

HTML



CSS



HTML & CSS: FUNDAMENTALS OF DEVELOPMENT

Instructor: Beck Johnson
Week 1



INTRODUCTIONS

- Who are you? Who am I?
- What do you do/study/etc?
- What is your experience with web development?
 - Do you have related skills like Photoshop, Wireframing, E-mail Marketing, etc?
- What are you hoping to get out of this class?

CLASS SCHEDULE

- Five sessions
- Thursdays from May 10 to June 7, from 6:30-9:30 p.m.
- 10 minute break somewhere in the middle
- No grades, no tests
- Questions and feedback highly encouraged!



COURSE OVERVIEW

- Basic rules of HTML and CSS
- Using CSS to style web pages
- Website structure, navigation, and file organization
- Hosting, FTP, Github
- Overview of Javascript and jQuery
- The role of the developer in the product lifecycle



TODAY

- Overview of a website
- Code editors
- Basic HTML
- Overview of CSS – font styling, colors, alignment
- What does a developer DO anyway?



ODDS & ENDS

kweeket.github.io/dev-101

Slides, sample files, “homework”, and interesting links will be posted here



OVERVIEW OF A WEBSITE

CONTENT, DESIGN, & CODE



CONTENT

Most important part
of any website

HTML



DESIGN

Critical to the best user
experience

CSS



CODE

Brings content and
design to life

JAVASCRIPT



What am I presenting?

TEXT

- Articles
- Links
- Lists

MEDIA

- Images
- Videos
- Audio

Content is entered using HTML



What is the experience?

USER EXPERIENCE

- Layout
- Navigation
- User flows

GRAPHIC DESIGN

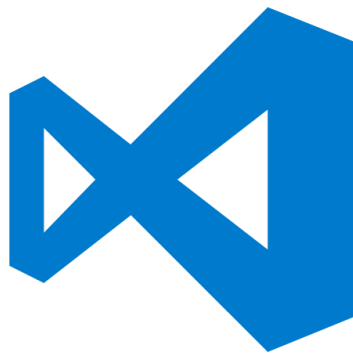
- Colors
- Fonts
- Backgrounds

Design is done with CSS



CODE EDITING TOOLS

CODE EDITORS



VS Code



Brackets



Atom



Sublime Text



Coda

HTML is just text

You can right-click and select "View Source" on any webpage to see how the developer made it

WEB BROWSERS



HTML and CSS require testing in all major modern browsers and devices

You can experiment directly in the browser before making permanent changes



DEVELOPER TOOLS

Chrome

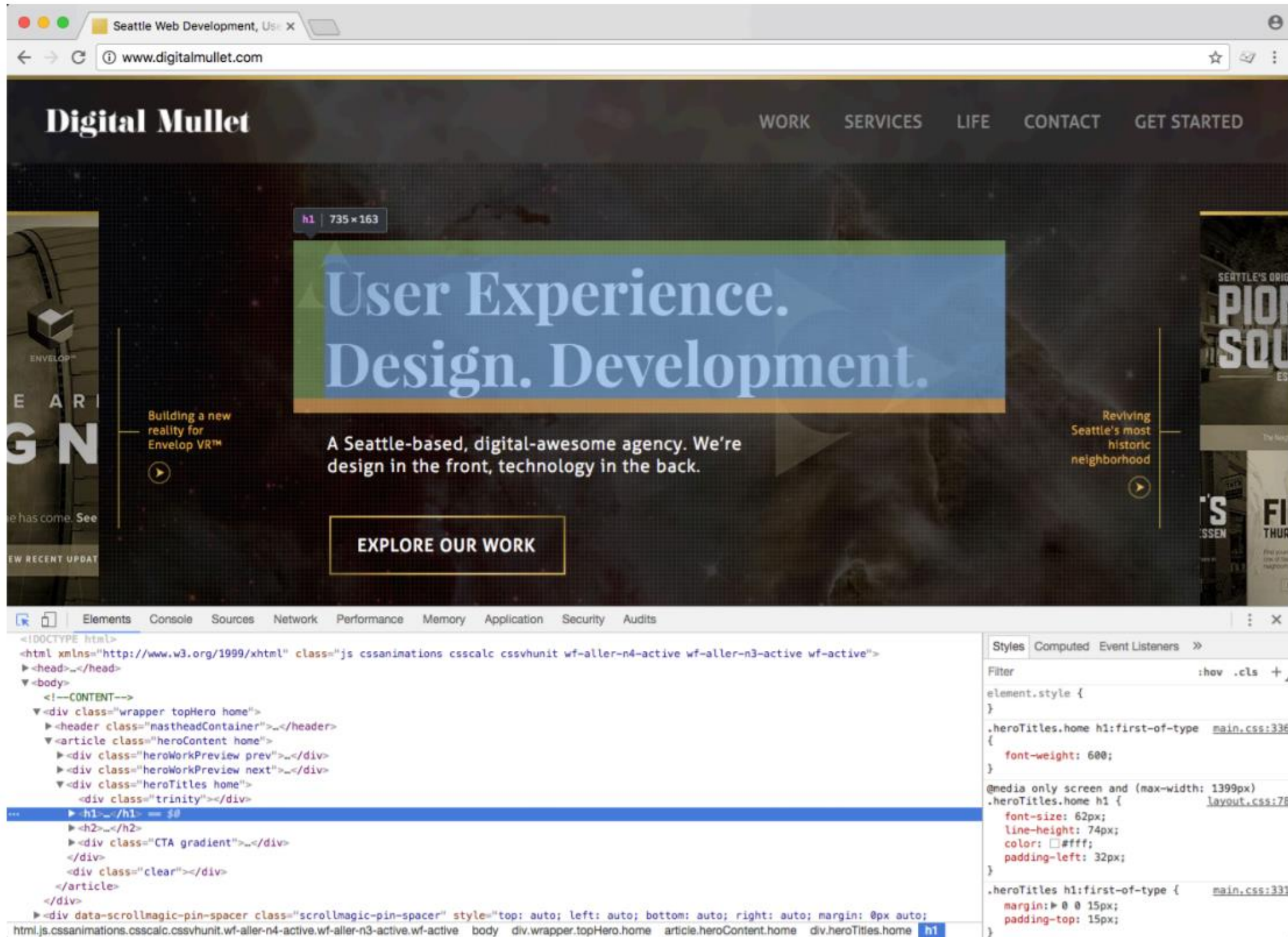
- Right-click > Inspect
- OR hit the F12 key

Safari

- Open Preferences > Advanced > Show Develop menu
- Right-click > Inspect Element

Internet Explorer

- F12 key





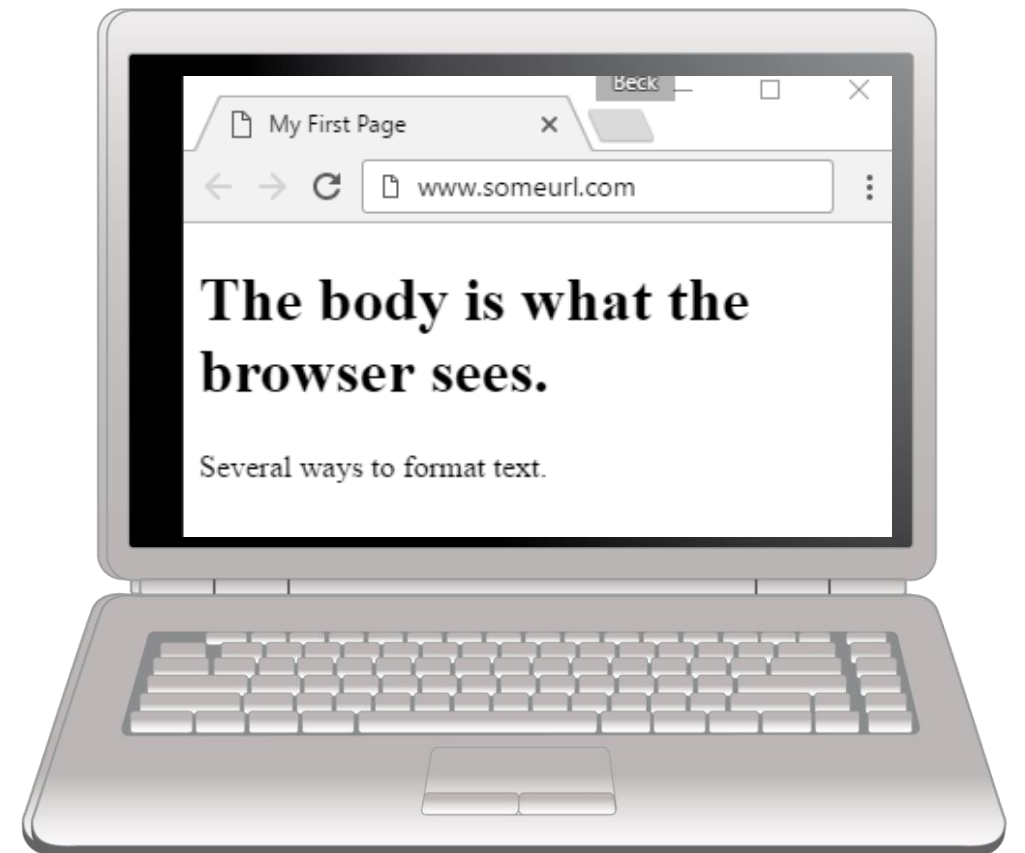
LET'S TRY IT

<html>

HTML DOCUMENTS

HTML DOCUMENT

```
<!doctype html>
<html>
<head>
  <meta charset="UTF-8">
  <title>My First Page</title>
</head>
<body>
  <h1>The body is what the browser sees.</h1>
  <p>Several ways to format text.</p>
</body>
</html>
```



HTML ELEMENTS

- HTML elements are contained in <> brackets
- Most HTML tags have an **opening** tag and a **closing** tag

<tag>Content goes in here</tag>

- Some types of tags are “self-closing”

<tag />

HTML ELEMENTS



HTML RULES

- Tags are written in lowercase

`<a>` not `<A>`

- Tags **must** be closed

`<p>Text in here.</p>`

`<div>Content in here.</div>`

`
` Self-closing line break

DOCTYPE

<!doctype html>

- The very first thing in any HTML document
- Tells the browser what version of HTML the document is written in (this one is HTML5)

DOCTYPE

These other doctypes are not commonly in use anymore:

```
<!doctype html PUBLIC "-//W3C//DTD XHTML 1.0  
Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-  
strict.dtd">
```

```
<!doctype html PUBLIC "-//W3C//DTD HTML 4.01  
Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/transitional.dtd">
```


HTML DECLARATION

<html>

- The top line after <doctype> declaration.
- Tells the browser “This is where everything starts!”

```
<html><!-- everything else --></html>
```

<!--HTML COMMENTS-->

<!-- Comments are great -->

- Are not visible to the user in their browser
- Great for leaving notes for yourself or other developers
- Can be seen in “view source”

<!--HTML COMMENTS-->

Sometimes they don't really have a point...

HEAD ELEMENT

`<head></head>`

- **Required** for a valid HTML document
- Holds information about the document that is (mostly) not visible to the user
- Can contain CSS and Javascript

```
<head>
```

```
  <!-- metadata and resources -->
```

```
</head>
```

META TAGS

`<meta charset="UTF-8"/>`

- Used to specify "meta" information to the browser like page description, author, search engine keywords, and character encoding
- UTF-8 represents Unicode, a system to handle text consistently in a variety of languages.

`<head>`

`<meta name="author" content="Your Name" />`

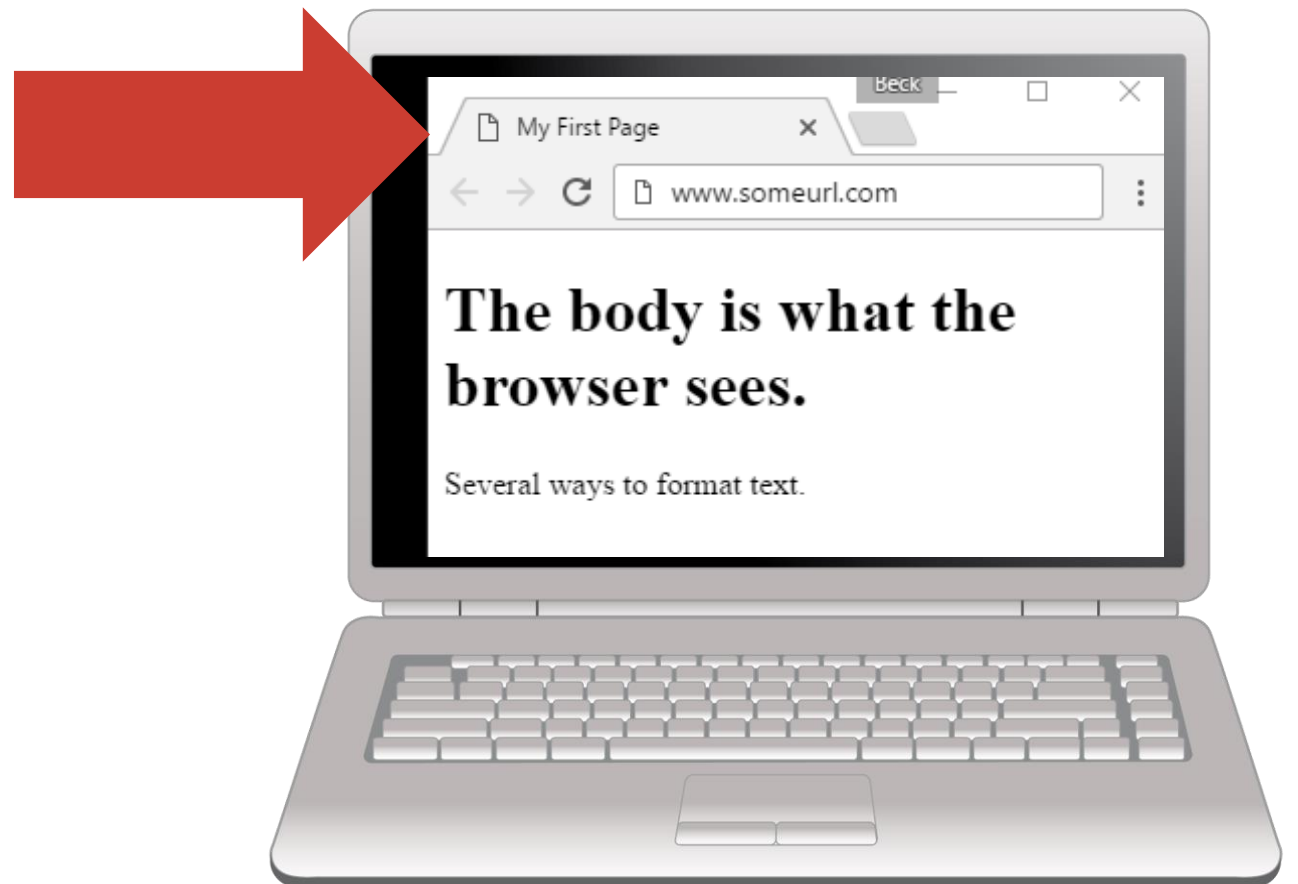
`<meta name="description" content="A thrilling page"/>`

`</head>`

TITLE TAG

`<title>My First Page</title>`

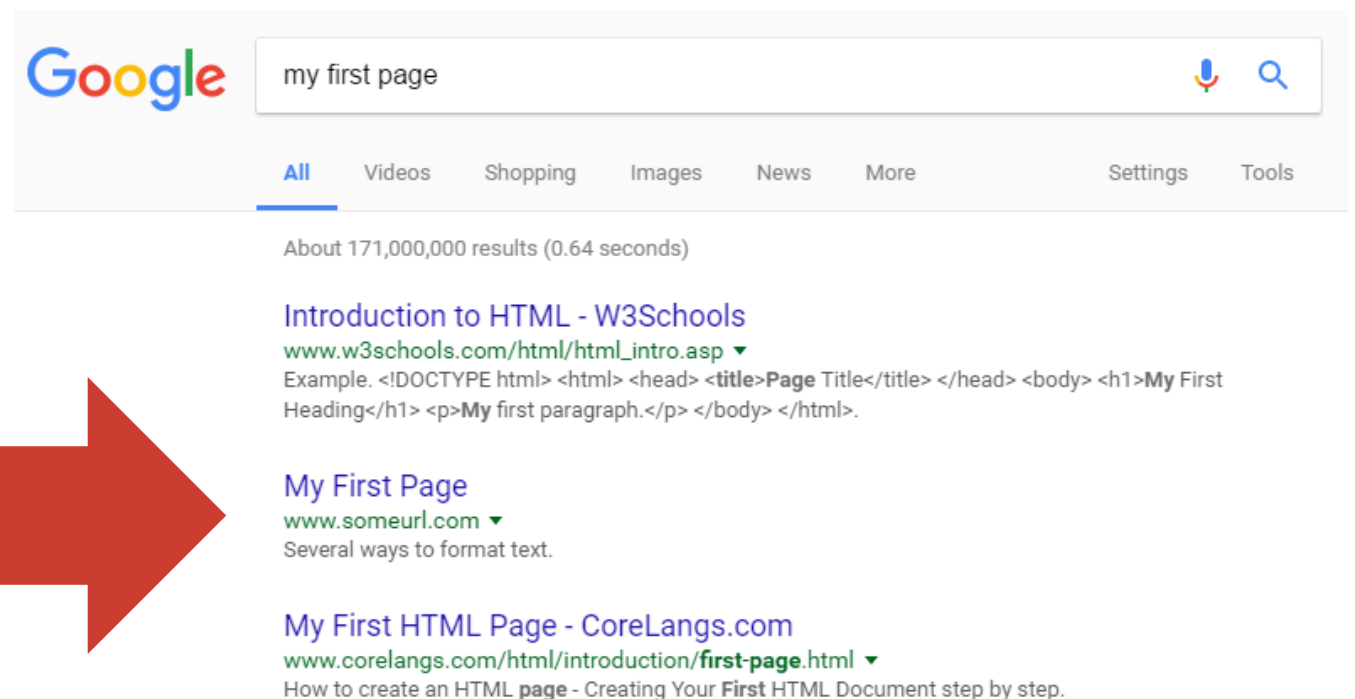
- Displays in the browser tab
- **Required** inside `<head>`



TITLE TAG

`<title>My First Page</title>`

- Name of the page when page is bookmarked
- The title for the page in search results on Google (or Bing)



BODY ELEMENT

`<body></body>`

The part of the HTML document that's visible to the user

- Contains all content of the document, such as tags, links, images, tables, etc.

`<body>`

`<!-- all my sweet content -->`

`</body>`

MAJOR BODY ELEMENTS

- **Headings** for dividing up your page and content
- **Paragraphs** of text
- Bulleted, ordered, unordered **lists**
- **Images**
- **Links** to other pages, websites, or resources.

HEADINGS

`<h1>`My Page Header`</h1>`

Headings range from most important to least important

`<h1>` to `<h6>`

Search engines use `<h1>` to determine important information about the page

HEADINGS

<h1>Heading 1</h1>

<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

<h5>Heading 5</h5>

<h6>Heading 6</h6>

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

PARAGRAPHS

<p>Hi! I'm a paragraph!</p>

- Browsers automatically add space around <p> elements (although this can be changed with CSS)

LAYOUT

- `<header>` wraps header content
- `<footer>` wraps footer content
- `<nav>` indicates that everything inside is related to navigation
- `<section>` is used to define content sections

FORMATTING

`` indicates *emphasis*

- By default, this displays as `italic`

`` indicates **importance**

- By default, this displays as `bold`

LIST ELEMENTS

```
<ul>  
  <li>Puppies</li>  
  <li>Kittens</li>  
</ul>
```

Unordered lists `` appear in the browser by default with **bullets**

- Puppies
- Kittens

LIST ELEMENTS

```
<ol>  
  <li>Puppies</li>  
  <li>Kittens</li>  
</ol>
```

Ordered lists `` appear in the browser by default with **numbers**

1. Puppies
2. Kittens

LIST ELEMENTS

```
<ul>  
  <li>Puppies</li>  
  <li>Kittens</li>  
</ul>
```

Both unordered and ordered lists can only contain **list items** `` directly

IMAGES

```

```

- Images do not have a closing tag
- Images have two required **attributes**:
 - **src** is where the file lives (local or external)
 - **alt** is a description of the image (used for screen readers, search engines, etc)

IMAGES

```

```

- **height** and **width** resize images and ensure the page doesn't jump
- **title** is shown as a tooltip in some browsers when you hover your mouse over the image



LINKS WITH THE ANCHOR TAG

```
<a href="http://google.com">Google</a>
```

The `<a>` element defines an "anchor" or link

- Anything inside `<a>` is clickable – can be text, an image, or any other valid HTML

SOME <A>TTIBUTES

```
<a href="http://google.com" title="Search"  
target="_blank">Google</a>
```

- **href** is the URL where the link should send the user
- **title** appears as a tooltip when you mouse over the link. It is read by screen readers
- **target="_blank"** opens link in a new tab

URL-SCUSE ME?

URL stands for “Uniform Resource Locator”

UNIFORM

because it is a global standard

RESOURCE LOCATOR

because that’s what an URL does – it locates a resource that lives on the internet

RELATIVE FILE PATHS

Relative paths are URLs that go to a resource in relation to the page you're on

- Resources “local” to you should all be relative paths (your images, HTML documents, fonts, CSS, and JS files)

```
<a href="other-page.html">Link to another page on my website</a>
```

```

```

 (image is in same folder)

```

```

 (image is in parent folder)

ABSOLUTE FILE PATHS

Absolute paths are URLs that start with **http**

```
<a href="http://google.com">Ubiquitous  
search engine</a>
```

- These documents are **not hosted by you**, so if someone renames or deletes the file, your link will be broken



PRACTICE TIME!

ASSIGNMENT

Create a website that about something that interests you

- At least two pages that are linked to each other
- Include a link to an outside website. Bonus: have the link open in a new tab
- Use three heading tags and at least one paragraph
- Use at least one list
- Show at least two images – one local and one remote
- Add one HTML comment
- Validate your website



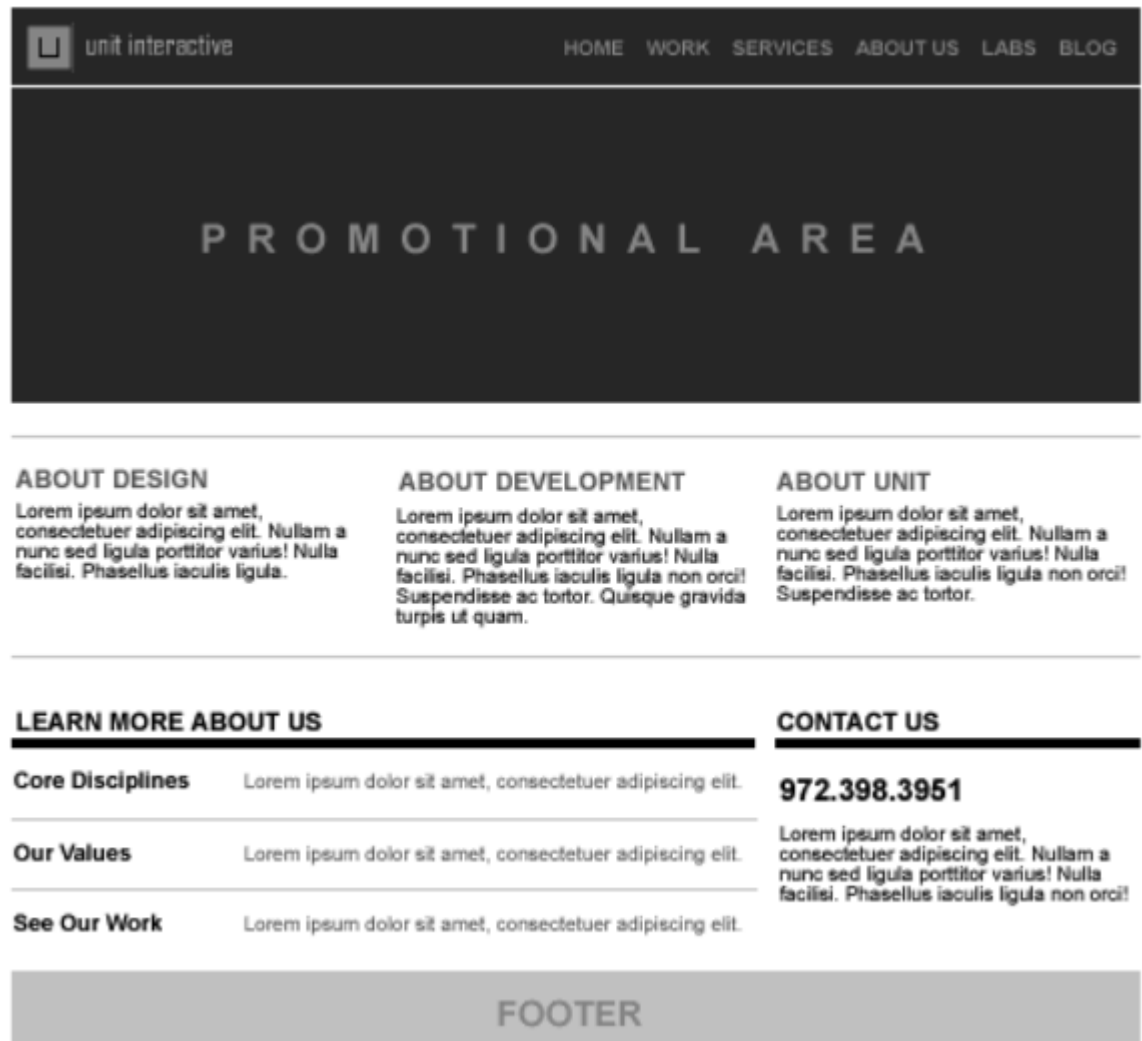
THE ROLE OF THE DEVELOPER

LIFECYCLE OF A WEBSITE

Step 1

User Interface Designers (UX) create wireframes based on research and conversations with the client

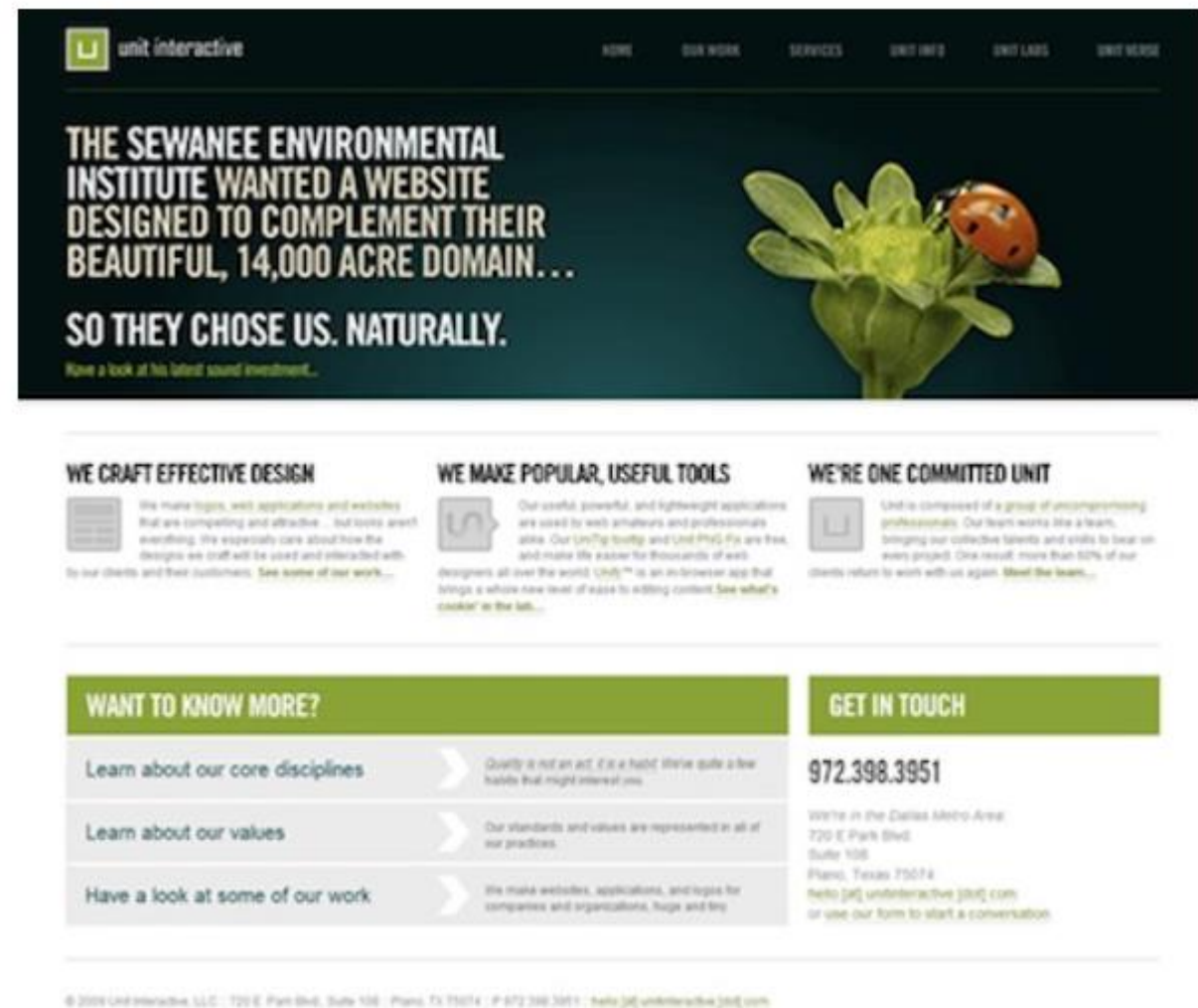
- Wireframes show layout and content



LIFECYCLE OF A WEBSITE

Step 2

Visual Designers
turn wireframes
into comprehensive
layouts, or “comps”

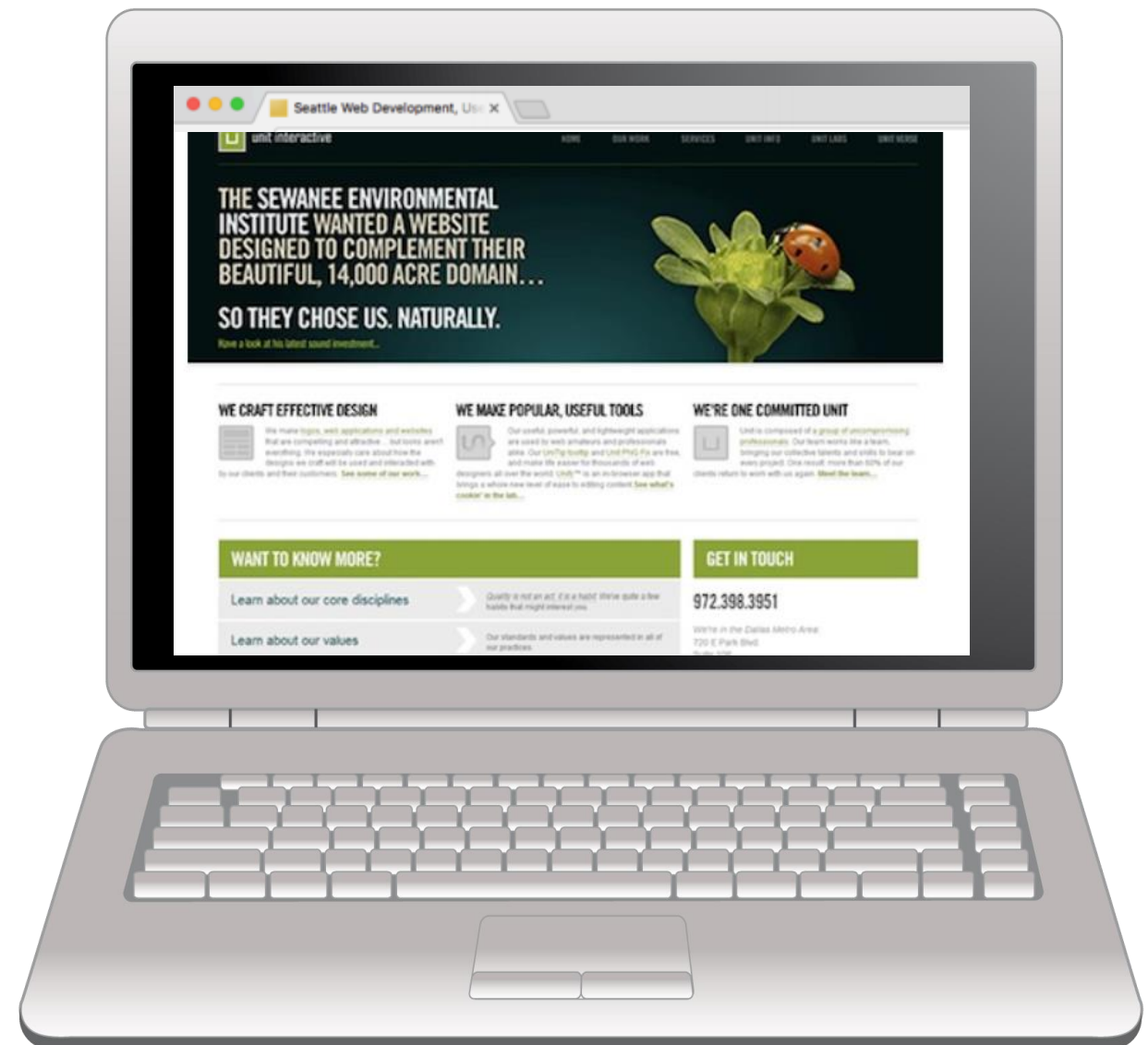


LIFECYCLE OF A WEBSITE

Step 3

Developers turn comps into HTML and style with CSS

- Add interactivity with Javascript



THE ROLE OF THE DEV

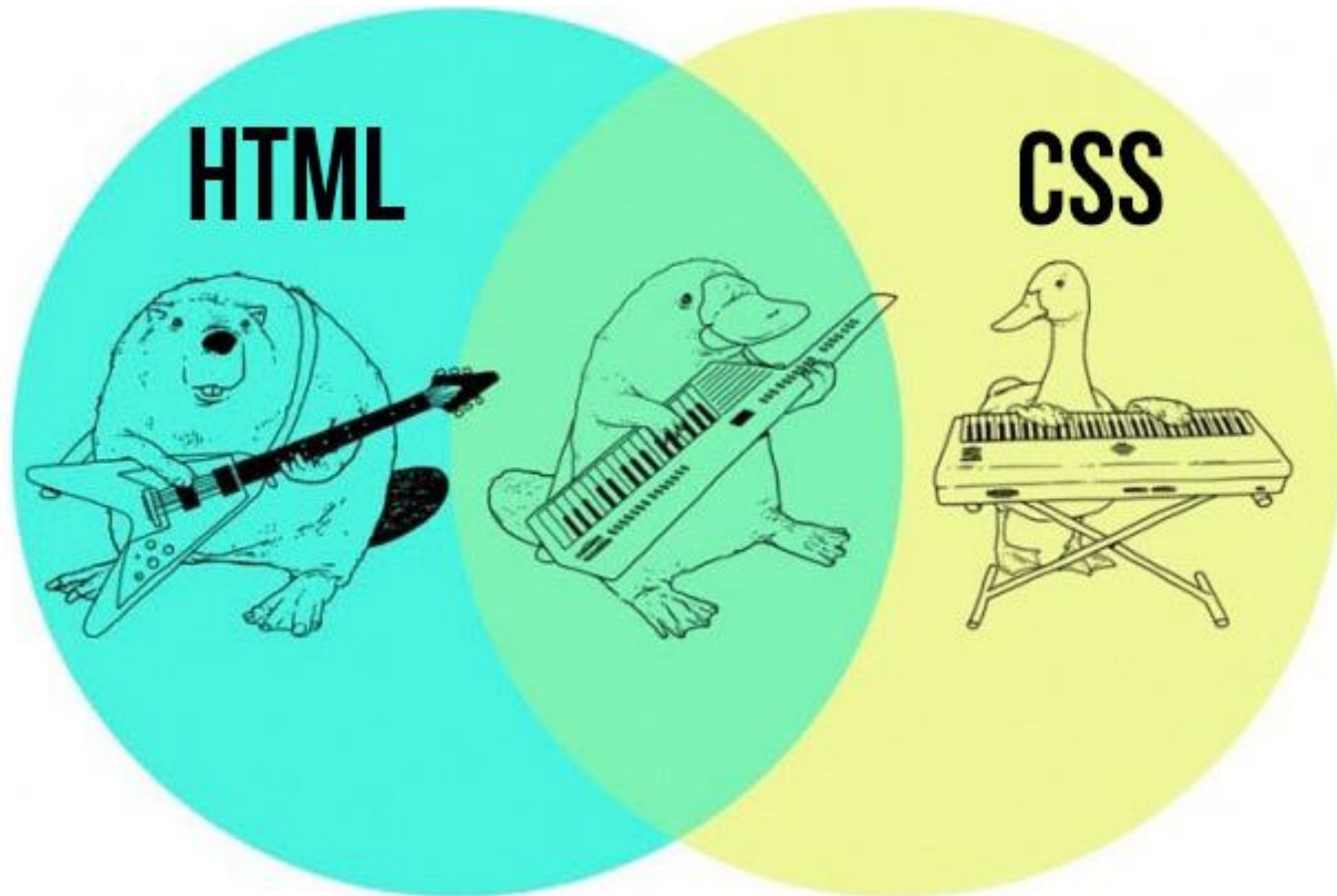
What do I do besides code?

- Problem solve
- Innovate with new technologies and frameworks
- Bring designs to life with animation and motion
- Optimizing for fast loading

CSS



HTML + CSS = WEBPAGE



CASCADING STYLE SHEETS

- CSS is a language for specifying how documents are presented to users
- Allows us to override the browser's default presentation styles with a custom version
- Provides consistent and scalable ways to style single elements, single pages, or entire websites
- Separates look and feel from content/markup (HTML)

CSS: FAIR WARNING

- There is **A LOT** you can do with CSS
- We won't get anywhere close to covering everything!
- We will cover CSS for text styles, colors, positioning, layout, and a couple of extras

WHY USE CSS?

- Helps you avoid duplication by keeping styles in one place (one external stylesheet for many HTML pages)
- Makes style maintenance easier - for example, update the font for the entire site in one line of code!
- Separating presentation from content enforces style consistency

CSS GOES WHERE?

CSS is a different type of language than HTML, and has its own syntax

- CSS can go directly in your HTML file, inside a `<style></style>` element
- You can also create a .css file that can be linked to your HTML page
 - Styles inside a .css file don't need a `<style></style>` tag because the whole file is assumed to be in the CSS language

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

- `selector` is the **thing** you want to style
- `property` is the **attribute** you want to style
- `value` is how you want to style it
- Values always end in semicolons (;)

So!

```
<style>  
    p { color: blue; }  
</style>
```

"All paragraphs will have blue text "

EXAMPLE CSS RULE

```
p { color: blue; }
```

- **selector** is `p` (all `<p>` tags in the HTML)
- **property** is `color`
- **value** is `blue` (many color names are supported, or use the hex code `#0000ff`)

EXAMPLE CSS RULE

```
p {  
  color: blue;  
  font-size: 14px;  
}
```

Multiple **properties** can be defined for a single **selector**, each separated by a semicolon (;)

{ } COMMON FONT PROPERTIES

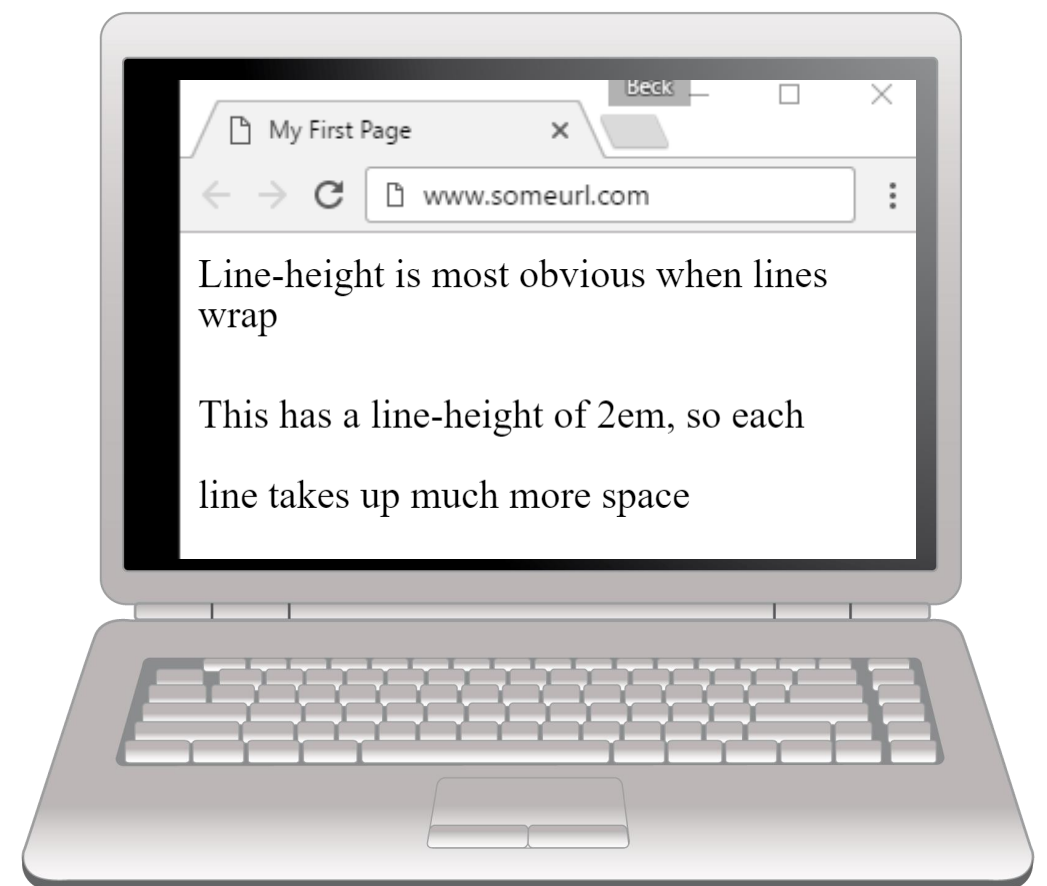
line-height: a number followed by a measurement of the height of a line of that element, in ems (em) or pixels (px)

- similar to **leading** in typography

```
p { line-height: 1.4em; }
```

font-size: a number followed by a measurement of the height of that element's text in ems (em) or pixels (px)

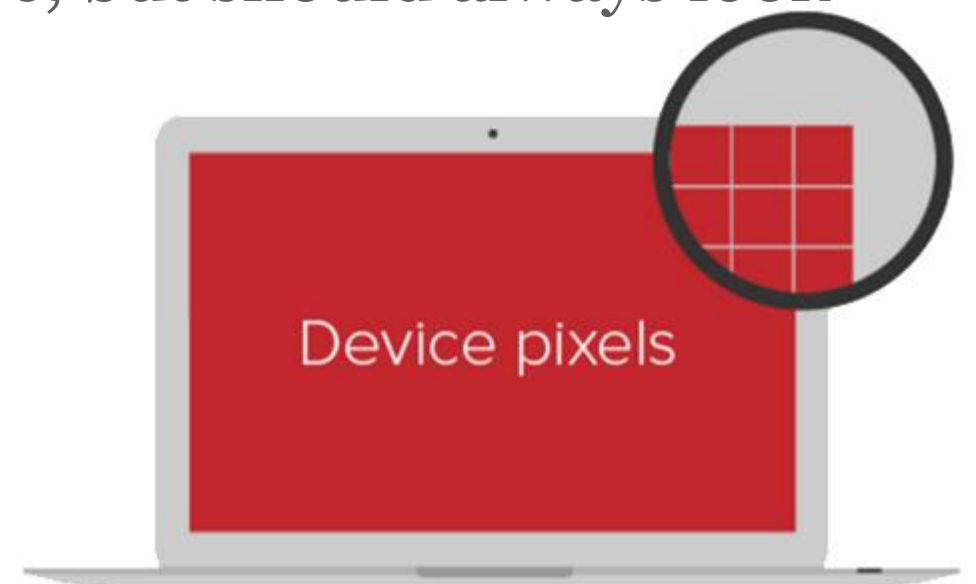
```
p { font-size: 14px; }
```



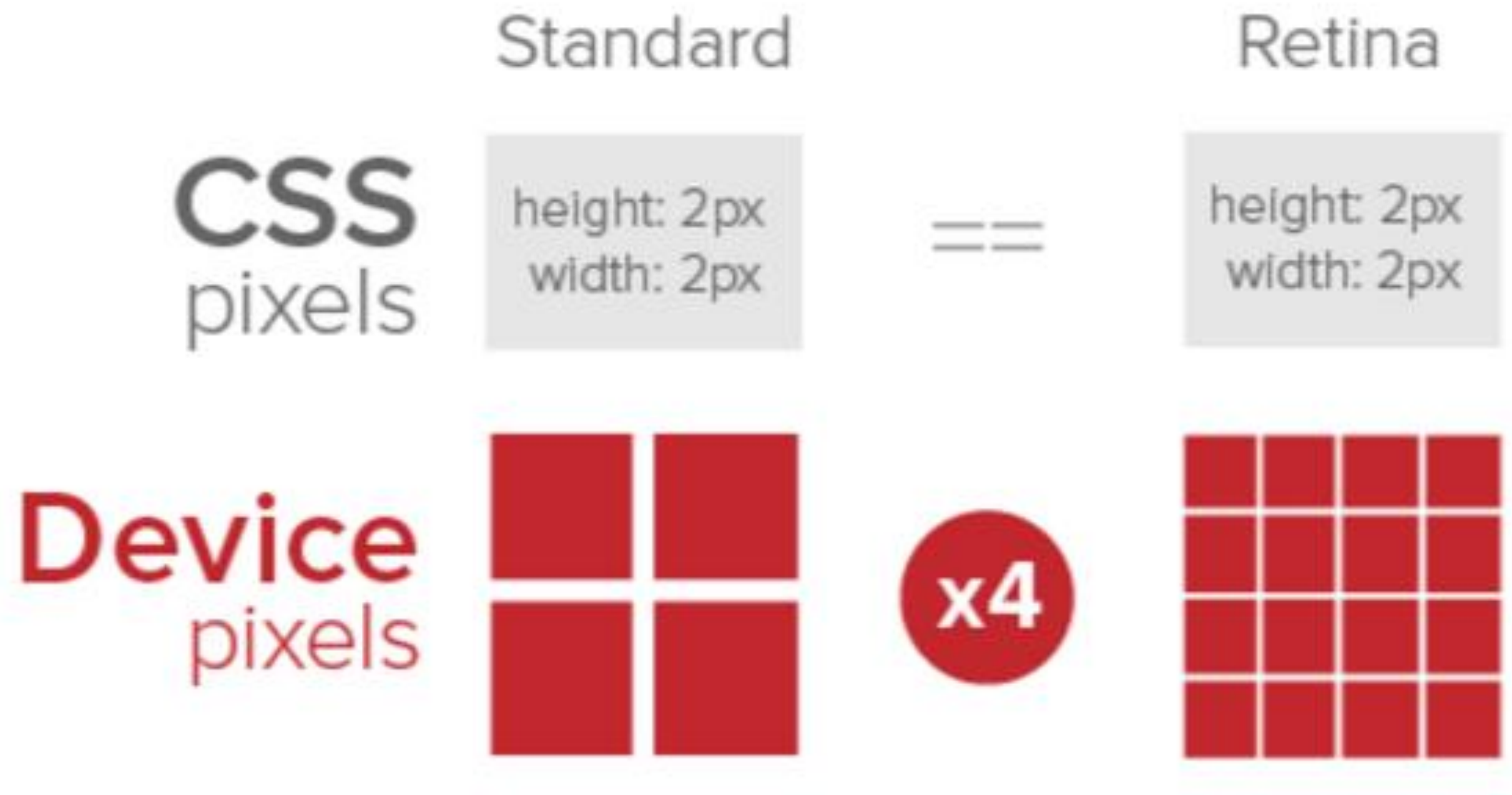
{ } QUICK ASIDE ABOUT UNITS

The two standard units for sizing in CSS are **px** and **em**

- **px** is an abstract unit that isn't related to font height and isn't a physical unit of measurement
 - Devices with more PPI (pixels per inch) may use several "device" pixels when displaying a 1px line
 - That means that px size varies by device, but should always look "about the same"



{ } QUICK ASIDE ABOUT UNITS



{ } AH-EM

- **em** refers to the height of the letter 'm' of the font being used
- This unit of measurement is a description of the **relative** size between this element and its parent
- So `h2 { font-size: 2em; }` means the header is 2 times as big as the letter 'm' of the default font in your html document

{ } THAT WASN'T QUICK

Because em is **relative**, that means that if the parent's font size is increased, the children will get bigger too.

	<code>body { font-size: 100%; }</code>	<code>body { font-size: 120%; }</code>
<code>font-size: 1em</code>	The quick brown fox	The quick brown
<code>font-size: 12px</code>	The quick brown fox	The quick brown fox

{ } COMMON FONT PROPERTIES

font-style: `normal` by default – can also be `italic` or `oblique`

font-weight: `normal` by default – can also be `bold`, or values of `100`, `200`, etc. (depending on the typeface)

font-family: the name of a typeface installed on the user's computer

```
p {  
    font-family: Arial, Helvetica, sans-serif;  
}
```

- The W3 has a list of [“web safe” fonts](#) that most people will have installed locally

{ } FONT TRANSFORM

text-transform: changes font casing. Can be **uppercase** (all caps), **lowercase**, or **capitalize** (first letter of all words capitalized)

letter-spacing: change font kerning by specifying the space between letters in ems (em) or pixels (px)

```
p {  
  font-family: Arial;  
  text-transform: uppercase;  
  letter-spacing: 2.4px;  
}
```

LOOKS LIKE THIS

{ } COLORS

- **color:** changes the color of **text**
- **background-color:** sets the **background** color of an element
- Color **value** can be set using **names**, **HEX**, **RGB**, or **RGBA**
 - Name: **white**
 - Hex: **#ffffff**
 - RGB: **rgb(255, 255, 255)**
 - RGBA: **rgba(255, 255, 255, 0.8)**

{ } COLOR EXAMPLES

```
p {  
    color: black;  
}
```

```
p {  
    background-color: #000000;  
}
```

```
p {  
    background-color: rgb(0, 0, 0);  
}
```

{ } TEXT-ALIGN

You can change the alignment of text using the **text-align** property.

Values:

- center
- left
- right
- justify

```
h1 { text-align: center; }
```



{ } CSS COMMENTS

Just like HTML, CSS can have **comments**

- Start with `/*`
- End with `*/`

```
<style>
```

```
    /* I am a CSS comment! */
```

```
</style>
```



PRACTICE TIME!

PRACTICE

Add a `<style></style>` section in the `<head>` on your page

Make some style changes using CSS

- Consider changing font color, font family, font size, text alignment, and background color

“HOMEWORK”

- Practice!
- Next time you see a cool website, inspect how they did it
- If you have questions during the week, feel free to email me at beckjohnson@gmail.com
- Optional: read chapters 6-7 of *HTML and CSS: Design and Build Websites*

