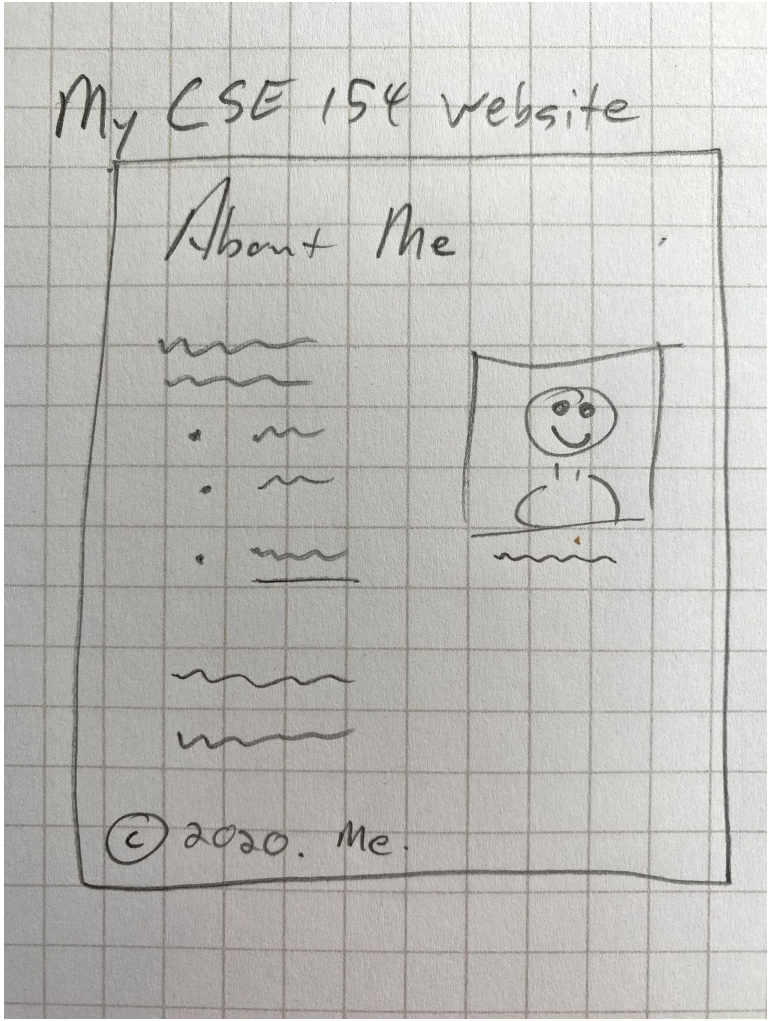
Like "blueprints" throughout the internet.

Tips when drafting HTML/CSS web pages

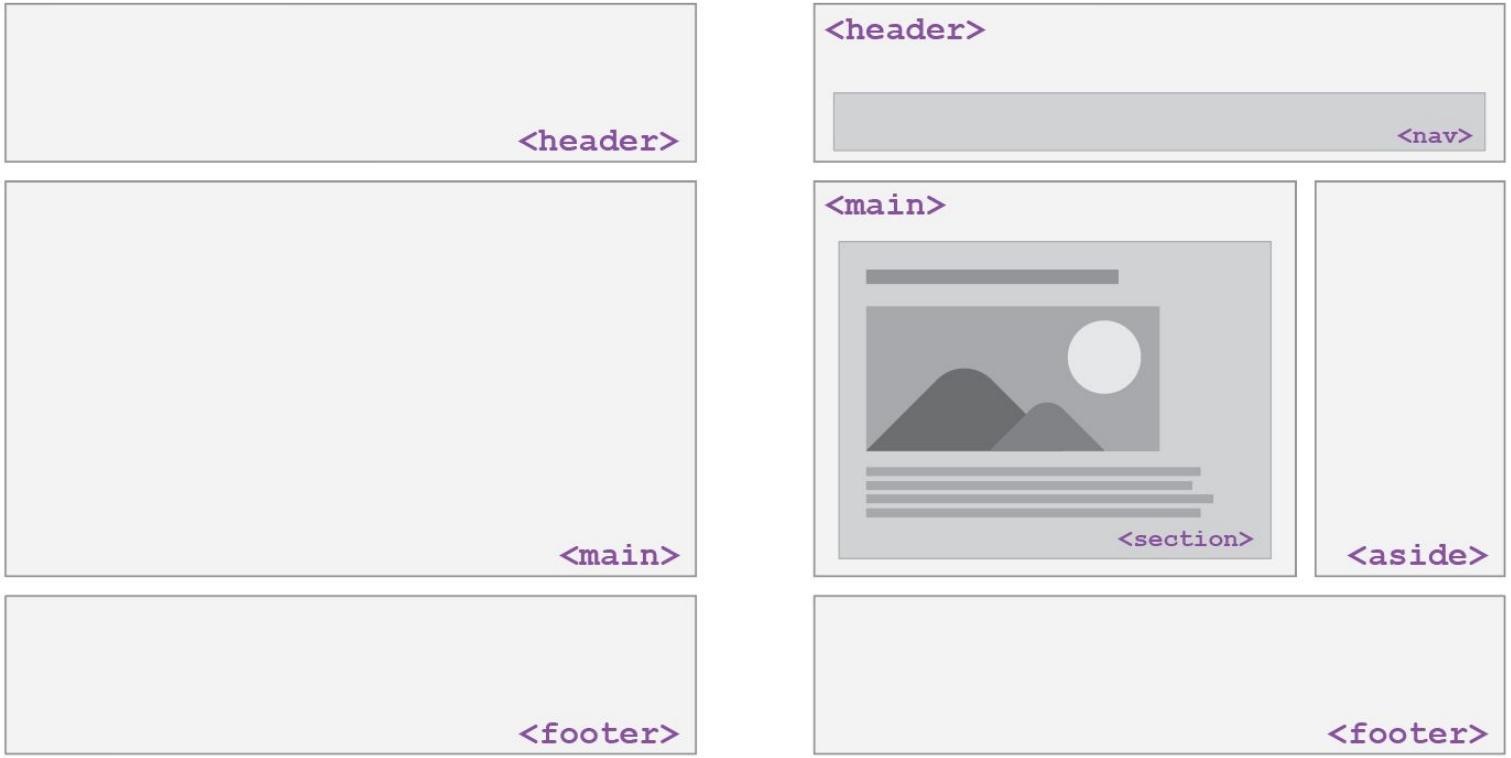
Always start with a sketch/wireframe before jumping into code!

A great resource on getting started with wireframes can be found here.

You don’t need to be an artist.



HTML5 Semantic Tags to Define Structure



Structure of an HTML page

An HTML page is saved into a file ending with extension `.html`

The `<head>` tag describes the page and the `<body>` tag contains the page's content

The `DOCTYPE` tag tells the browser to interpret our page's code as HTML5, the latest/greatest version of the language

```
<!DOCTYPE html>
<html>
  <head>
    information about the page
  </head>
  <body>
    page contents
  </body>
</html>
```

General Outline with HTML5

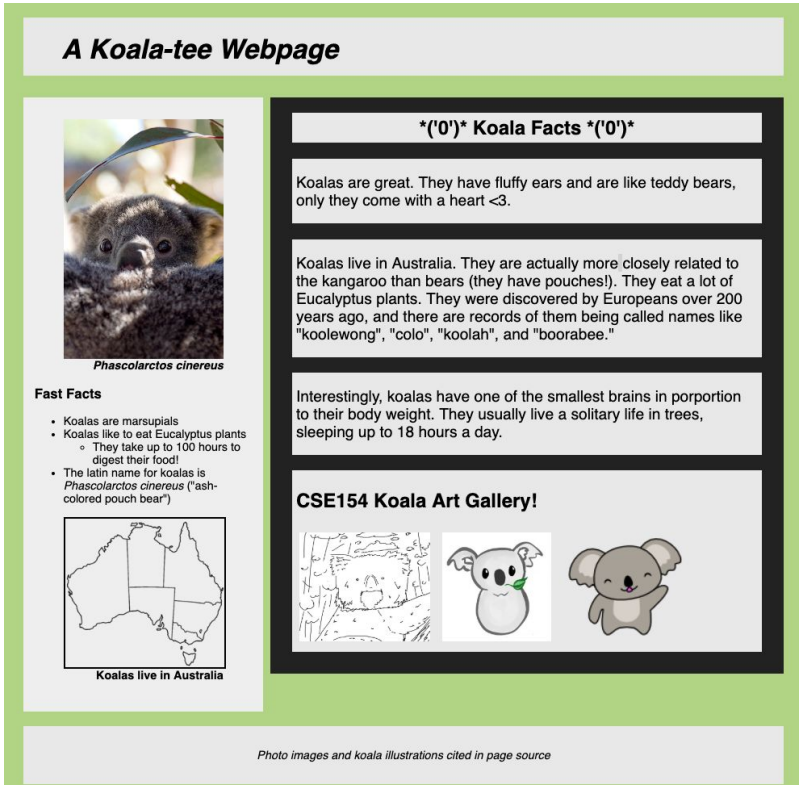
```
<body>
  <header>
    <!-- Header of the webpage body (e.g. logo, navigation bar) -->
  </header>
  <main>
    <!-- Main section of the webpage body (where most content is) -->
  </main>
  <footer>
    <!-- Footer of the webpage body (e.g. copyright info) -->
  </footer>
</body>
```

For different types of pages, you may have more elements but there are the ones you should follow as a guide for most of your web pages.

HTML vs. Rendered Web Page

```
<!DOCTYPE html>
<html>
  <head>
    <title>Koala Fan Page</title>
  </head>
  <body>
    <header>
      <h1>A Koala-tee Webpage</h1>
    </header>
    <main>
      <aside>
        <!-- Left sidebar -->
      </aside>
      <section>
        <!-- Koala facts (header and paragraphs)-->

        <article>
          <!-- Koala art gallery -->
        </article>
      </section>
    </main>
    <footer>
      <!-- Image citations -->
    </footer>
  </body>
</html>
```



HTML5 and Semantic Tags

`<main>`
Main content of the document - unlike <header> and <footer> tags, there can only be one main element in the <body>. The content inside should be unique and not contain content that is repeated across pages (e.g. sidebars, nav link, search bars, etc.)

`<header>`
Header element - contains header information for page body or section/article, including logos, navigation bar, etc.

`<footer>`
Footer element - contains footer information for page body or section/article, including copyright information, contact, links, etc. Also often used with block quotes to cite sources (see CP1 about.html for an example!).

`article` vs. `section`

We get this question a LOT
Others ask this too
Here are two resources to help you:

- Ian Devlin article (a course reading)
- YouTube video

`Articles` are complete, standalone content. `Sections` are pieces of a greater whole.

And remember: `div` has no semantic meaning, should only be added for selecting content in CSS/JS, and should be your "last resort"

Some important HTML Details

Links (Anchors): <a>

Links, or “anchors”, to other pages (inline)

```
<p>
   Search for it on <a href="http://www.google.com/">Google</a>!
</p>
```

code

Search for it on Google!

output

Uses the `href` (Hypertext REFerence) attribute to specify the destination URL

- This can be absolute (to another web site) or relative (to another page on this site)

Anchors are inline elements; must be placed in a block element such as a `<p>` or `<h1>`

Images: <img>

Inserts a graphical image onto the page (inline)

```
<img src="img/koala-with-leaf.png" alt="A Koala with a leaf" title="Logo">
```

code

output

The `src` attribute specifies the image URL

Motivating `alt` text

HTML5 also requires an `alt` attribute describing the image, which improves accessibility for users who can't otherwise see it.
The value of the `alt` attribute is also what you see if the image is not successfully loaded.

```
<img src="img/koala-with-leaf-broken.png" alt="A Koala with a leaf" title="Logo">
```

code

output

More about Images

If placed in an <a> anchor tag, the image becomes a link

```
<a href="https://courses.cs.washington.edu/courses/cse154/20au/">
  <img src="img/cse154logo.png" alt="CSE154 Course Logo" title="Logo">
</a>
```

code

output

Relative vs. Absolute Paths for Links and Images

Relative: paths are relative to the document linking to the path.

- Linked files within the same directory: “filename.jpg”

```
<a href="my-other-page.html">Check out my other page!</a>
```

- Linked files within a subdirectory (e.g. “img”): “img/filename.jpg”

```
<img src="img/koala-with-leaf.png" alt="A Koala with a leaf"
title="Logo">
```

Absolute: paths refer to a specific location of a file, *including the domain and protocol.*

- Typically used when pointing to a link that is published online (not within your own website).
- Example: "https://validator.w3.org/"

Citing External Material

```
<figure>
  <!--
    Image source: Wikipedia, Made by User:Golbez
    [CC BY-SA 3.0 (http://creativecommons.org/licenses/by-sa/3.0/)]
  -->
  <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/3/3b/Australia_states_blank.pn
g/257px-Australia_states_blank.png"
       alt="Koala-land map">
  <figcaption>Koalas live in Australia</figcaption>
</figure>
```

In your CP's, you must cite all resources that were not original (and you should give your own images credits) either on the page (as in about.html or in a footer) and/or in the page source code.

Nested Lists

A list can contain another list:

```
<ul>
  <li>Koalas are marsupials</li>
  <li>Koalas like to eat Eucalyptus plants
    <ul>
      <li>
        They take up to 100 hours to
digest their food!
      </li>
    </ul>
  </li>
  <li>
    The latin name for koalas is
    <em>Phascolarctos cinereus</em>
    ("ash-colored pouch bear")
  </li>
</ul>
```

Fast Facts

- Koalas are marsupials
- Koalas like to eat Eucalyptus plants
    - They take up to 100 hours to digest their food!
- The latin name for koalas is *Phascolarctos cinereus* ("ash-colored pouch bear")

HTML Character Entities

A way of representing any Unicode character within a web page

character(s)	entity
< >	< >
é è ñ	é è ñ
™ ©	™ ©
π δ Δ	π δ Δ
И	И
" &	" &

- A complete list of HTML entities
- How you you display the text `&` on a web page?

Block and Inline Elements (explanation)

Block elements contain an entire large region of content

- Examples: paragraphs, lists, table cells
- The browser places a margin of whitespace between block elements for separation

Inline elements affect a small amount of content

- Examples: bold text, code fragments, images
- The browser allows many inline elements to appear on the same line
- Must be nested inside a block element

Block and Inline Elements: example

```
<em>text</em>
<em>text</em>
<em>text</em>
<p>text</p>
<p>text</p>
<p>text</p>
```

code

text text text
text
text
text

output

Rules and Exceptions

Block vs. inline:

- Some block elements can contain only other block elements: `<body>, <form>`
- `<p>` tags can contain only inline elements and plain text
- Some block elements can contain either: `<div>, <li>`

Some elements are only allowed to contain certain other elements

- `<ul>` is only allowed to contain `<li>` (but <li> can contain <ul> for nested lists!)

Some elements are only allowed once per document:

- `<html>, <body>, <head>, <main>`

Nesting Tags

Tags can “nest” inside of other tags

```
<body>
  <p>
    This is a <em>really, <strong>REALLY</strong></em> awesome paragraph.
    And here's a neat list:
  </p>
  <ol>
    <li>with one list item</li>
    <li>with another list item</li>
  </ol>
</body>
```

code

This is a *really*, ***REALLY*** awesome paragraph. And here's a neat list...

1. with one list item
2. and another list item!

output

Incorrectly Nesting Tags

```
<p>
  HTML is <em>really,
  <strong>REALLY<em class="bad"></em></em> lots of</strong> fun!
</p>
```

incorrect

Tags must be correctly nested

- A closing tag must match the most recently opened tag
- The browser may render it correctly anyway, but it is invalid HTML

How would we get the above effect in a valid way?

```
<p>
  HTML is <em>really,
  <strong>REALLY lots of</strong><em class="good"></em></em> fun!
</p>
```

correct

How can we check? Gitlab HTML Validator

Gitlab Validation Guide

- Checks your HTML code to make sure it follows our official HTML syntax
- More picky than the browser, which may render bad HTML correctly

NOTE: To be eligible for full credit on your creative projects and homework assignments you **MUST** validate all of your files and pass with no errors. Warnings are ok.

Web Standards

Moreover, it is important to write proper HTML code and follow proper syntax

Why use valid HTML5 and web standards?

- More interoperable across different web browsers
- More likely that our pages will display correctly now and in the future
- To ensure accessibility

Accessible Design

Slides based on content from Profs. Richard Ladner, Jake Wobbrock, and Amy Ko.

This is another great resource to learn more about why/how to make websites accessible!

Thinking about accessibility as web developers

Who uses the web?
What are the different ways people visit and interact with websites?
Why is it important to think about users when developing websites?

Disabilities

- Everyone has different abilities
- Nearly 1 in 5 people have a disability in the U.S. (from the U.S. Census)
- Some kinds of disabilities (from W3C Web Accessibility Initiative (WAI)):
  - Visual
  - Auditory
  - Speech
  - Physical
  - Cognitive, learning, and neurological
  - Behavioral

Temporary and Situational Disability

Disabilities can be temporary

- having a broken arm in a cast
- difficulty hearing after a loud concert

Disabilities can be situational

- trying to open your door while carrying groceries
- trying to talk on the phone in a noisy room
- trying to read your phone under direct sunlight
- it's raining (in Seattle?!?), and touchscreen doesn't work

Disability affects all of us

Accessible Design

Designs that account for all abilities are called accessible designs

Exercise for after lecture: Try your phone’s screen reader!

Enable your phone's screen reader

- iOS: Settings > General > Accessibility > VoiceOver > Hit the switch
- Android: Settings > Accessibility > Talkback > Hit the switch

Input works differently now. For example, tap now reads the screen and double-tap selects. Use two or three fingers to scroll by page. Play with it for a minute.

Try closing your eyes and reading a webpage or a social networking site. Try writing an email.

Views of disabilities

Medical view

- People with disabilities are patients who need treatment and/or cure.

Legal view

- People with disabilities have rights and responsibilities, such as access to public buildings, voting, education, etc.
- Lawsuits can occur, but they should not be the motivating factor for making a system accessible

Sociocultural view

- Variation in ability is natural. "Disability" is caused by how society is designed, not by nature.
- Building for inclusion builds innovation (e.g., curb cuts, closed captioning)

Tools and Resources

From the A11y Project

- A really great compendium of resources
- An accessibility workshop from GHC'18

Tools

- Web Accessability Evaluation Tool: http://wave.webaim.org/
- Color Schemes: http://colorbrewer2.org/
- Color blindness checker: http://www.color-blindness.com/coblis-color-blindness-simulator/
- Text readability: http://juicystudio.com/services/readability.php

Resources

- Web Content Accessibility Guidelines (something to know about when you apply for jobs): https://www.w3.org/WAI/intro/wcag
- Teach Access Tutorial (general background and covers an important standard called ARIA). http://teachaccess.org/initiatives/tutorial/
- Web design and development course by AccessComputing http://www.washington.edu/accesscomputing/webd2/
- A11ycast - YouTube Videos to teach developers how accessibility works.

Accessible Web Design Principles

- Use document structure (Semantic) tags: e.g., `<article>`, `<strong>`
- Don't use deprecated style tags like `<b>`
- Provide metadata: e.g., `<html lang="en">`
- Provide alternatives: e.g., img `alt` tag, video captions, transcripts, allow both keyboard and mouse input
- Avoid directional text: eg. "the diagram on the right shows..."

Note: These design principles help in other ways as well

- Captions allow people to watch your video without turning sound on.
- Transcripts help people find your page through Google.
- Structure and metadata help programs understand your page.

More about HTML and accessibility here.

Preview to CSS

The Bad Way to Produce Styles

```
<p>
  <font face="Arial">Welcome to Greasy Joe's.</font>
  You will <b>never</b>, <i>ever</i>, <u>EVER</u> beat
  <font size="+4" color="red">OUR</font> prices!
</p>
```

code

Welcome to Greasy Joe's. You will **never**, *ever*, EVER beat OUR prices!

output

Tags such as `b, i, u` and `font` are discouraged in strict HTML

They are bad because of:

- Accessibility
- Code organization
- Changing style easily

Cascading Style Sheets (CSS): <link>

```
<head>
  ...
  <link href="filename" rel="stylesheet">
  ...
</head>
```

- **CSS** describes the appearance and layout of information on a web page (as opposed to HTML, which describes the content)
- Can be embedded in HTML or placed into separate `.css` file (preferred)

Basic CSS Rule Syntax

```
selector {
  property: value;
  property: value;
  ...
  property: value;
}
```

template

```
p {
  color: red;
  font-family: sans-serif;
}
```

example

- A CSS file consists of one or more rules
- A rule selector specifies HTML element(s) and applies style properties