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, email marketing, etc?
- What are you hoping to get out of this class?

CLASS SCHEDULE

- Six sessions over six weeks
- Wednesdays from Feb 13 to March 13, at 6:30-9:30 pm
- 10 minute break somewhere in the middle
- No grades, no tests
- Questions and feedback highly encouraged!

COURSE OVERVIEW

- Basics of HTML and CSS
- Using CSS to style web pages
- Website structure, navigation, and file organization
- Preparing images for use on the web
- Overview of related technologies (Javascript, Git)

TODAY

- Code editors
- Basic HTML
- Basic CSS – font styling, colors, alignment
- Build your first web page!

ODDS & ENDS

beckjohnson.com

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

OVERVIEW OF A WEBSITE

DEFINITIONS

A **website** is a bunch of webpages connected to one another with links.

A **webpage** is a text file with the extension .html

- "Marked up" with HTML tags
- Styled using CSS

To get your website online, you upload it to a **server**, which is a computer that "serves" webpages when they're asked for.

CONTENT, DESIGN, & CODE

CONTENT

What the site “is”

HTML

DESIGN

How the site looks

CSS

CODE

How the site responds

JAVASCRIPT

HTML EDITORS

HTML is just text

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

You can open and edit **any** HTML file in **any** text editor

HTML EDITORS

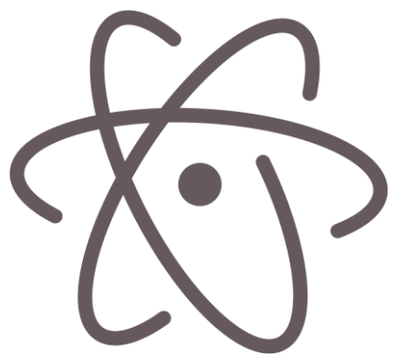
However, specialized code editors make writing code easier by providing

- Syntax highlighting
- Autocomplete
- Auto formatting

HTML EDITORS

VS Code

Brackets



Atom

Sublime Text

Coda

WEB BROWSERS

A web browser reads an .html file and turns “mark up” language into formatted text

All browsers have a mode called “**Developer Tools**” that show you the HTML, CSS, and Javascript used to create that page

DEVELOPER TOOLS

Chrome/Firefox

- Right-click anywhere on the page > Inspect
- OR hit the **F12** key

Safari

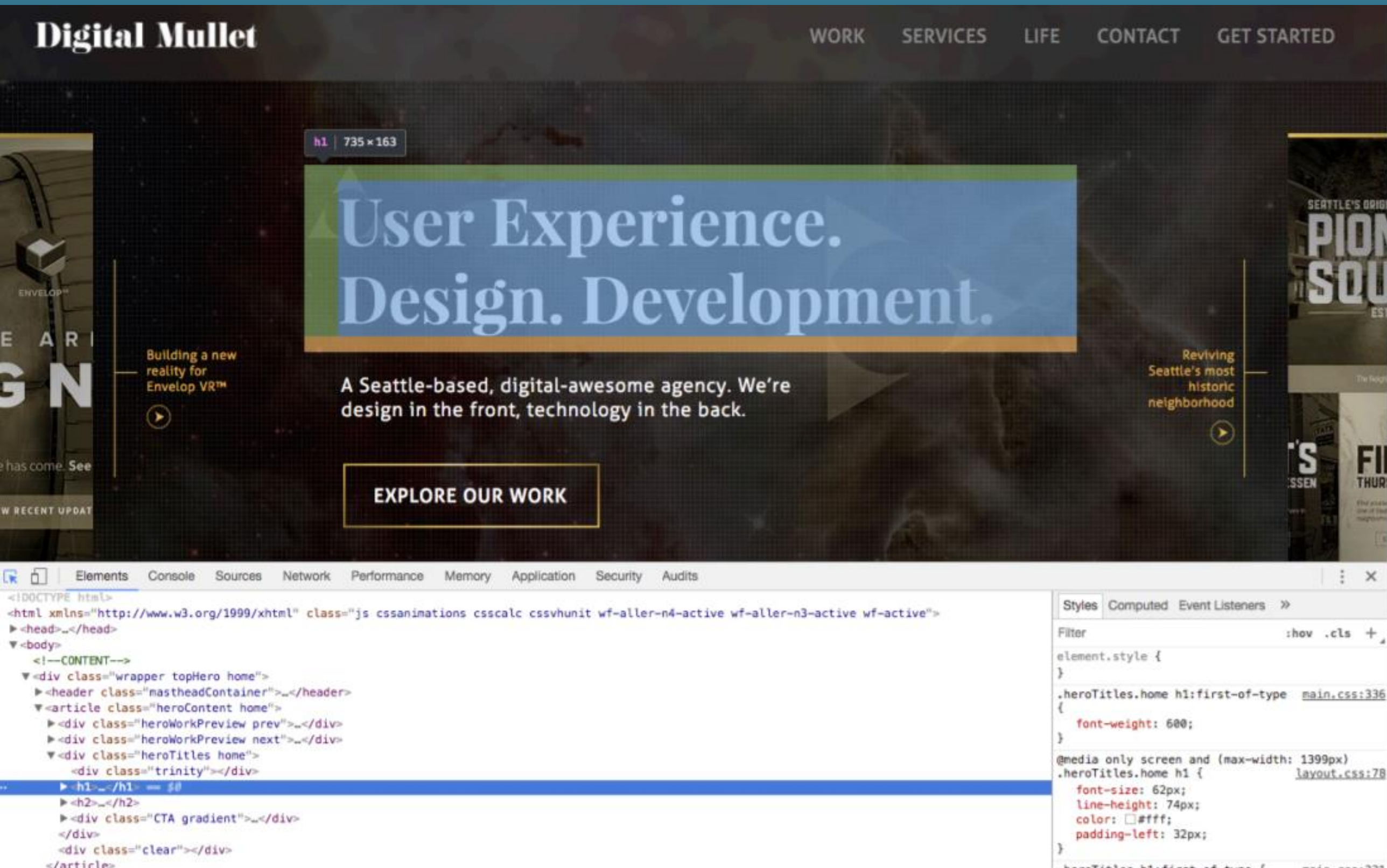
- Open Preferences > Advanced > Show Develop menu
- Right-click > Inspect Element

Internet Explorer

- F12 key

LET’S TRY IT

DEVELOPER TOOLS



WEB BROWSERS




You can experiment directly in the browser using dev tools before making permanent changes

- You can modify both HTML and CSS
- Any changes disappear when you refresh the page – copy to a local file if you want to keep them!

<html>

HTML DOCUMENTS

HTML IS FOR CONTENT

A browser doesn't know what the content on your page actually is, unless you tell it

- HTML tags describe your content to a browser so that it can be displayed properly

- It provides a way for CSS to be applied, to give better control over how the webpage is presented

HTML DOCUMENT

```
<!doctype html>
<html>
<head>
    <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>
```

BLOG, LEARN TECH SKILLS

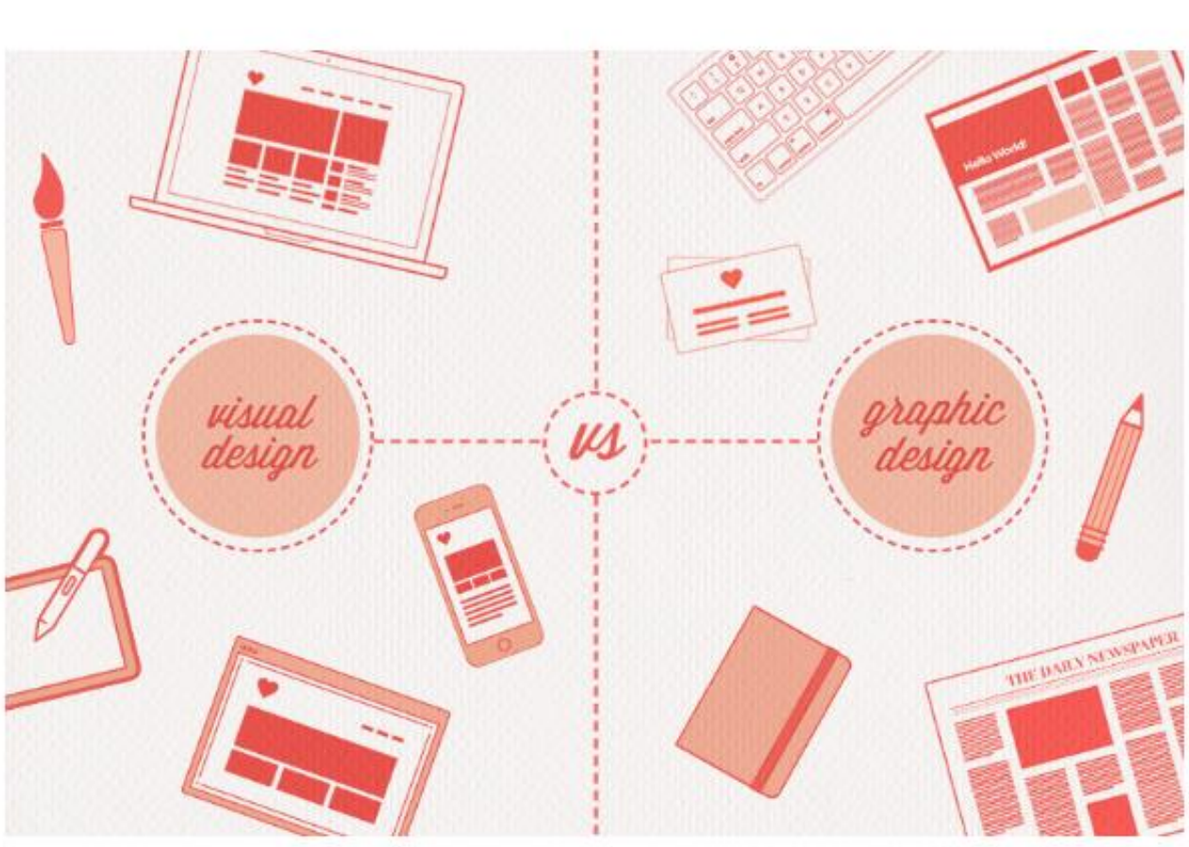
Visual Design vs. Graphic Design: What's the Difference?

Cameron Chapman

Last updated
January 25th, 2018

51 Comments

SHARE THIS



Graphic designer. Visual designer. User interface (UI) designer. User experience (UX) designer. Web designer.

header

THE HARD REFRESH by skillcrush

SKILLCRUSH COURSES FREE 10-DAY BOOTCAMP LOG IN

nav

TECH 101 CAREER LIFE CULTURE PODCAST FREE RESOURCES

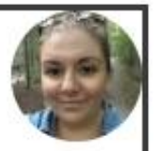
BLOG, LEARN TECH SKILLS

h1

Visual Design vs. Graphic Design: What's the Difference?

section

aside

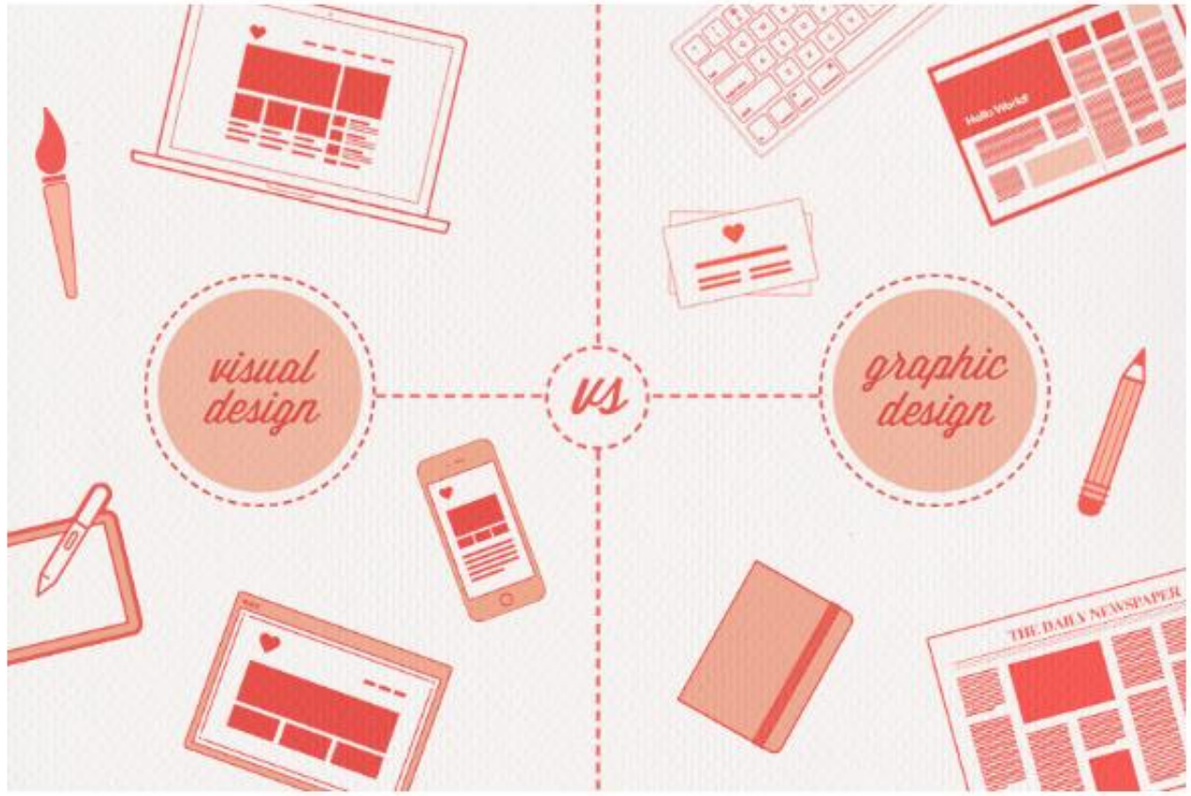


Cameron Chapman

Last updated January 25th, 2018

51 Comments

SHARE THIS



p

Graphic designer. Visual designer. User interface (UI) designer. User experience (UX) designer. Web designer.

Visual Design vs. Graphic Design: What's the Difference?

Cameron Chapman Last updated January 25th, 2018 52 Comments

Graphic designer. Visual designer. User interface (UI) designer. User experience (UX) designer. Web designer.

How many different designer job titles *are* there? And are they all just fancy names for the same thing?

```
<!doctype html>
<html>
<head>
   <title>The Difference Between Visual Design and Graphic Design</title>
</head>
<body>
   <header>
      <img src="logo.jpg" alt="The Hard Refresh Logo" />
      <nav>
            <ul>
                <li><a href="/courses.html">Skillcrush Courses</a></li>
                <li><a href="/bootcamp.html">Free 10-Day Bootcamp</a></li>
            </ul>
      </nav>
      <nav>
             <ul>
                <li>Tech 101
                        <ul>
                           <li><a href="/html.html">HTML & CSS</a></li>
                           <li><a href="/design.html">Design</a></li>
                        </ul>
                </li>
                <li><a href="/career.html">Career</a></li>
             </ul>
         </nav>
   </header>
   <section>
        <h1>Visual Design vs. Graphic Design: What’s the Difference?</h1>
        <aside>
                <img src="Cameron-chapman.jpg" alt="Headshot of Cameron Chapman" />Cameron Chapman
        </aside>
        <img src="visual-vs-graphic.jpg" alt="Visual vs graphic design" />
        <p> Graphic designer. <a href="/design.html">Visual designer</a>. User interface (UI) designer.
            User experience (UX) designer. <a href="/design.html">Web designer</a>.</p>
   </section>

</body>
</html>
```

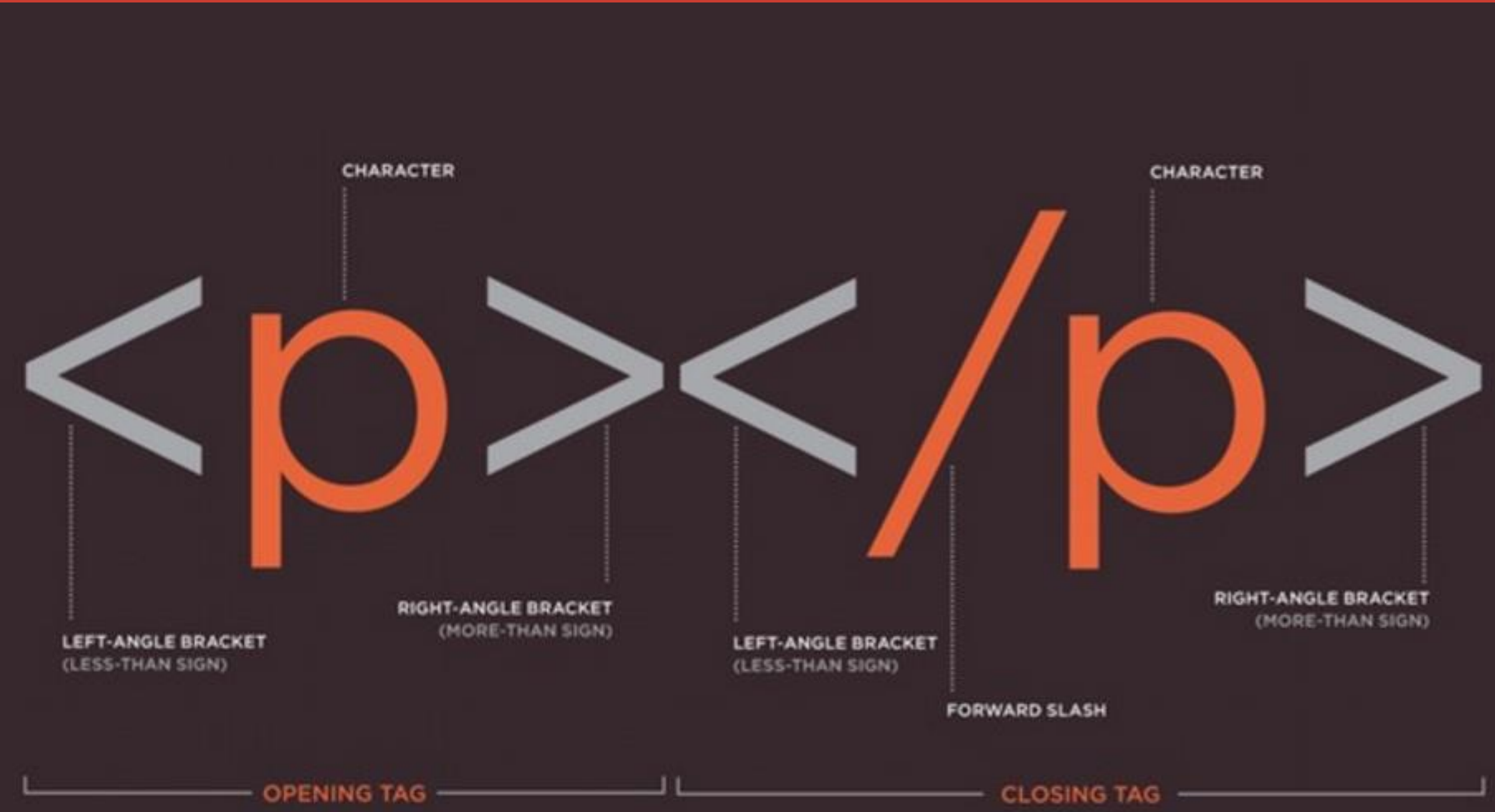
HTML ELEMENTS

- HTML tags (sometimes called “elements”) are contained in <> brackets

- HTML tags wrap around the content they are describing, with an **opening** tag and a **closing** tag

```
<tag>Content goes in here</tag>
```

HTML ELEMENTS



HTML RULES

Tags are written in lowercase

<a> not <A>

Tags must be closed

```
<p>This is a paragraph</p>

<h1>This is a heading</h1>
```

A few are “self closing” because the content they describe are part of the tag itself

```
<img/>
```

HTML ATTRIBUTES

Some tags have `attributes` that provide more information or meaning

Attributes have a `name` and a `value`, joined with a `=` sign

- The `value` is wrapped in quotes

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

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)
- If you don’t specify what version to use, the browser will guess…

DOCTYPE

These old doctypes are not commonly in use anymore:

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

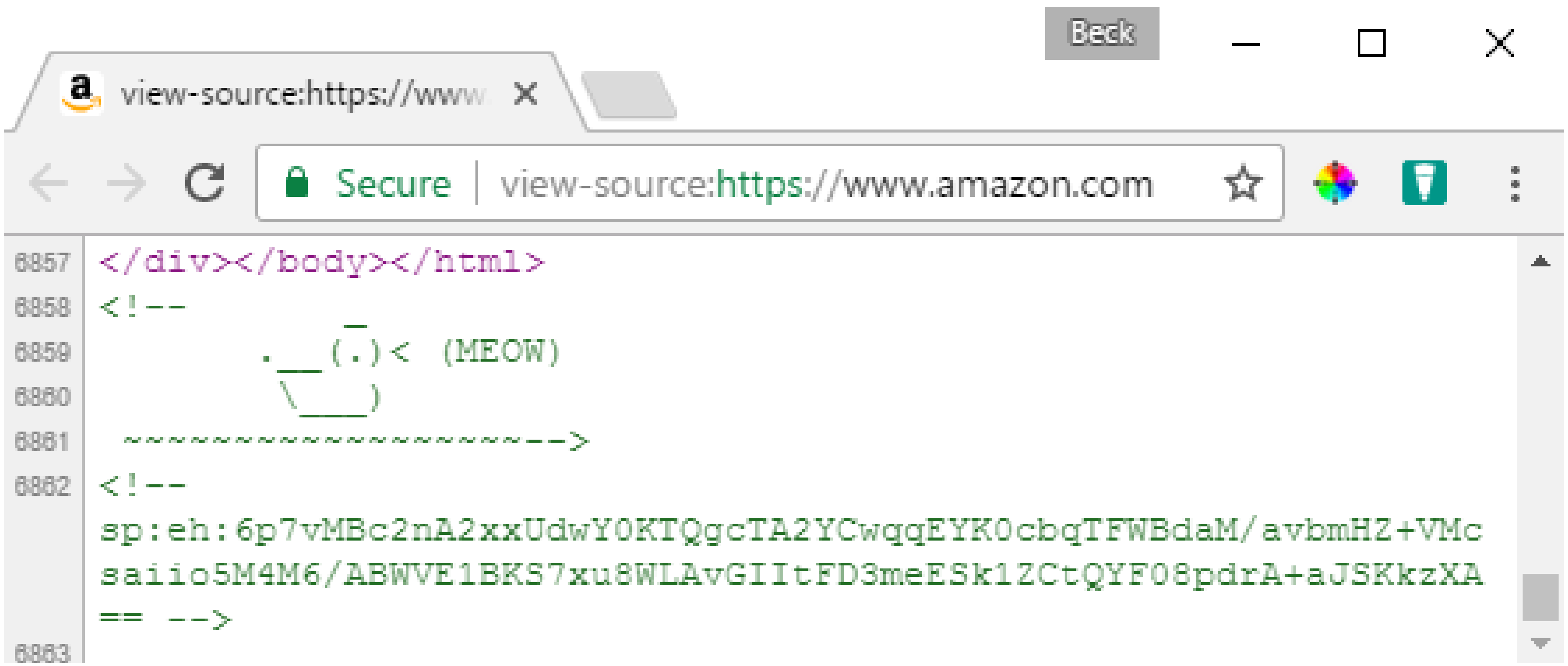
<!--HTML COMMENTS-->

```
<!-- 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” (right-click in a browser -> View Source)

<!--HTML COMMENTS-->

Sometimes they don't really have a point…



HEAD ELEMENT

```
<head></head>
```

Holds information about the document that is (mostly) not visible to the user

- Any content other than `title` is **not shown on the page**!

- Can contain CSS and Javascript

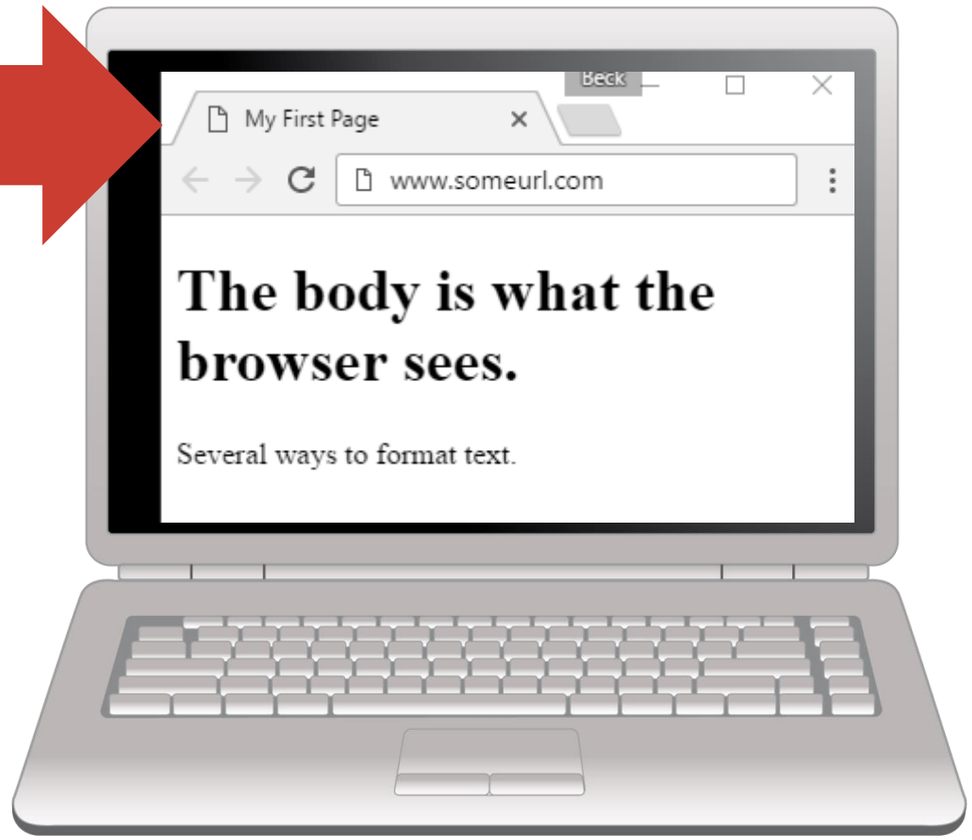
- Don't mix this up with "headings" such as `<h1>`

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

TITLE TAG

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

- Displays in the browser tab



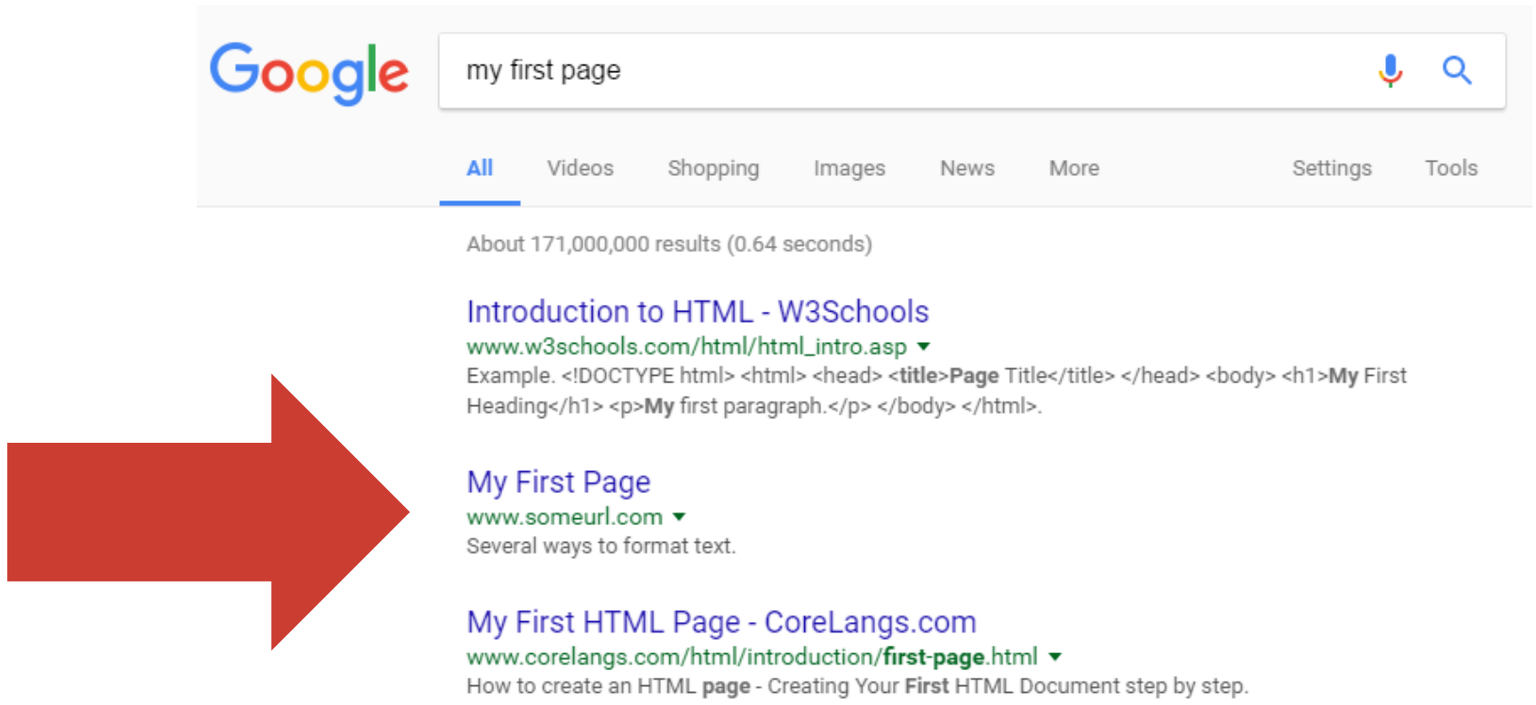
- **Required** inside <head>

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

TITLE TAG

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

- Also the name of the page when page is bookmarked
- The title for the page in search results on Google



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, text, etc.

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

    <!-- all my sweet content -->

</body>
```

HEADINGS

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

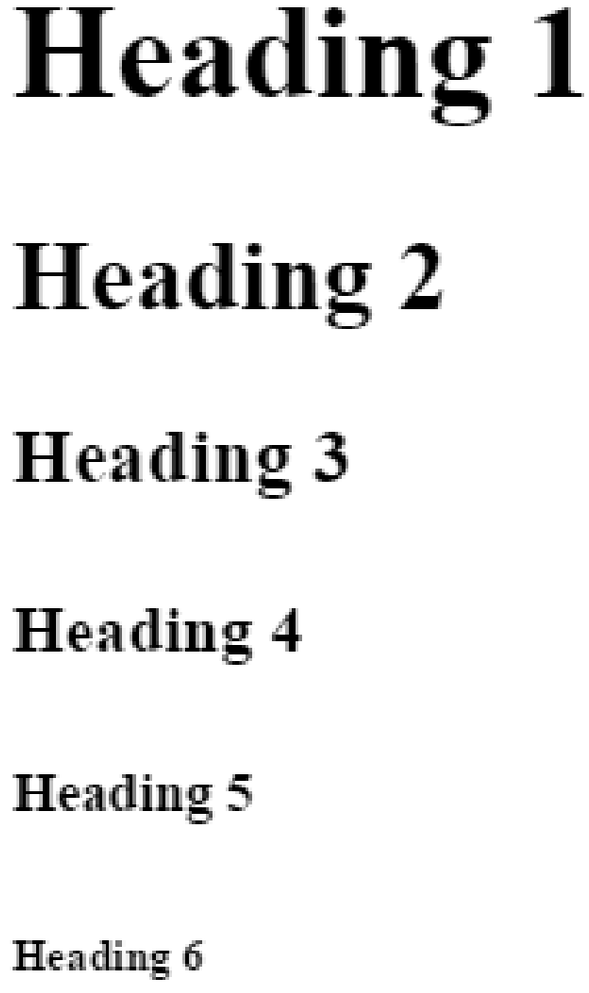
Headings range from most important to least important

`<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>
```



PARAGRAPHS

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

Browsers automatically add a bottom margin to <p> elements when they display the webpage (although this can be changed with CSS)

HTML ignores white space, so this displays the same:

```
<p>Hi!

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

FORMATTING

 indicates *emphasis*

- By default, this displays as italic

 indicates **importance**

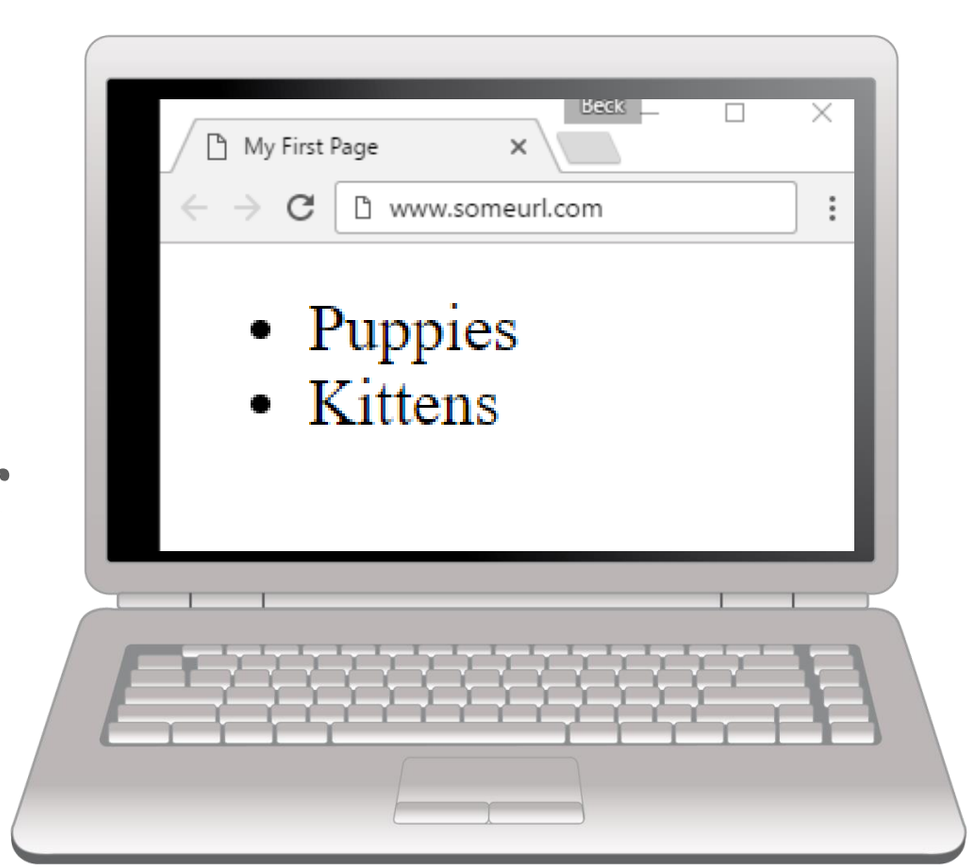
- By default, this displays as bold

```
<em>That's</em> not a knife.
<strong>This</strong> is a
knife.
```

LIST ELEMENTS

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

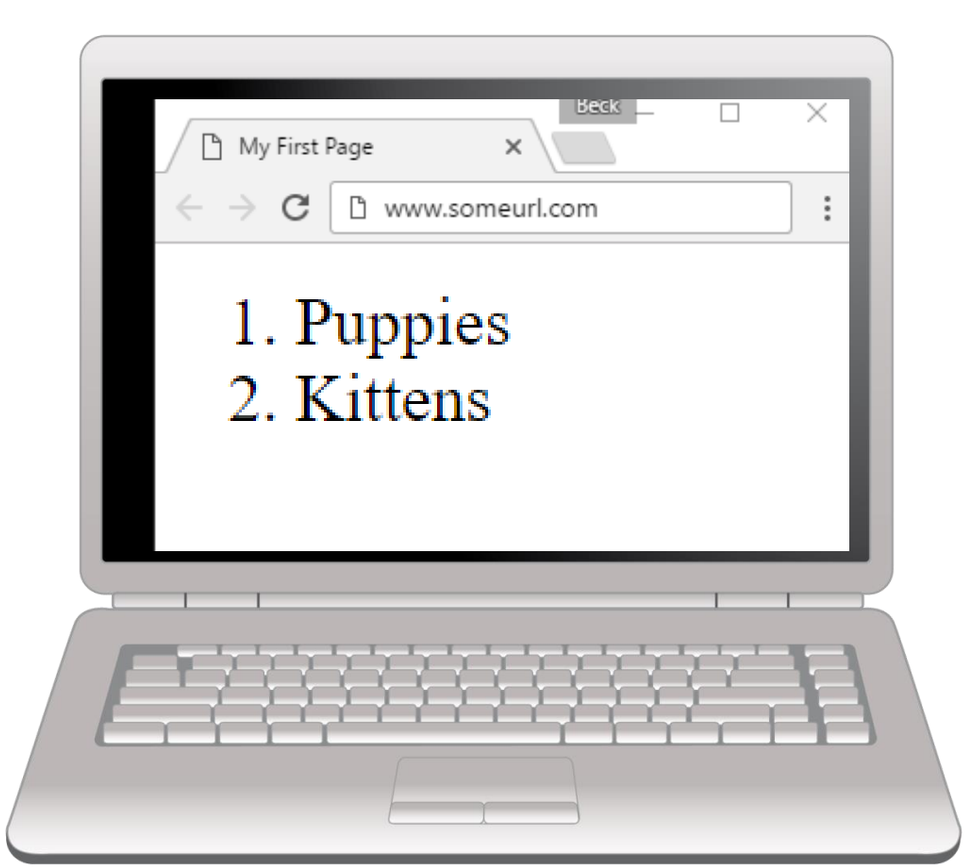
Unordered lists `<ul>` appear in the browser with **bullets**



LIST ELEMENTS

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

Ordered lists `<ol>` appear in the browser with **numbers**



LIST ELEMENTS

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

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

IMAGES

```
<img src="kitten.jpg" alt="Cute kitten" />
```

Images are “self-closing" (meaning they end with `/>`) and have two required **attributes**:

- `src` is a **path** to where the file lives (local or external)

- `alt` is a **description** of the image (used for screen readers, search engines, etc)

IMAGES

```
<img src="kitten.jpg" alt="Cute kitten"
height="200" title="Ollie" />
```

- `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 tag” or link

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

- The browser will automatically underline links and turn the text inside blue

SOME <A>TTRIBUTES

```
<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's read aloud by screen readers
- `target="_blank"` forces the link to open 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>
```

```
<img src="image.gif" />
```
(image is in same folder)

```
<img src="../images/image.gif" />
```
(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>
```

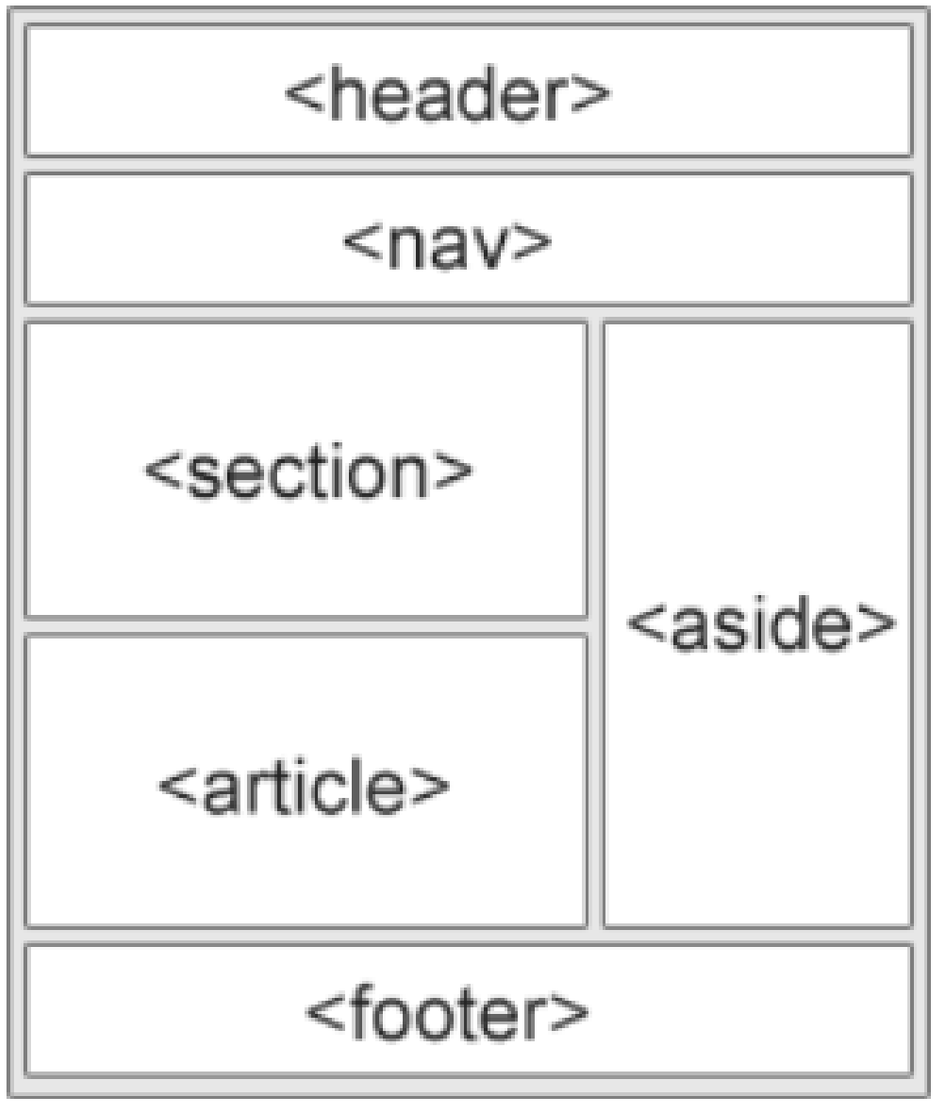
- These documents are **not hosted by you**, so if someone renames or deletes the file, your link will be broken

LAYOUT

`<nav>` indicates that everything inside is related to navigation

`<section>`, `<article>` and `<main>` are used to define content sections

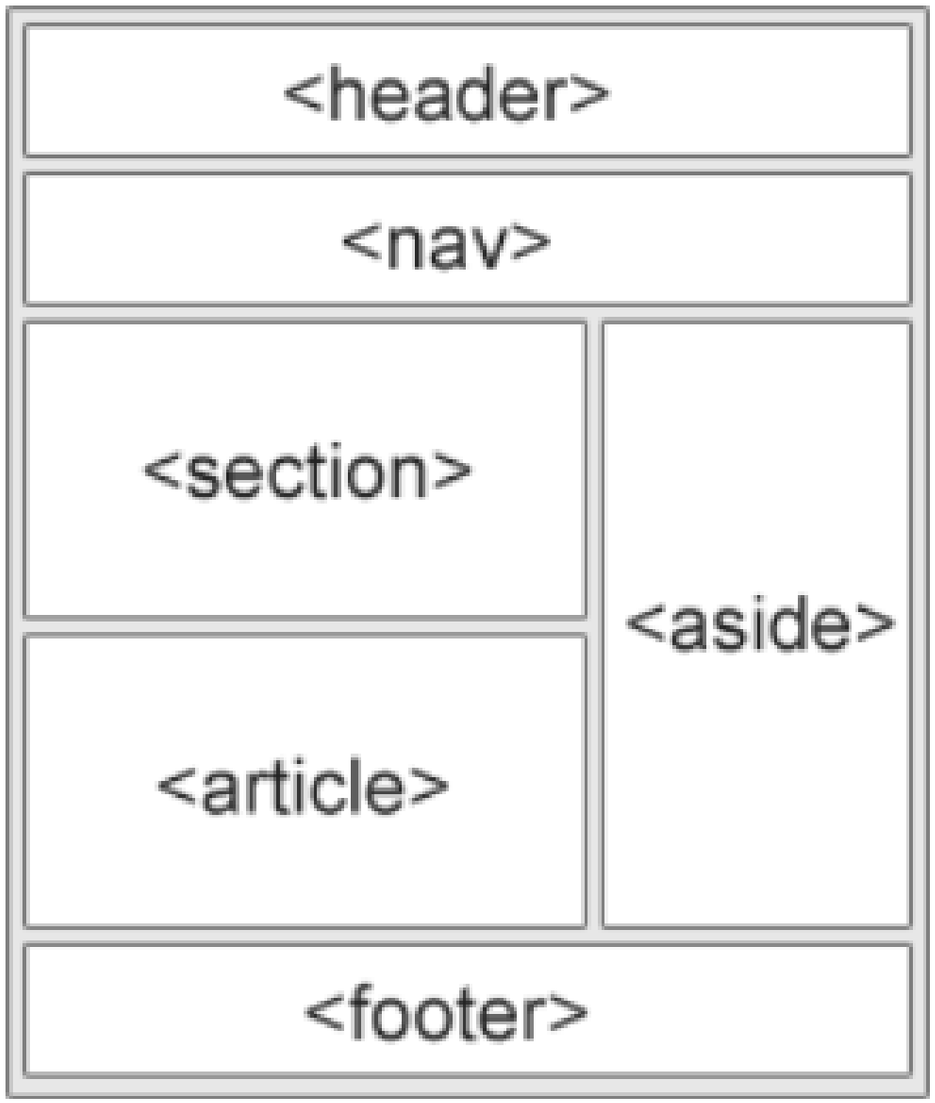
`<footer>` wraps footer content (for example copyright)



LAYOUT

`<nav>`, `<section>`, `<article>`, `<main>` and `<footer>` separate content into logical sections, and they don't visually change the page until you apply CSS

But, they help organize content and allow search engines to understand the structure of your page

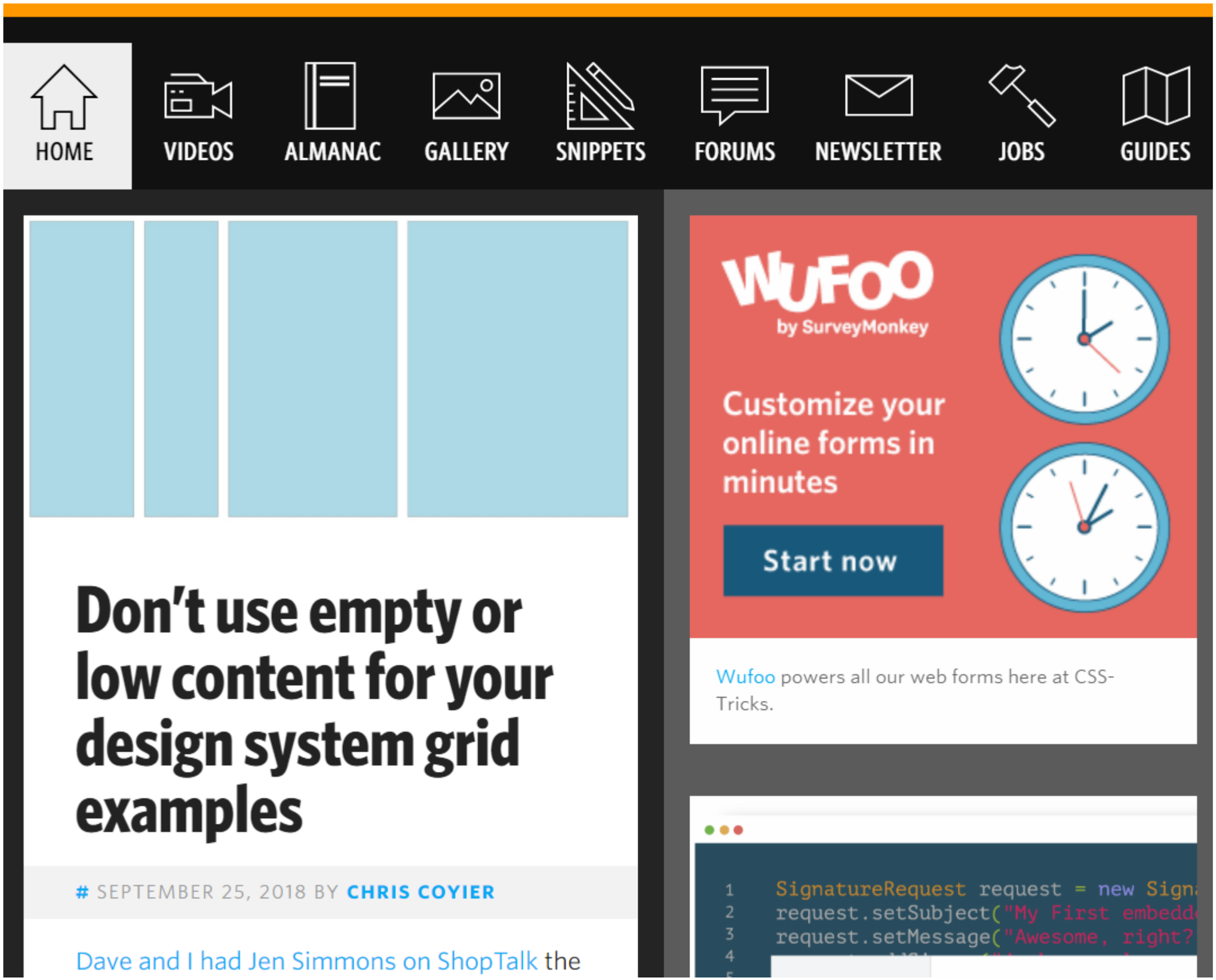


header

nav

aside

main



PRACTICE TIME!

ASSIGNMENT

Create a website that about something that interests you

- Open VS Code and create a new file
  - Name it **index.html**
- Use some heading tags and at least one paragraph
- Use at least one list
- Show at least two images
- Validate your website by copying it into the form here: https://validator.w3.org/#validate_by_input

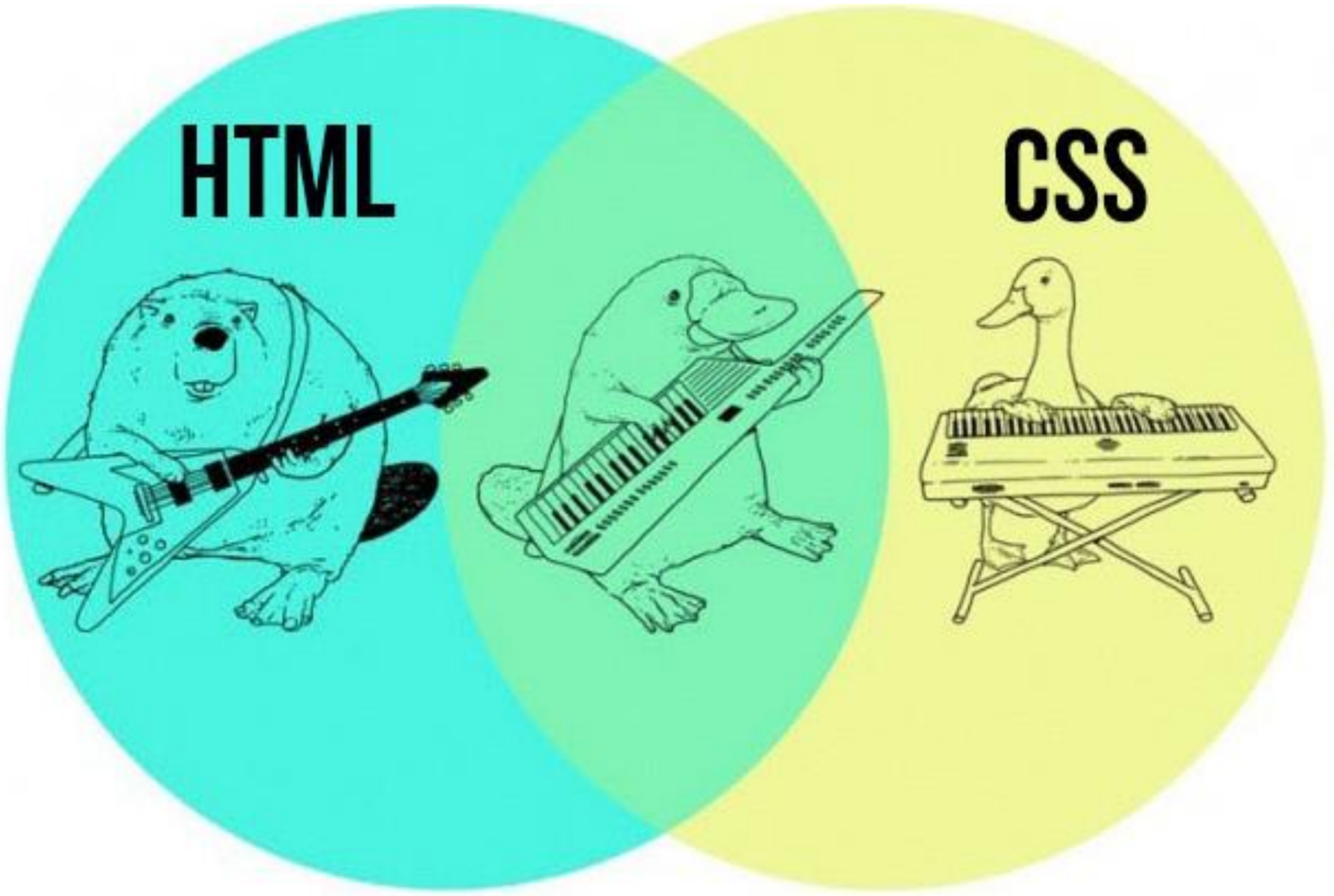
CSS

CSS IS FOR PRESENTATION

After your content has been created and formatted with HTML, use CSS to change how it is presented

- This is similar to changing margins, font color, or font family in MS Word (it doesn't change how things are structured, just how they look)

HTML + CSS = WEBPAGE



CASCADING STYLE SHEETS

- CSS is a language for specifying how a website is visually 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

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

CSS: FAIR WARNING

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

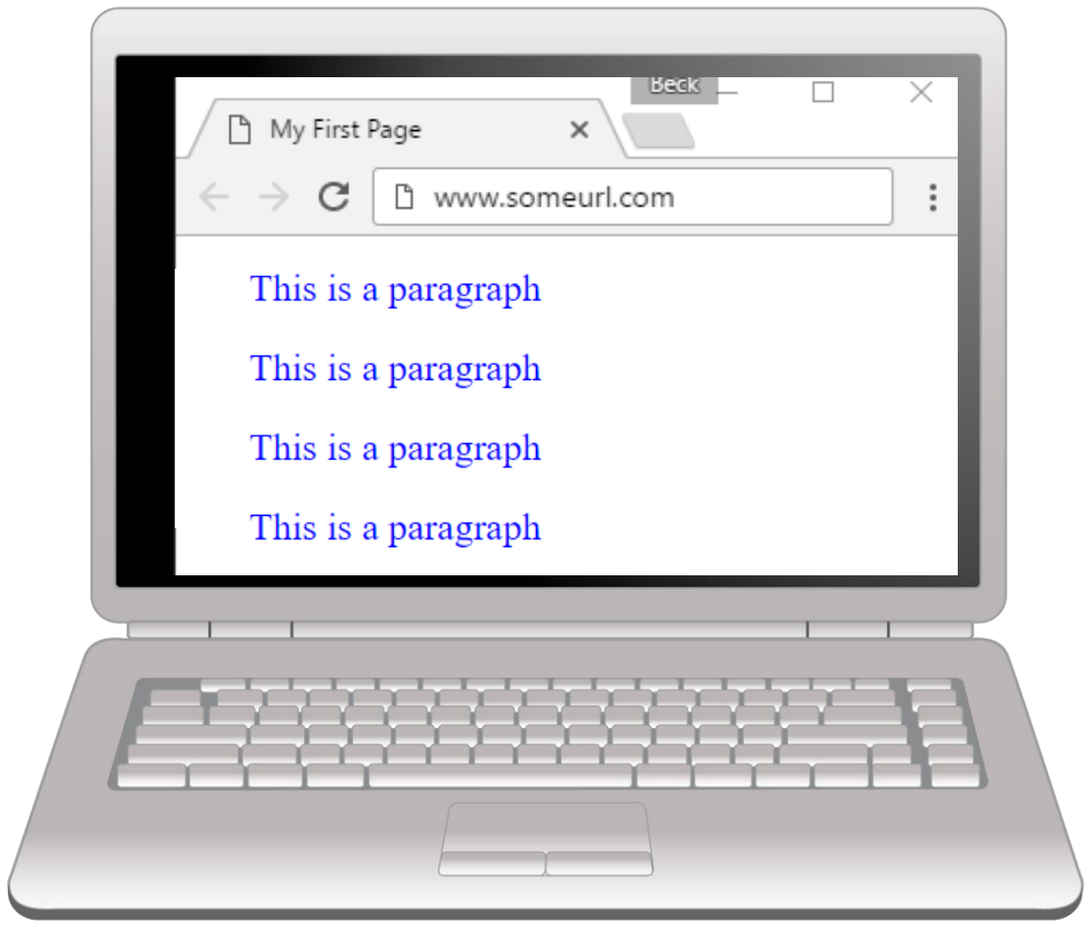
```
selector { property: value; }
```

- `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 (`;`)

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

"All paragraphs will have blue text "

```
<html>
▼<head>
    <style>
        p { color: blue; }
    </style>
  </head>
▼<body>
  ▼<section>
      <p>This is a paragraph</p>
      <p>This is a paragraph</p>
      <p>This is a paragraph</p>
      <p>This is a paragraph</p>
    </section>
  </body>
</html>
```



EXAMPLE CSS RULE

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

- `selector` is `p` (all `<p>` tags in the HTML page)
- `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

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

`100` - Thin
`200` - Extra Light (Ultra Light)
`300` - Light
`400` - Normal
`500` - Medium
`600` - Semi Bold (Demi Bold)
`700` - Bold
`800` - Extra Bold (Ultra Bold)
`900` - Black (Heavy)

{ } COMMON FONT PROPERTIES

Weight mappings for a font family with 400, 700 and 900 weight faces

Weight mappings for a font family with 300 and 600 weight faces

{ } COMMON FONT PROPERTIES

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

```
ul {
    font-style: italic;
}
```

If an italic version of the font is not installed, the browser will artificially slope the normal typeface.

Artificial sloping versus real italics

{ } COMMON FONT PROPERTIES

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

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

Typically the `value` is a list of acceptable fonts that can used

- Goes down the list until a font file is found, or until one of the default options is used instead (`serif`, `sans-serif`, `cursive`, `fantasy`, or `monospace`)

- The W3 has a list of "web safe" fonts that most people will have installed locally

{ } COMMON FONT PROPERTIES

`font-size:` how big the font should be

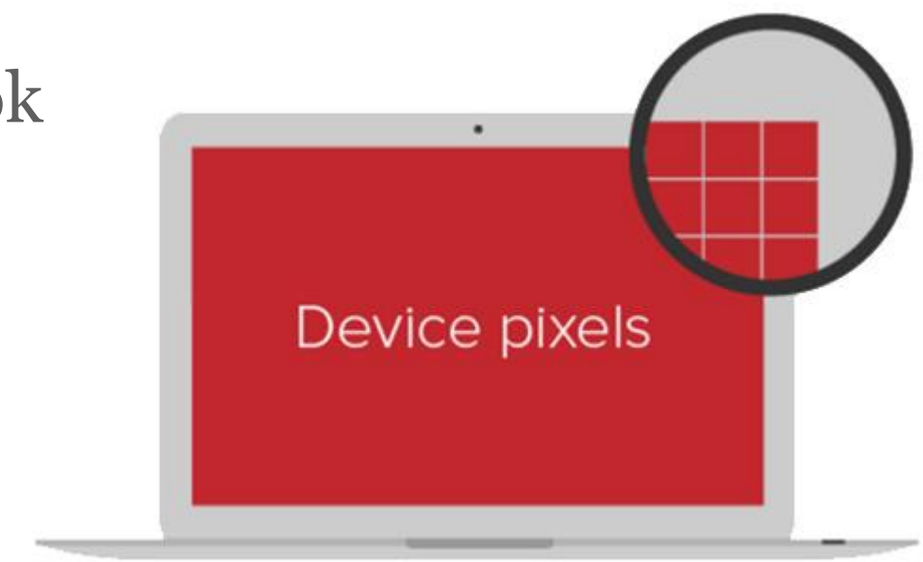
The `value` is a number representing the size of that element's text in ems (em) or pixels (px)

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

{ } QUICK ASIDE ABOUT UNITS

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

- **px** stands for pixels, but it won't be actual device pixels
  - Devices with more PPI (pixels per inch) may use several device pixels when displaying 1px
- 1px should look like ~1/96 inch on a device with a pixel density of 96dpi, held at arm's length
- That means that 1px should always look "about the same" even though it's not technically an absolute size



{ } 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 heading 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.

	`body { font-size: 100%; }`	`body { font-size: 120%; }`
`font-size: 1em`	The quick brown fox	The quick brown
`font-size: 12px`	The quick brown fox	The quick brown fox

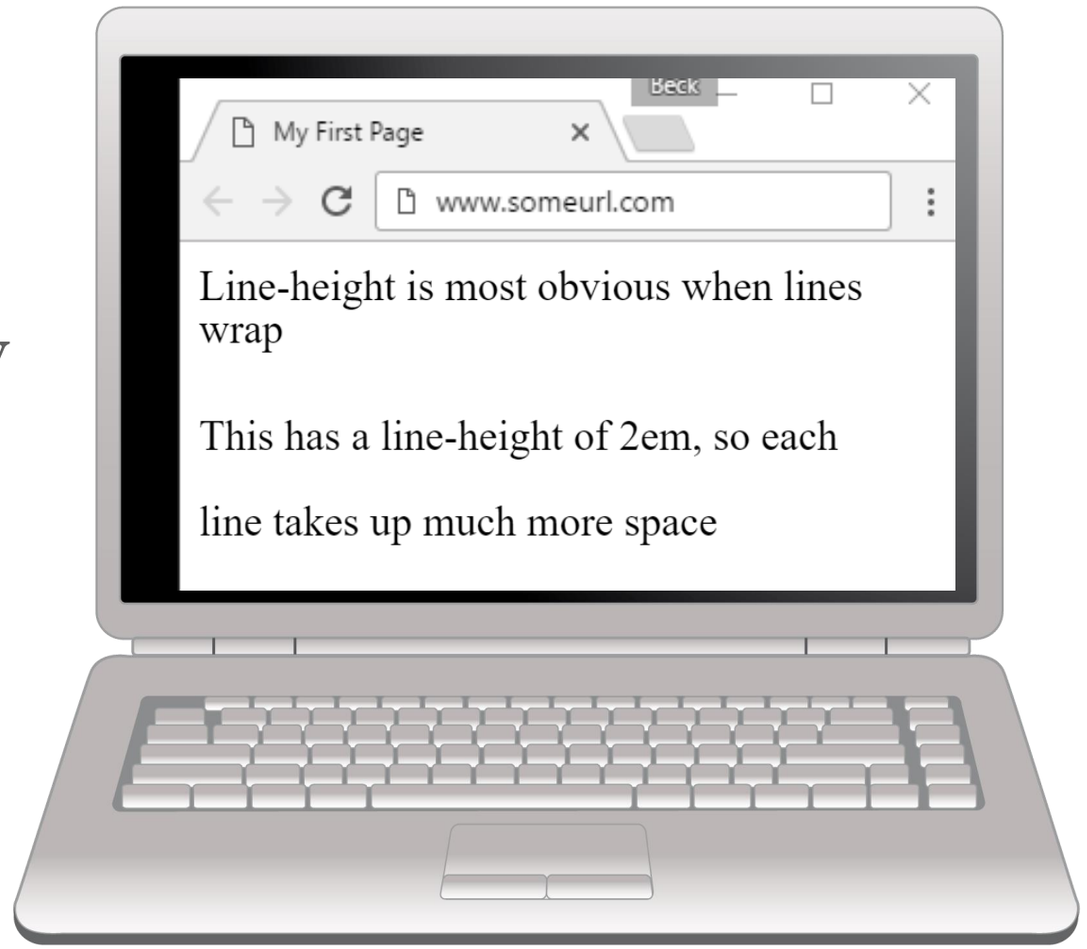
{ } COMMON FONT PROPERTIES

`line-height:` the height of a line

The `value` is a number representing the height of one line in ems (em) or pixels (px)

- similar to **leading** in typography

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



{ } FONT TRANSFORM

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

`letter-spacing:` change word tracking 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

From moment-zero.com

{ } COLORS

`color:` changes the color of **text**

`background-color:` sets the **background** color of an element

- Color `value` can be a **name**, **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 {
    color: #fff;
    background-color: #000;
    font-family: Impact, sans-serif;
}
```

This is a paragraph

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

{ } TEXT-ALIGN

You can change how things are aligned using the `text-align` property.

Values:

- `center`
- `left`
- `right`
- `justify`

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

{ } STATES IN CSS

CSS also allows you to apply styles based on the **state** of an element

- Being **hovered** over with a mouse
- Gaining **focus** via tabbing or clicking

This is known as a CSS pseudo-class (because it doesn't really exist)

Whenever you see a `:` in a selector, that style will only apply to elements that are in that state

{ } PSEUDO EX-PSAMPLE

```
p:hover { background-color: #999; }
```

This paragraph gets fancy when you hover over it

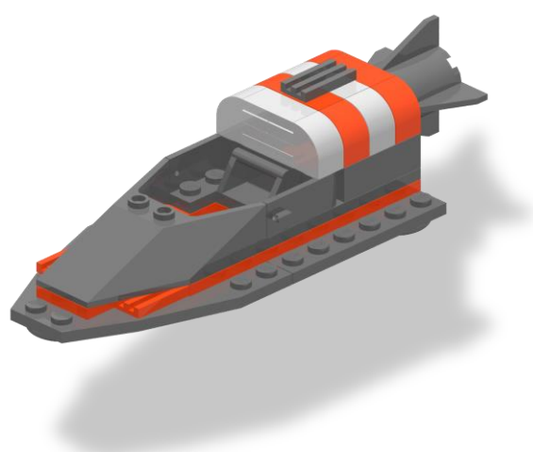
{ } LINK STATES

Links have two additional states

- `:visited` only applies once the user has viewed the page that is being linked to
- `:link` only applies to unvisited pages

`a:link`

`a:visited`



`a:hover`

`a:active`

{ } LINK STATES

```
a { color: blue; }

a:visited { color: gray; }

a:hover {color: purple; }

a:active { color: yellow; }
```

Let’s inspect a live demo of how this looks

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
- Make something change when hovered

HOMEWORK

For most of the class, we will be updating the same website.

So make sure the site you made today is something you will be interested in working on for the next few weeks!

- If not, start a new project using what you learned today

HOMEWORK

Before next week, create **one new HTML page** called about.html

- Provide some info about yourself
- Include at least 3 different types of tags
- Use CSS to make it look nice

Email the html file to
beckjohnson@gmail.com

"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*