HTML & CSS: LEVEL 1

Instructor: Beck Johnson

Week 1

INTRODUCTIONS

- Who are you?
- What do you do/study/etc?
- What is your experience with web development?
- What are you hoping to get out of this class?

CLASS SCHEDULE

- Thursdays from Feb 23 to March 23, from 6:30-9:30pm
- 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
- Preparing images for use on the web
- Intro to related technologies (Javascript, Git)

SESSION OVERVIEW

- Code editors and web authoring tools
- Anatomy of an HTML document
- Basic HTML Elements
- Basic CSS
- Code and file organization
- Build your first webpage!

ODDS AND ENDS

kweeket.github.io

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

DESIGN

critical to the best user experience

CODE

brings content and design to life

CONTENT

What am I presenting?

TEXT

- Articles
- Links
- Lists

MEDIA

- Images
- Videos
- Audio

DESIGN

What is the experience?

USER EXPERIENCE

- Layout
- Navigation
- User flows
- Ease of use

GRAPHIC DESIGN

- Colors
- Fonts
- Backgrounds
- Icons

</> CODE

How does the computer understand?

HTML structures and organizes content

CSS stylizes the content and creates layout

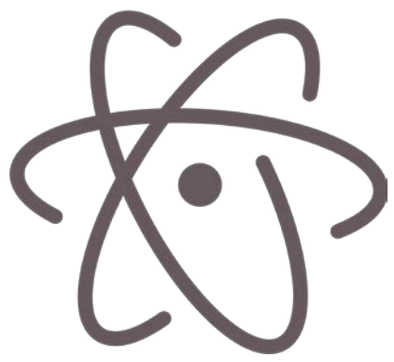
JAVASCRIPT adds interactivity

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: Developer Tools

- Right-click > Inspect
- F12 key

Safari: Developer Tools

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

Internet Explorer: Developer Tools

- 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>

<br/> 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)

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

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"/>
```

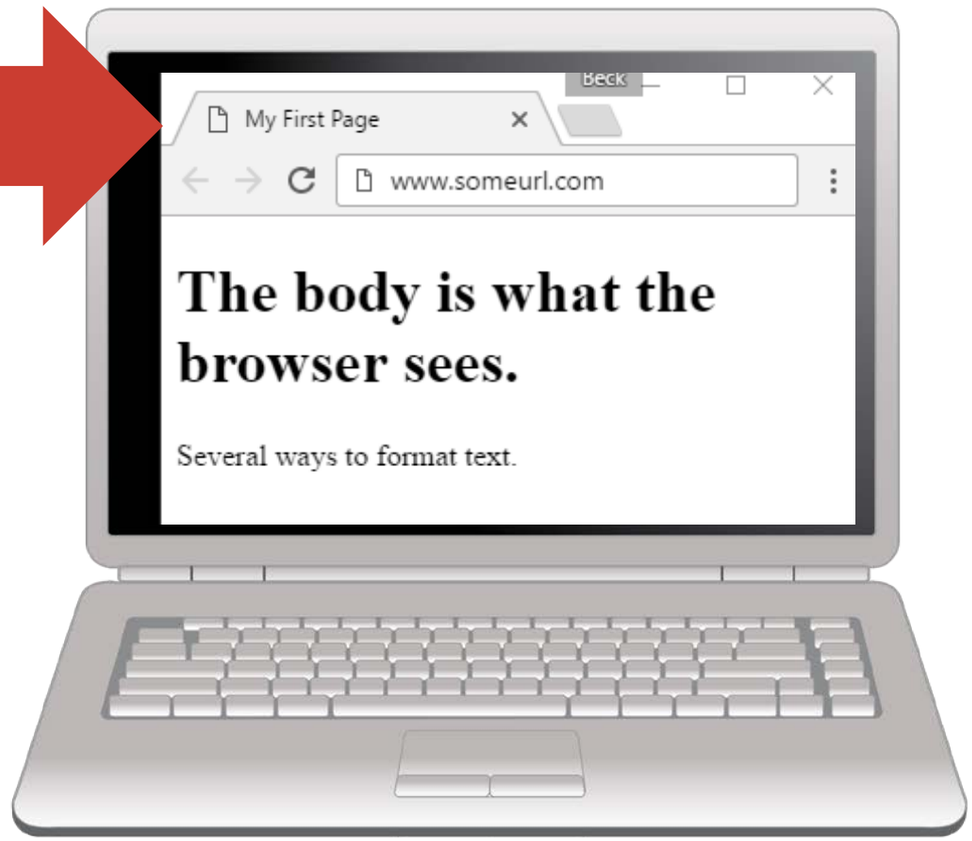
- Used to specify "meta" information to the browser like page description, author, search engine keywords, and character encoding
- UTF-8 represents Unicode, a system to handle text consistently in a variety of languages.

```
<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

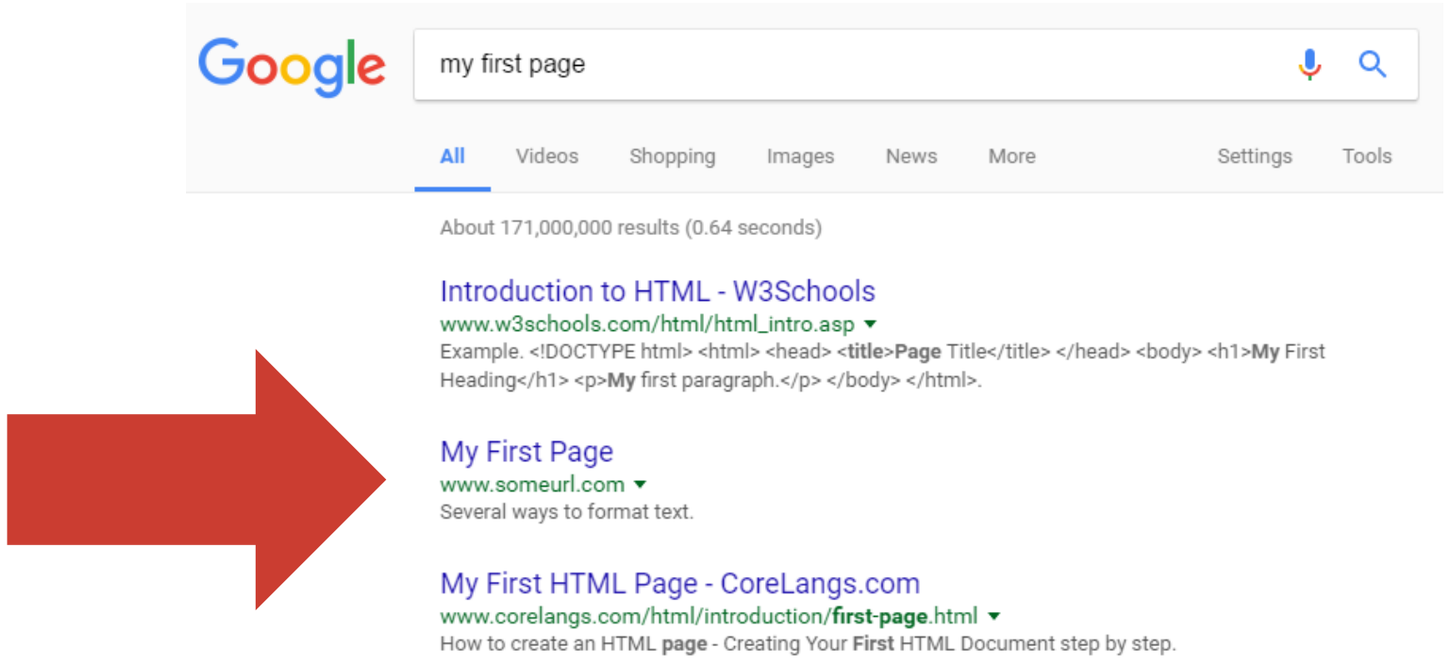


- **Required** inside <head>

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>
```

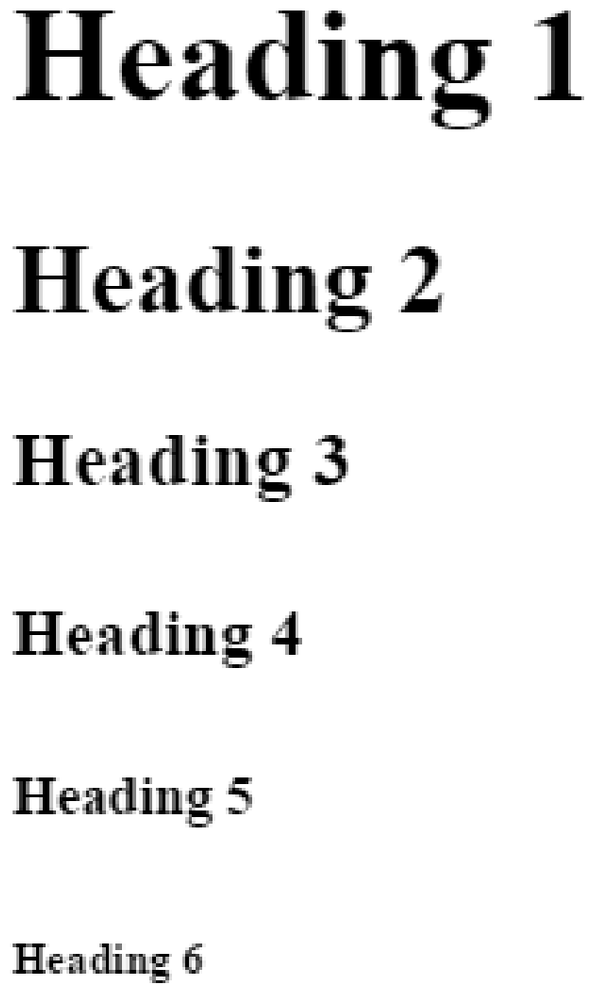
Headings range from most important to least important

<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>
```



PARAGRAPHS

```
<p>Hi! I’m a paragraph!</p>
```

- Most text in the document
- Browsers automatically add space around `<p>` elements (although this can be changed with CSS)

FORMATTING

`<em>` indicates *emphasis*

- By default, this displays as `<em>`*italic*`</em>`

`<strong>` indicates **importance**

- By default, this displays as `<strong>`**bold**`</strong>`

LIST ELEMENTS

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

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

- Puppies
- Kittens

LIST ELEMENTS

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

Ordered lists `<ol>` 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** `<li>` directly

IMAGES

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

- 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

```
<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" or link

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

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 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>

<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

FILE ORGANIZATION

FILE ORGANIZATION

- If you structure your site correctly, you are one step closer to faster updates

- The next person to work on or look at your code will be able to understand what you've done and where to find things

FILE ORGANIZATION

Typical files in a website include:

HTML files (.html)
CSS files (.css)
Javascript files (.js)
Images (.png, .jpg, .gif)

- HTML should usually go in the **main** (root) directory

- Make **subdirectories** for media, CSS, and Javascript files

FILE NAMING RULES

- Use a consistent naming convention when naming files and folders

- Capitalization matters - kittens.png is **not** the same as KITTENS.png

- Use only letters, numbers, hyphens (-) or underscores (_).

- No spaces in file names

- Your homepage is **index.html** by default

PRACTICE TIME!

ASSIGNMENT

Create a website that about something that interests you

- At least 2 pages that are linked to each other
- Include a link to an outside website. Bonus: have the link open in a new tab
- Use 3 heading tags and at least one paragraph
- Use at least one list
- Show at least 2 images - one local and one remote
- Add one HTML comment
- Validate your website

- CSS is a language for specifying how documents are presented to users

- Ability to override the browser's default presentation styles with custom versions

- Provides consistent and scalable ways to style single elements, single pages, or entire websites

- Separates look and feel from content/markup

- 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

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

```
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!

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

"All paragraphs will have blue text "

```
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`)

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

- Multiple `properties` can be defined for a single `selector`, each separated by a semi-colon (`;`)

`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

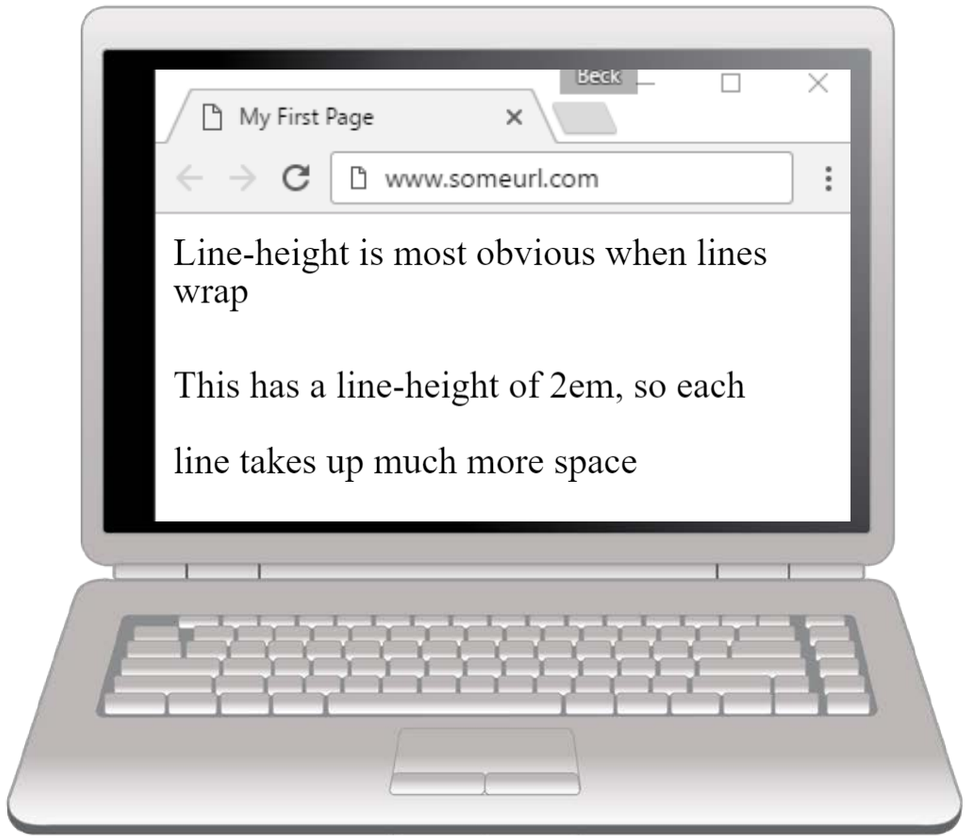
```
p {
    font-family: Arial, Helvetica, sans-serif;
}
```

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

`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)

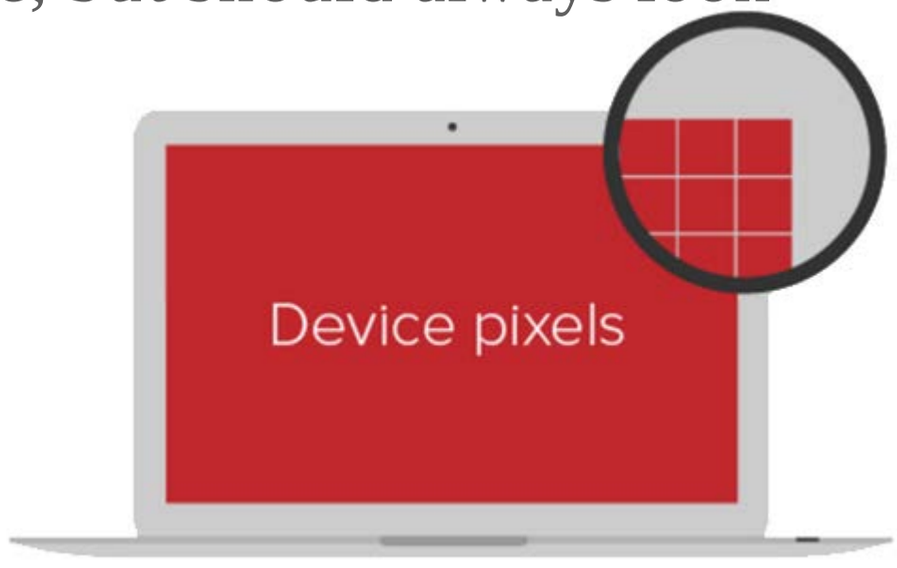
```
p { font-size: 14px; }
```

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

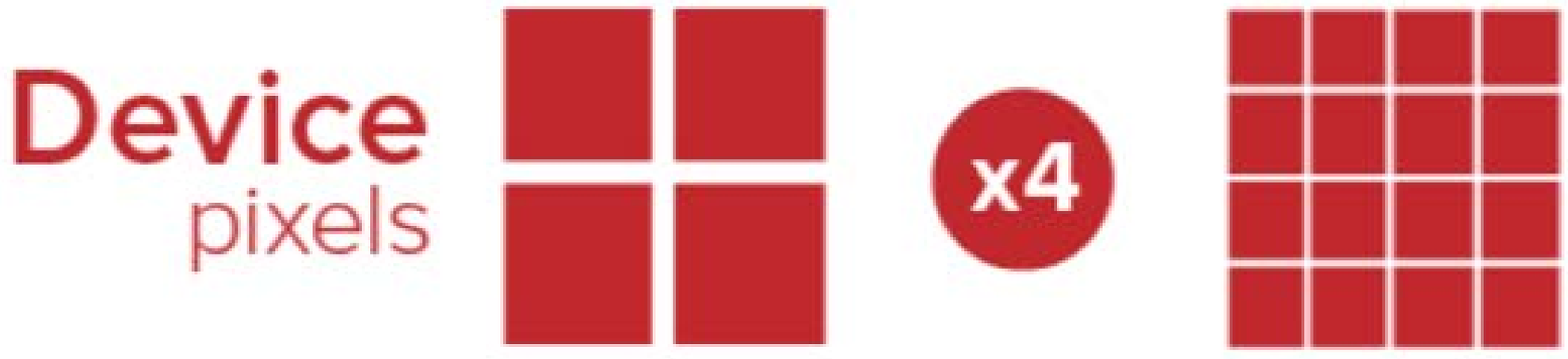
- **px** is an abstract unit that isn't related to font height and isn't a physical unit of measurement

- 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"



	Standard		Retina
CSS pixels	height: 2px width: 2px	==	height: 2px width: 2px



- **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

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

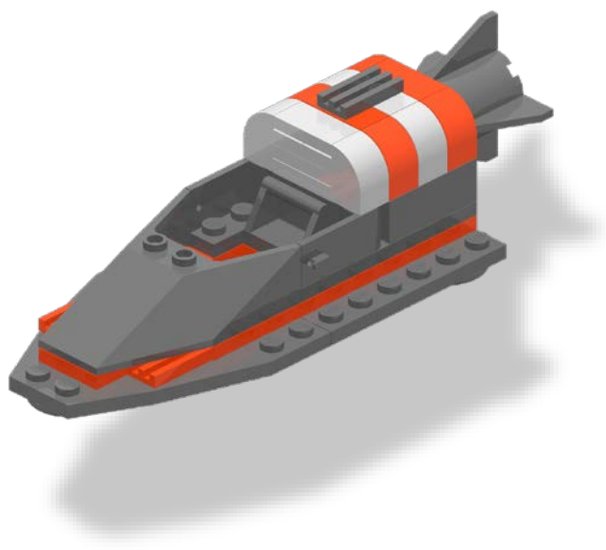
- `color:` changes the color of text
- `background-color:` sets the background color of an element
- Color `value` can be set using **names**, **HEX, RGB,** or **RGBA**
  - Name: `white`
  - Hex: `#ffffff`
  - RGB: `rgb(255, 255, 255)`
  - RGBA: `rgba(255, 255, 255, 0.8)`

{} FOUR LINK STATES

You can style a link differently depending on what **state** it's in

a:link

a:visited



a:hover

a:active

{} FOUR 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

{} TEXT-ALIGN

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

Values:

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

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

{} MULTIPLE SELECTORS & PROPERTIES

- You can add multiple `selectors` to a CSS rule
- You can add multiple `properties` to a CSS rule
- Example: style all ordered and unordered lists:

```
ul,
ol {
    font-size: 16px;
    font-weight: bold;
    color: #444444;
}
```

PRACTICE TIME!

PRACTICE

- Add a `<style></style>` section in the `<head>` on your homepage (index.html)
- Make some style changes using CSS
  - Consider changing font color, font family, font size, link color, text alignment, and background colors

"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

- Optional: read chapters 6-7 of *HTML and CSS: Design and Build Websites*