
LET'S GET EVERYTHING SET UP!

1. Navigate to the FEWD 47 Dashboard (saraheholden.com/fewd47/) and download the Lesson 2 starter code and slides. You'll want to keep the dashboard open for other links and resources we'll be referencing in class.
2. Move the starter code and slides from your Downloads folder to the **fewd** folder we added to your Desktop last class.
3. Double-click on `starter_code_lesson_2.zip` to unzip it
4. After you've unzipped, be sure to delete the original .zip file to avoid confusion and make sure you don't unzip it again later!
5. Open the entire **fewd** folder with Sublime Text (either drag and drop the folder on the Sublime icon in the dock on Mac, or open Sublime and go to file > open... and select `starter_code_lesson_2` folder).
6. Log in to the FEWD 47 Slack (fewd47.slack.com) and join the class2 channel.

WEEKLY OVERVIEW

WEEK 1

HTML Basics

WEEK 2

CSS Basics / CSS Box Model

WEEK 3

CSS Layout / Lab

LEARNING OBJECTIVES

- Apply relative paths to `` and `<a>` tags.
- Differentiate between basic web color principles: RGB, RGBA, hex.
- Use CSS to add basic styles to an HTML page.

AGENDA



- Review
- More HTML Basics
- Intro to CSS
- Lab

GA GENERAL ASSEMBLY

CSS BASICS

Sarah Holden

FEWD

QUESTIONS

EXIT TICKET QUESTIONS

- Where can we find out about [networking/community events](#)?
- Where can I find the reference for commands/shortcuts we can use in Sublime? [Mac](#) and [Windows](#)
- I'd be interested in learning more best practices (i.e. formatting) of what we learned.
- What's the correct (or most readable) space and tab style?
- Using [blockquote](#)
- How much can you do with JS? What are the limitations?
- Are there versions of HTML other than HTML5?

EXIT TICKET QUESTIONS

Mac: ⌘ = Command, ⇧ = Shift

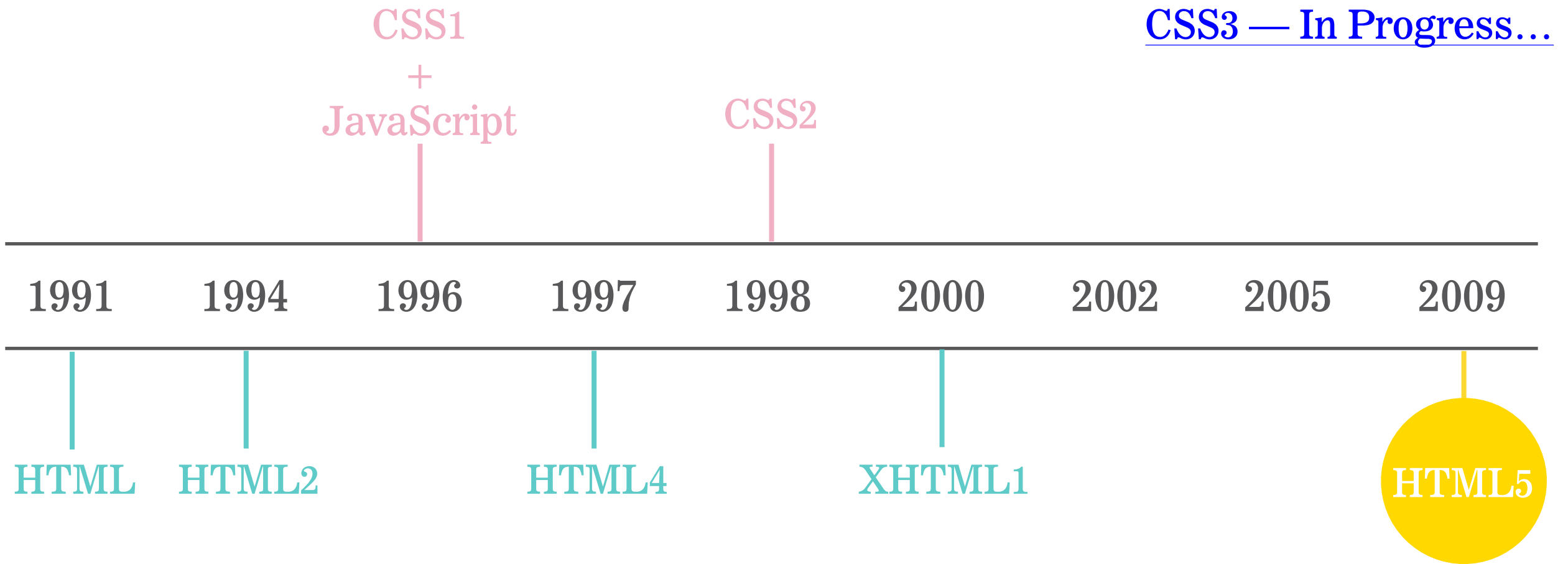
PC: ⌘ = Control, ⇧ = Shift

- Comment Selection/Line: ⌘ + /
- Select Next Match of a Selection: ⌘ + D
- Indent selection: Select text then Tab
- Unindent Selection: Select text then ⇧ + Tab
- Multiple Columns: ⌘ + (number of tabs) Example: ⌘ + 2
- Create multiple cursors: ⌘ + click
- Drag to create multiple cursors: alt + hold down mouse/finger and drag

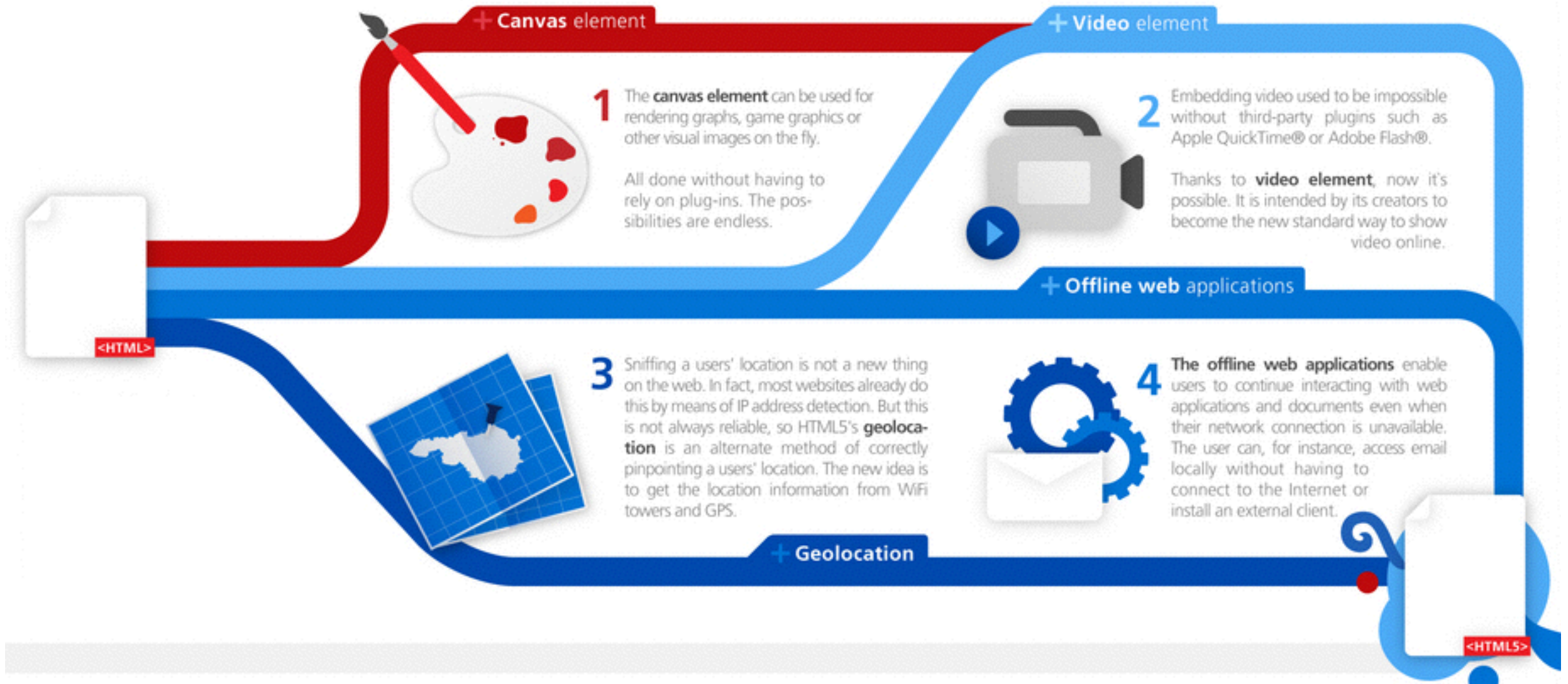
EXIT TICKET QUESTIONS

- Where can we find out about [networking/community events](#)?
- Where can I find the reference for commands/shortcuts we can use in Sublime? [Mac](#) and [Windows](#)
- Using [blockquote](#)
- I'd be interested in learning more best practices (i.e. formatting) of what we learned.
- What's the correct (or most readable) space and tab style?
- How much can you do with JS? What are the limitations?
- Are there versions of HTML other than HTML5?

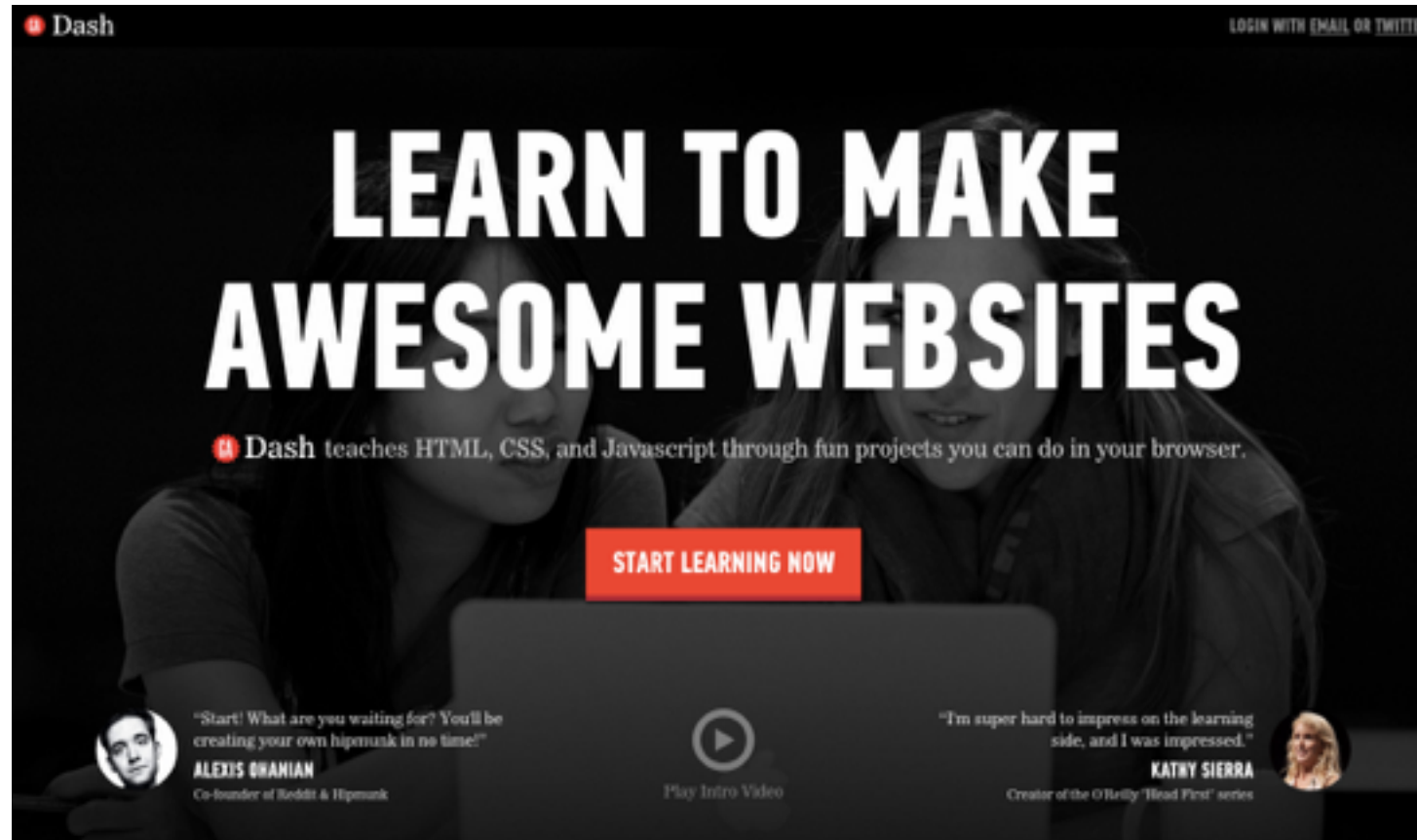
HTML — A BRIEF HISTORY



- HTML5 introduces a number of new elements and attributes.



EXIT TICKET QUESTIONS

The image shows the landing page for 'Dash', a web development learning platform. The background is dark with a grayscale image of two people looking at a laptop. The main heading 'LEARN TO MAKE AWESOME WEBSITES' is in large, bold, white capital letters. Below it, a subheading states 'Dash teaches HTML, CSS, and Javascript through fun projects you can do in your browser.' A prominent orange button with the text 'START LEARNING NOW' is centered. At the bottom, there are two testimonials with circular profile pictures of the speakers. The left testimonial is from Alexis Ohanian, co-founder of Reddit & Hipmunk, who says, 'Start! What are you waiting for? You'll be creating your own hipmunk in no time!'. The right testimonial is from Kathy Sierra, creator of the O'Reilly 'Head First' series, who says, 'I'm super hard to impress on the learning side, and I was impressed.' In the center of the bottom section is a play button icon with the text 'Play Intro Video' below it. The top left corner has the 'Dash' logo, and the top right corner has a link to 'LOGIN WITH EMAIL OR TWITTER'.

EXIT TICKET QUESTIONS

- ▶ How do we submit assignments?
- ▶ Spotify
- ▶ Slack

FEWD

REVIEW

HTML SYNTAX — TAGS

Opening tag

Closing tag



The diagram illustrates the structure of an HTML element. It features the text `<tag name>content</tag name>` in a large, light pink font. The opening tag `<tag name>` is bracketed above by the label "Opening tag". The closing tag `</tag name>` is bracketed above by the label "Closing tag". The content `content` is highlighted in yellow. A large bracket below the entire string `<tag name>content</tag name>` is labeled "Element".

Element

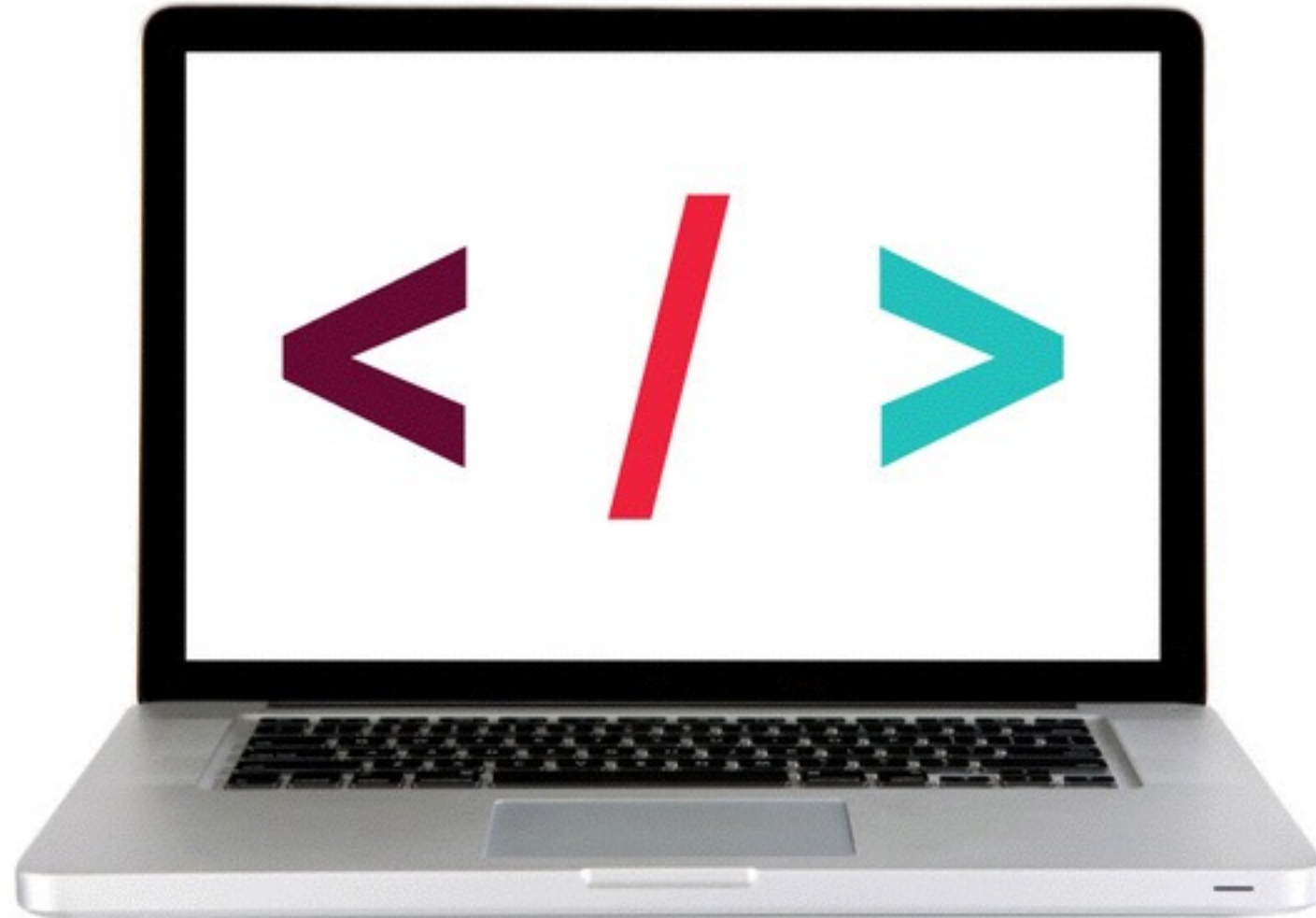
HTML SYNTAX — ATTRIBUTES

Attribute
Name

<tagName **name**=**"value"**></tagName>

Attribute
Value

LET'S TAKE A CLOSER LOOK — MARTY MCFLY



NAVIGATION AND HR

ELEMENT	DESCRIPTION	EXAMPLE
nav	Contains navigational links. For major navigational blocks.	<pre><nav> About Contact </nav></pre>
hr	Horizontal rule	<pre><hr></pre>

HTML ENTITY CODES

Use HTML [entity codes](#) to add special characters to your HTML.

<code>&Tab;</code>	<code>&NewLine;</code>	! <code>&excl;</code>	" <code>&quot; &QUOT;</code>	# <code>&num;</code>	\$ <code>&dollar;</code>	% <code>&percent;</code>	& <code>&amp; &AMP;</code>
' <code>&apos;</code>	(<code>&lpar;</code>) <code>&rpar;</code>	* <code>&ast; &midast;</code>	+ <code>&plus;</code>	, <code>&comma;</code>	. <code>&period;</code>	/ <code>&sol;</code> <code>&#x002F;</code> <code>&#47;</code>
:	; <code>&semi;</code>	< <code>&lt; &LT;</code>	= <code>&equals;</code>	> <code>&gt; &GT;</code>	? <code>&quest;</code>	@ <code>&commat;</code>	[<code>&lqb; &lbrack;</code>
\ <code>&bsol;</code>] <code>&rqb; &rbrack;</code>	^ <code>&Hat;</code>	— <code>&lowbar;</code>	` <code>&grave;</code> <code>&DiacriticalGrave;</code>	{ <code>&lcb; &lbrace;</code>	 <code>&verbar; &vert;</code> <code>&VerticalLine;</code>	} <code>&rcub; &rbrace;</code>

FEWD

MORE HTML BASICS

MORE HTML BASICS

HTML STRUCTURE

HTML STRUCTURE

- ▶ To generate HTML boilerplate in Sublime: `! + tab`.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Document</title>
  </head>
  <body>

</body>
</html>
```

HTML STRUCTURE — DOCTYPE

DESCRIPTION:

- Tells browser file is written in latest version of HTML — HTML5

BEST PRACTICES:

- Must be very first thing in HTML file!

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">  
    <title>Document</title>  
  </head>  
  <body>  
  
  </body>  
</html>
```

HTML STRUCTURE — HTML

DESCRIPTION:

- ALL of our HTML code should go within these tags.

BEST PRACTICES:

- Directly after DOCTYPE
- Opening tag — Line 2
- Closing tag — Last line

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="UTF-8">  
    <title>Document</title>  
  </head>  
  <body>  
  
  </body>  
</html>
```

HTML STRUCTURE — HEAD

DESCRIPTION:

- For behind-the-scenes info
- Metadata that's not displayed
- Info used by browser, search engines

BEST PRACTICES:

- Opens — right after opening html tag
- Closes — right before opening body tag
- Only one in each HTML file

```
<!DOCTYPE html>

<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>Document</title>
  </head>
  <body>

</body>

</html>
```

HTML STRUCTURE — META CHARSET

DESCRIPTION:

- Tells our browser which character set to use
- Should always use UTF-8

BEST PRACTICES:

- Required
- Goes inside of head tags

```
<!DOCTYPE html>

<html lang="en">

<head>
  <meta charset="UTF-8">
  <title>Document</title>
</head>

<body>

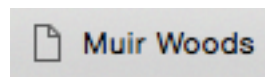
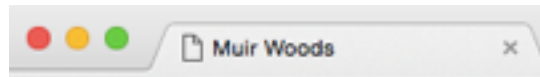
</body>

</html>
```


HTML STRUCTURE — TITLE

DESCRIPTION:

- Tells browser what site is called
- Used for browser tab
- Used for browser bookmarks
- Used by search engines



BEST PRACTICES:

- Required
- Goes inside of head tags

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

HTML STRUCTURE — BODY

DESCRIPTION:

- Wraps all content for our site
- Everything we want displayed in the browser window gets placed between the body tags — all our HTML elements such as `<h1>`, `<a>`, `<p>`, etc.

BEST PRACTICES:

- Required
- Opens — Right after closing head tag
- Closes — Right before closing html tag
- Only one in each HTML file

```
<!DOCTYPE html>

<html lang="en">

  <head>

    <meta charset="UTF-8">

    <title>Document</title>

  </head>

  <body>

    </body>

</html>
```

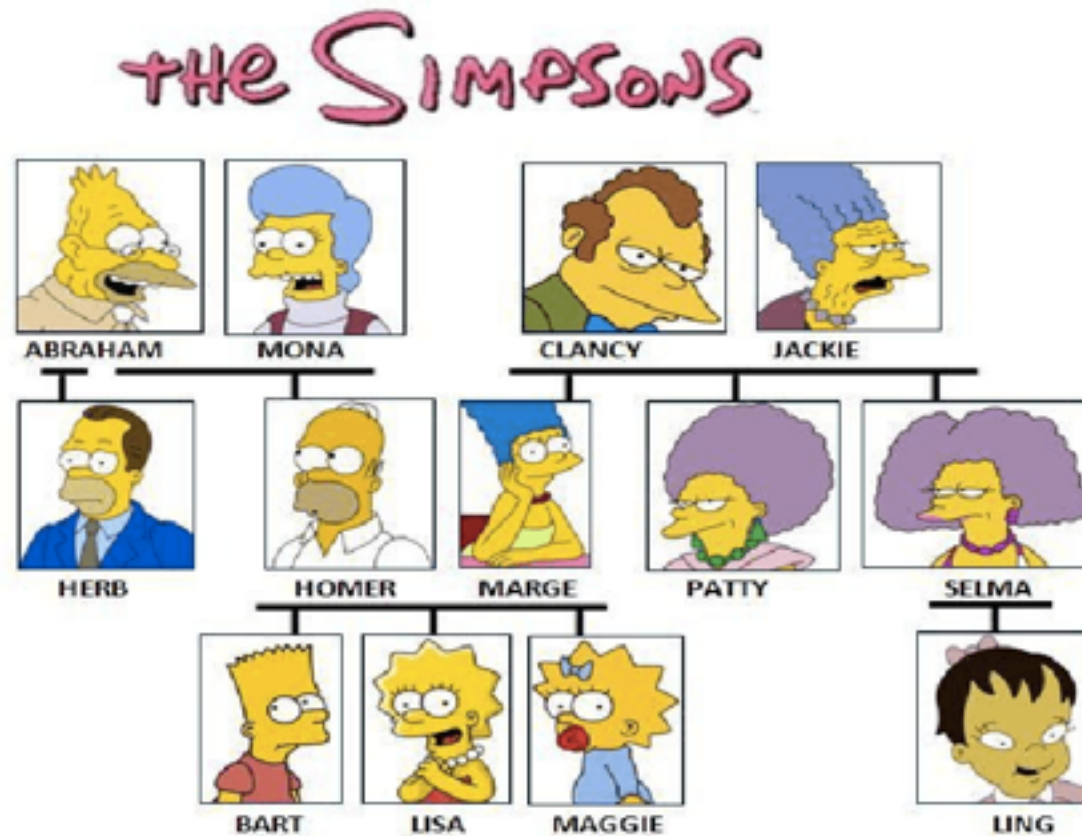
PAGE STRUCTURE — NESTING

- In our HTML files, there are certain tags that "live" or get placed inside other tags.
- `` elements get placed between opening and closing `` tags.
- We say that the `` "wraps" our `` elements.
- We can also say that our `` elements are "nested" inside our `` element.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <title>The Evolution of Denim</title>
  </head>
  <body>
    <h1>The Evolution of Denim</h1>
    <ul>
      <li>Dark Wash <a href="linkgoeshere">Jeans</a></li>
      <li>Stone Wash</li>
      <li>Chambray</li>
    </ul>
  </body>
</html>
```

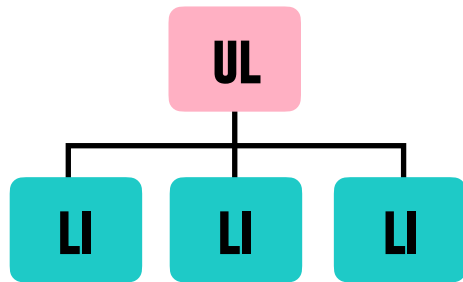
PAGE STRUCTURE — RELATIONSHIPS

- ▶ We often use the same terms we would use to describe relationships between family members to describe the relationships between HTML elements.



PAGE STRUCTURE — NESTING

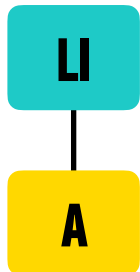
- ▶ Here we can say that our `` is the **parent** of our ``s
- ▶ We can also say that our ``s are **children** of the ``



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>The Evolution of Denim</title>
</head>
<body>
  <h1>The Evolution of Denim</h1>
  <ul>
    <li>Dark Wash <a href="linkgoeshere">Jeans</a></li>
    <li>Stone Wash</li>
    <li>Chambray</li>
  </ul>
</body>
</html>
```

PAGE STRUCTURE — NESTING

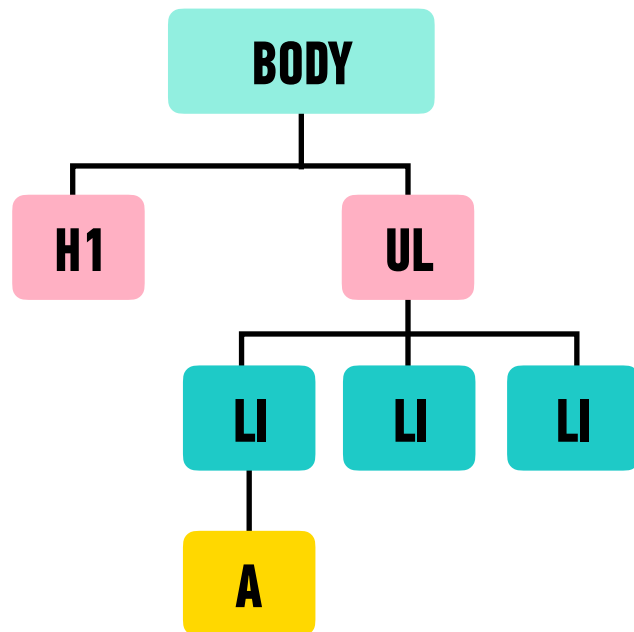
- ▶ Similarly, we can have an `<a>` tag that is nested inside, or wrapped by, our `` element.



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>The Evolution of Denim</title>
</head>
<body>
  <h1>The Evolution of Denim</h1>
  <ul>
    <li>Dark Wash <a href="linkgoeshere">Jeans</a></li>
    <li>Stone Wash</li>
    <li>Chambray</li>
  </ul>
</body>
</html>
```


PAGE STRUCTURE — NESTING

- ▶ Similarly, we can say that all of our HTML content, our h1, our ul, our li elements, are "wrapped" by the body, or "nested" inside the body since they are within the opening and closing body tags

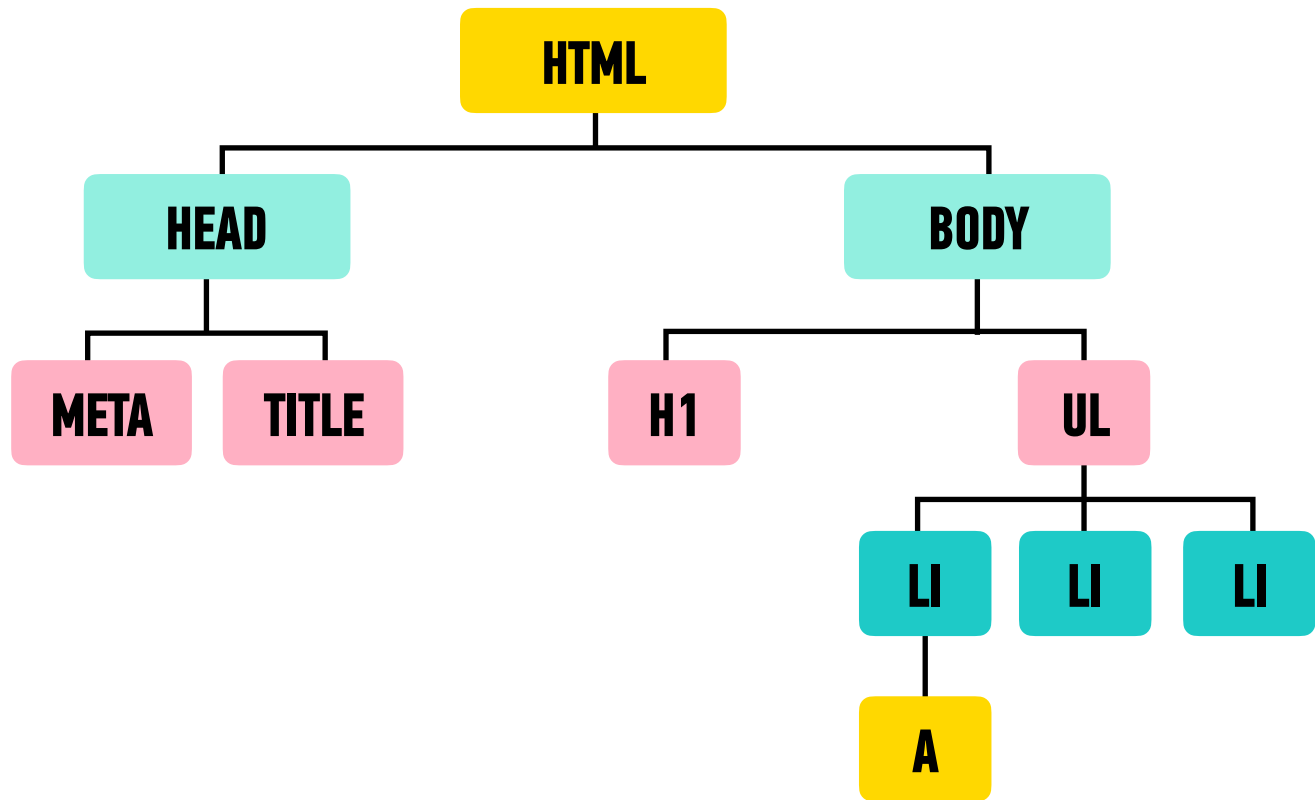


body

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>The Evolution of Denim</title>
</head>
<body>
  <h1>The Evolution of Denim</h1>
  <ul>
    <li>Dark Wash <a href="linkgoeshere">Jeans</a></li>
    <li>Stone Wash</li>
    <li>Chambray</li>
  </ul>
</body>
</html>
```

HTML

DOM TREE



```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>The Evolution of Denim</title>
</head>
<body>
  <h1>The Evolution of Denim</h1>
  <ul>
    <li>Dark Wash <a href="linkgoeshere">Jeans</a></li>
    <li>Stone Wash</li>
    <li>Chambray</li>
  </ul>
</body>
</html>
```

ACTIVITY



EXERCISE

KEY OBJECTIVE

- Be able to describe relationships between elements.

LOCATION

- Starter code > dom_tree

TIMING

2 min

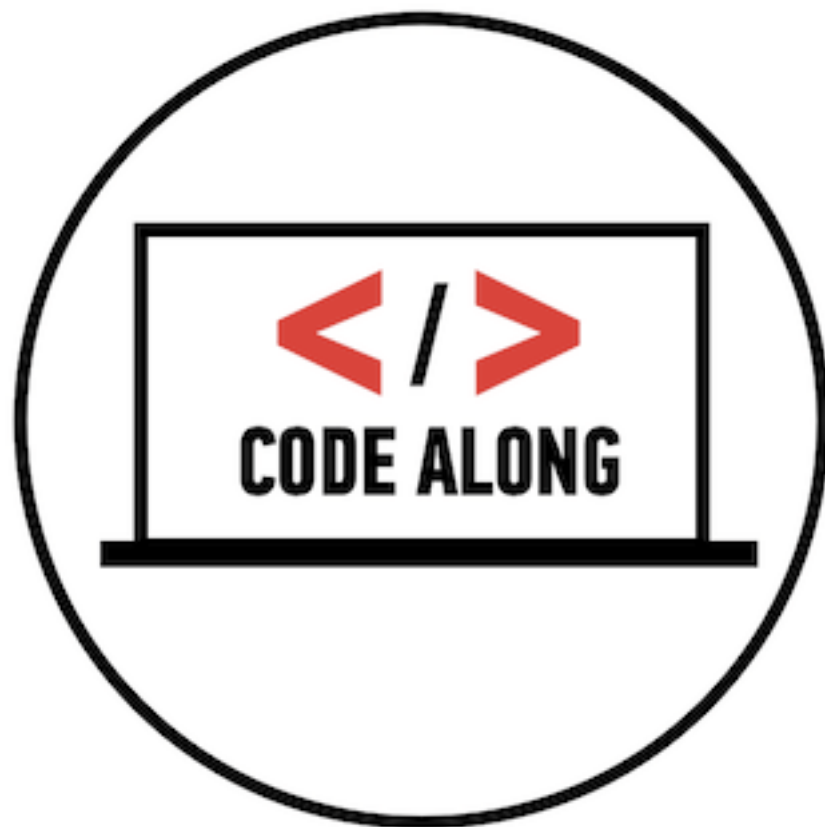
1. Which elements are "parents" of other elements? What are their "children"? Descendants?
2. Which Elements are "children"? What are their parents? What are their ancestors?

PAGE STRUCTURE — INDENT!

- ▶ It is best practice to indent any tags that are nested inside other tags.
- ▶ This makes our code easier to read and will help us see if we forget a closing tag.
- ▶ Indenting getting out of hand? In Sublime — Edit > Line > Reindent

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>The Evolution of Denim</title>
</head>
<body>
  <h1>The Evolution of Denim</h1>
  <ul>
    <li>Dark Wash <a href="linkgoeshere">Jeans</a></li>
    <li>Stone Wash</li>
    <li>Chambray</li>
  </ul>
</body>
</html>
```

CODE ALONG



HTML STRUCTURE



MORE HTML BASICS

IMAGES

IMAGES – THE IMG ELEMENT

Images are added to the page with the img element

- **Void element** — Doesn't need a closing tag
- Two *required* attributes — src and alt



```

```

IMAGES

The **src** attribute tells the browser where it can find an image.



Path to the image file

```

```

IMAGES

The **alt** attribute provides a text description of the image that:

- Replaces the image if it doesn't load
- Is used by screen readers

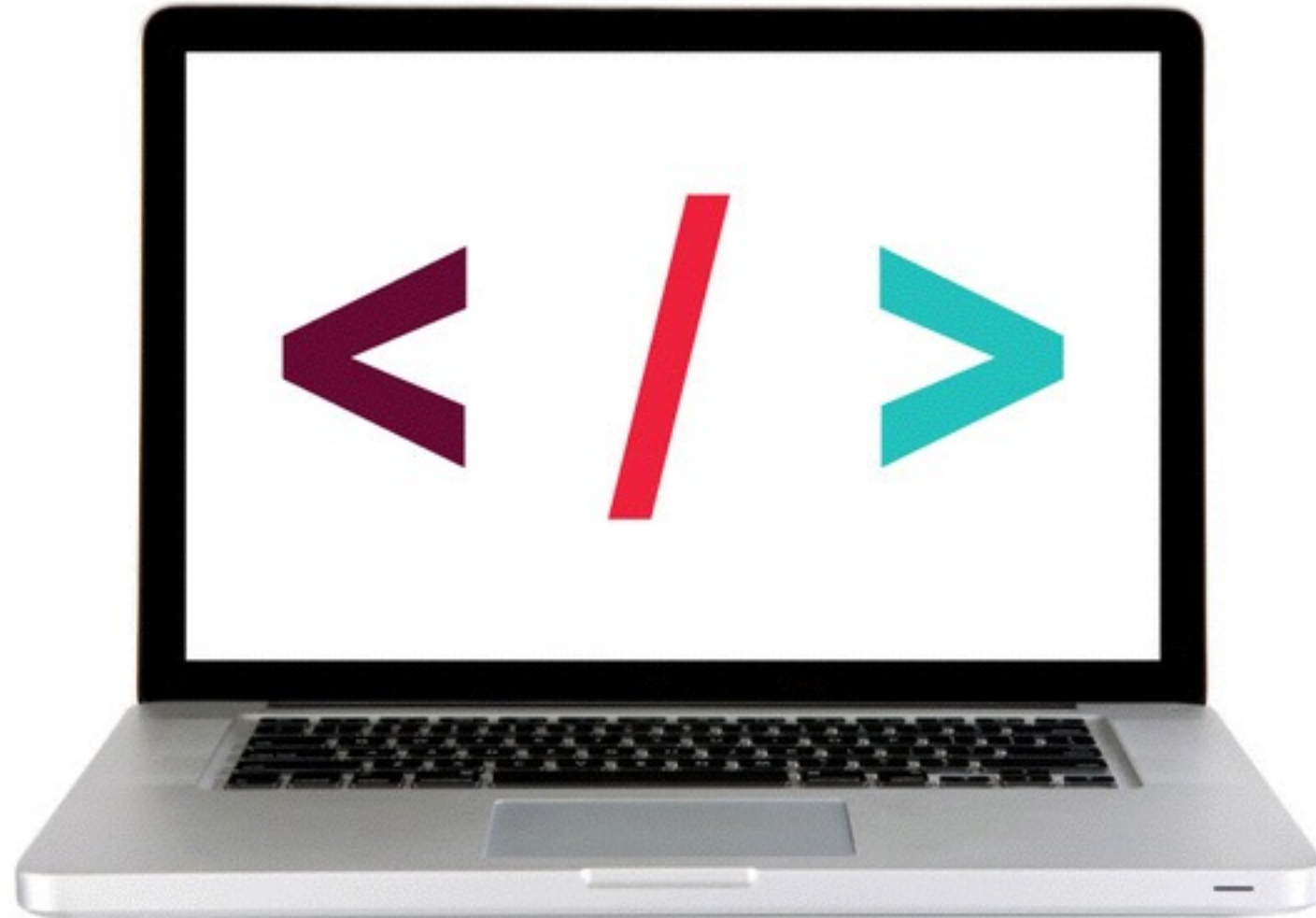


Text description

```

```

LET'S TAKE A CLOSER LOOK



MORE HTML BASICS

URLS

TYPES OF URLS

- There are two main types of URLs:



ABSOLUTE



RELATIVE

LINKING TO OTHER SITES – ABSOLUTE URLS

ABSOLUTE

WHEN YOU LINK TO ANOTHER SITE:

- Value of the href attribute will be the *full web address* for the site
- This is known as the **absolute URL**.

Absolute URL

```
<a href="http://www.amazon.com">Amazon</a>
```

```

```

LINKING TO OTHER PAGES ON THE SAME SITE — RELATIVE URLS

RELATIVE

WHEN YOU LINK TO ANOTHER PAGE OR FILE ON THE SAME SITE:

- You don't need to specify a domain name in the URL.
- Use **Relative URL**: indicates where pages are *in relation to the current page*

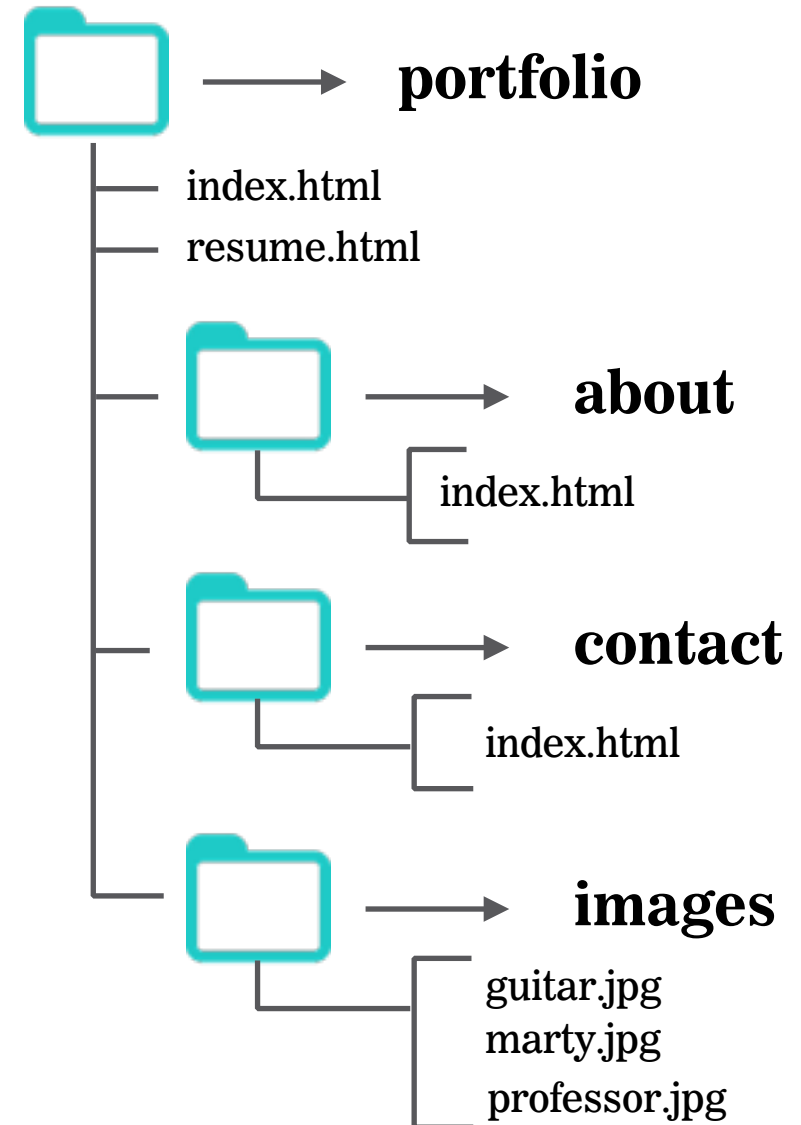
Relative URL

`About Page`

``

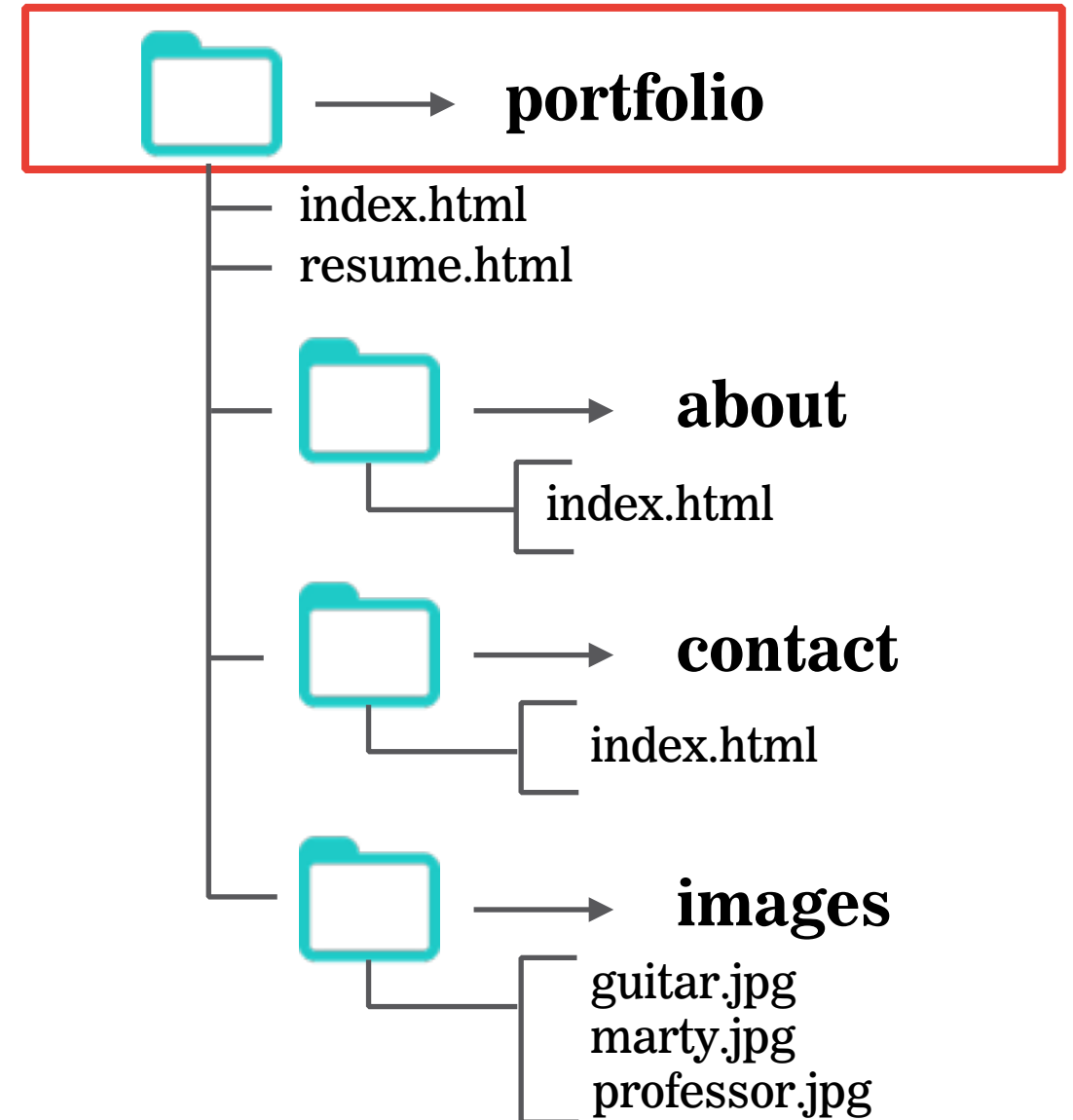
DIRECTORY STRUCTURE

- › On larger sites, it's *best practice* to organize code by placing the files for each major section of the site into a new folder
- › These folders are often referred to as **directories**.

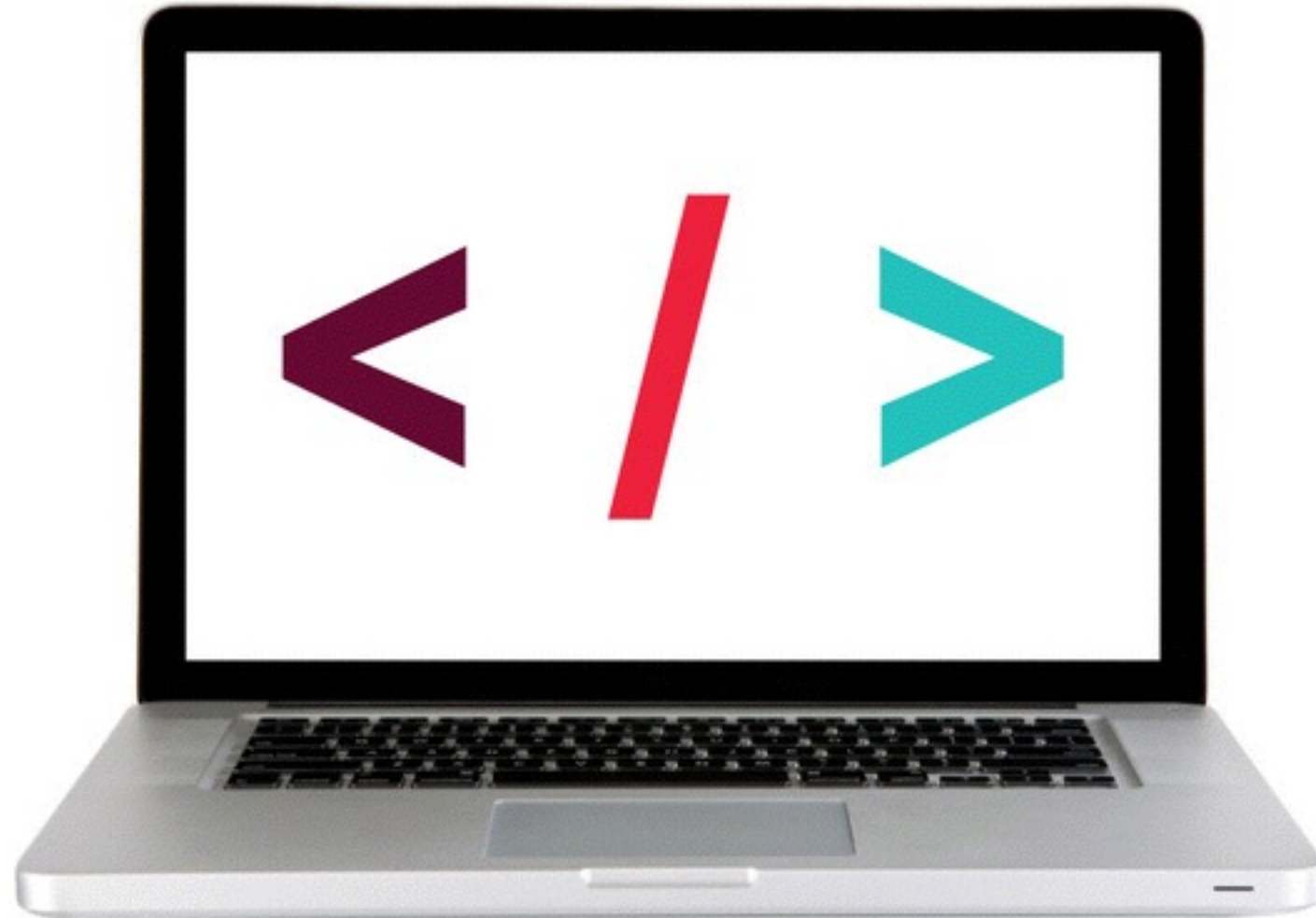


DIRECTORY STRUCTURE — ROOT

- The topmost folder is called the **root** folder. It contains all other files and folders for a website.

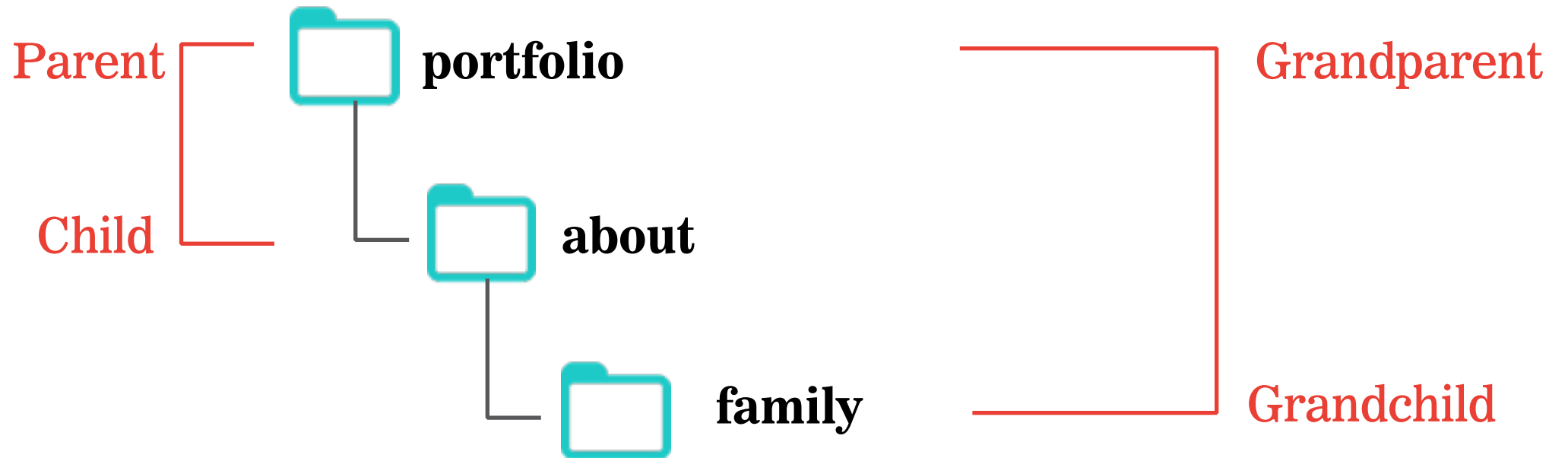


LET'S TAKE A CLOSER LOOK



DIRECTORY STRUCTURE

- › Relationships between folders can be described using similar language to that of a family tree

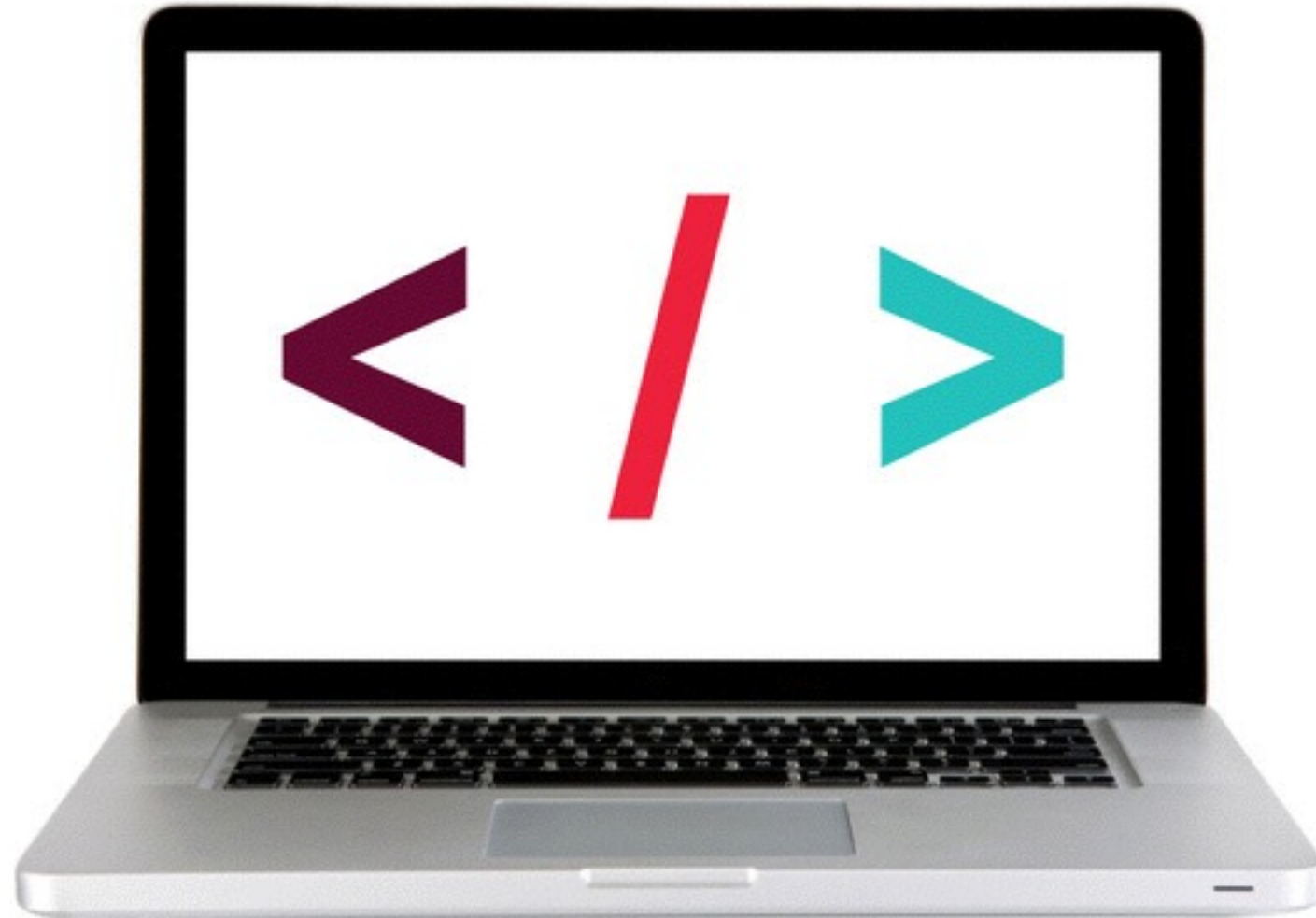


RELATIVE URLS — SAME FOLDER

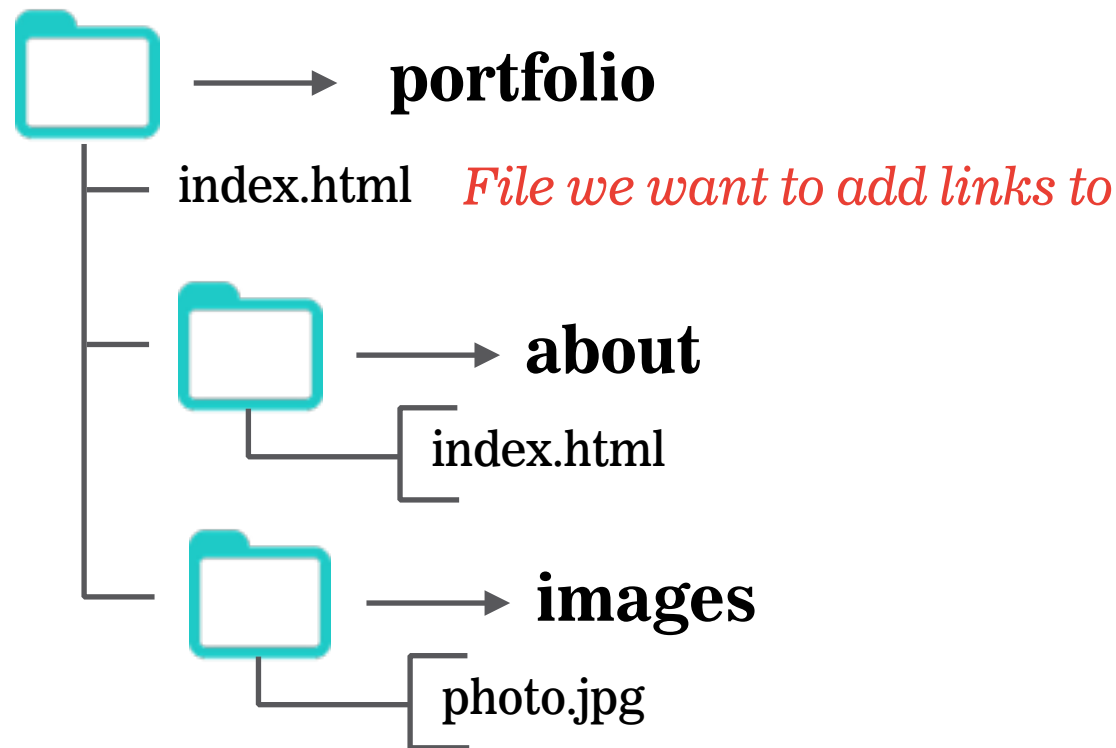


	RELATIVE URL	EXAMPLE
SAME FOLDER	File name	<code>Info</code>
		<code></code>

LET'S TAKE A CLOSER LOOK

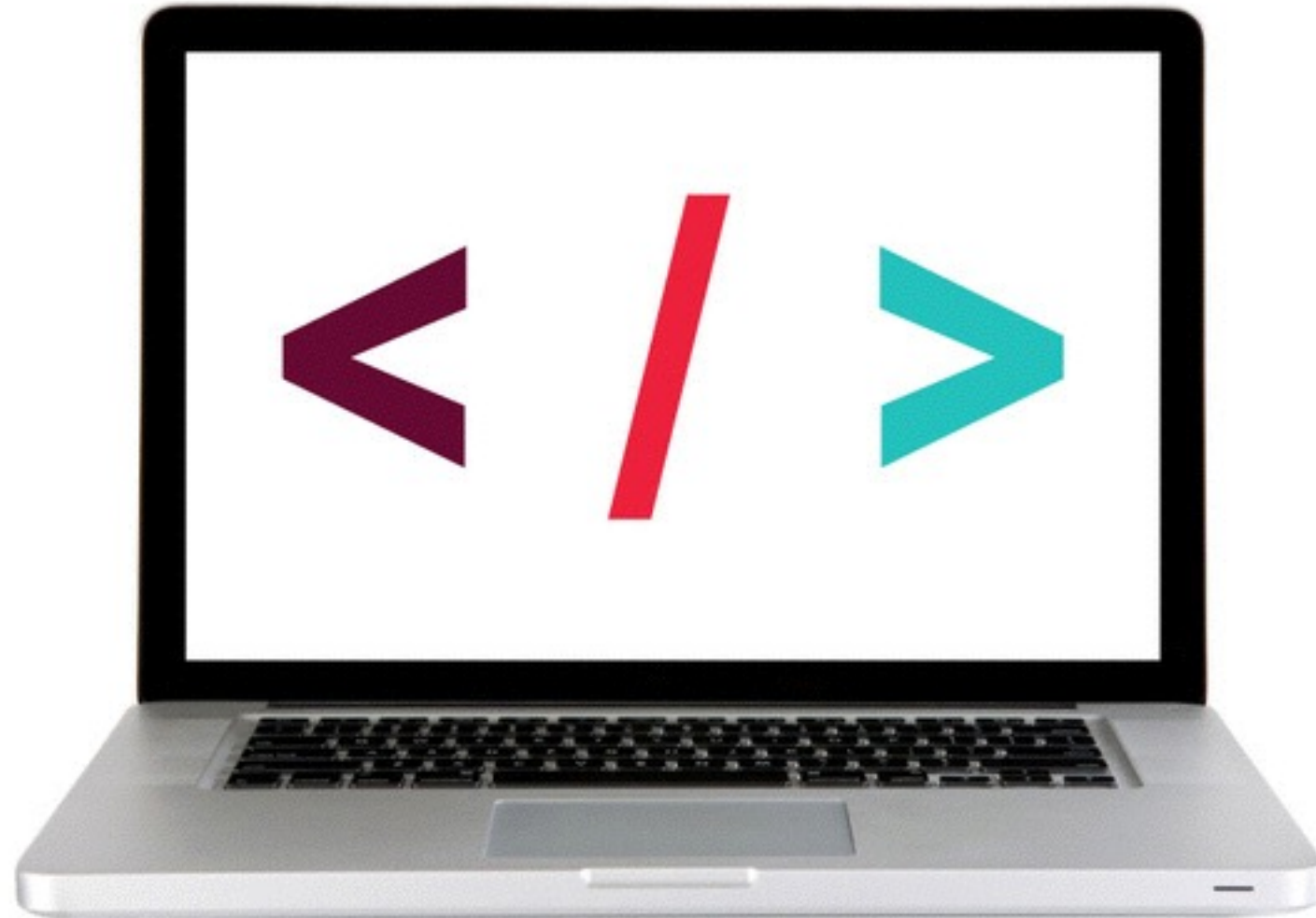


RELATIVE URLS — CHILD FOLDER



	DESCRIPTION	EXAMPLE
CHILD	Name of child folder + / + file name	<code></code>
		<code>About</code>

LET'S TAKE A CLOSER LOOK



ACTIVITY



EXERCISE

KEY OBJECTIVE

- Practice adding relative URLs to a project

KEY OBJECTIVE

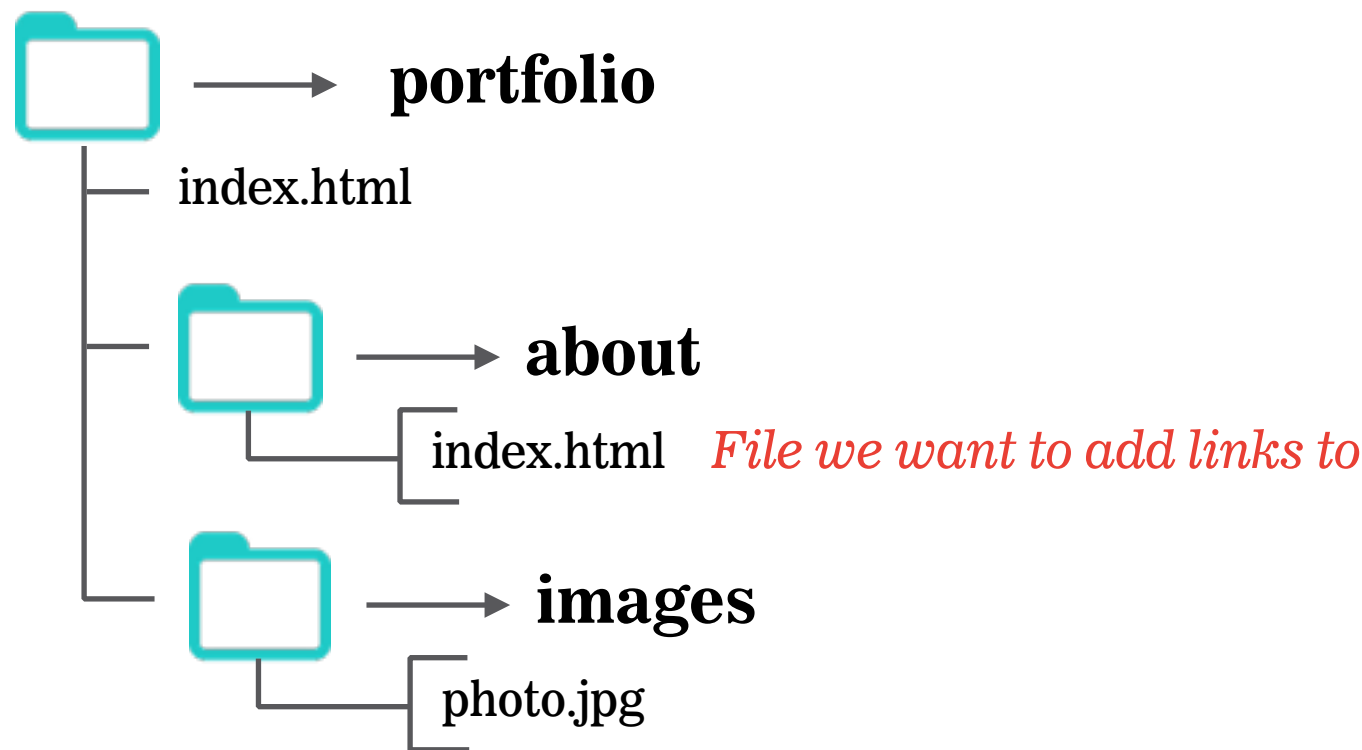
- Starter code > Portfolio Folder

TIMING

4 min

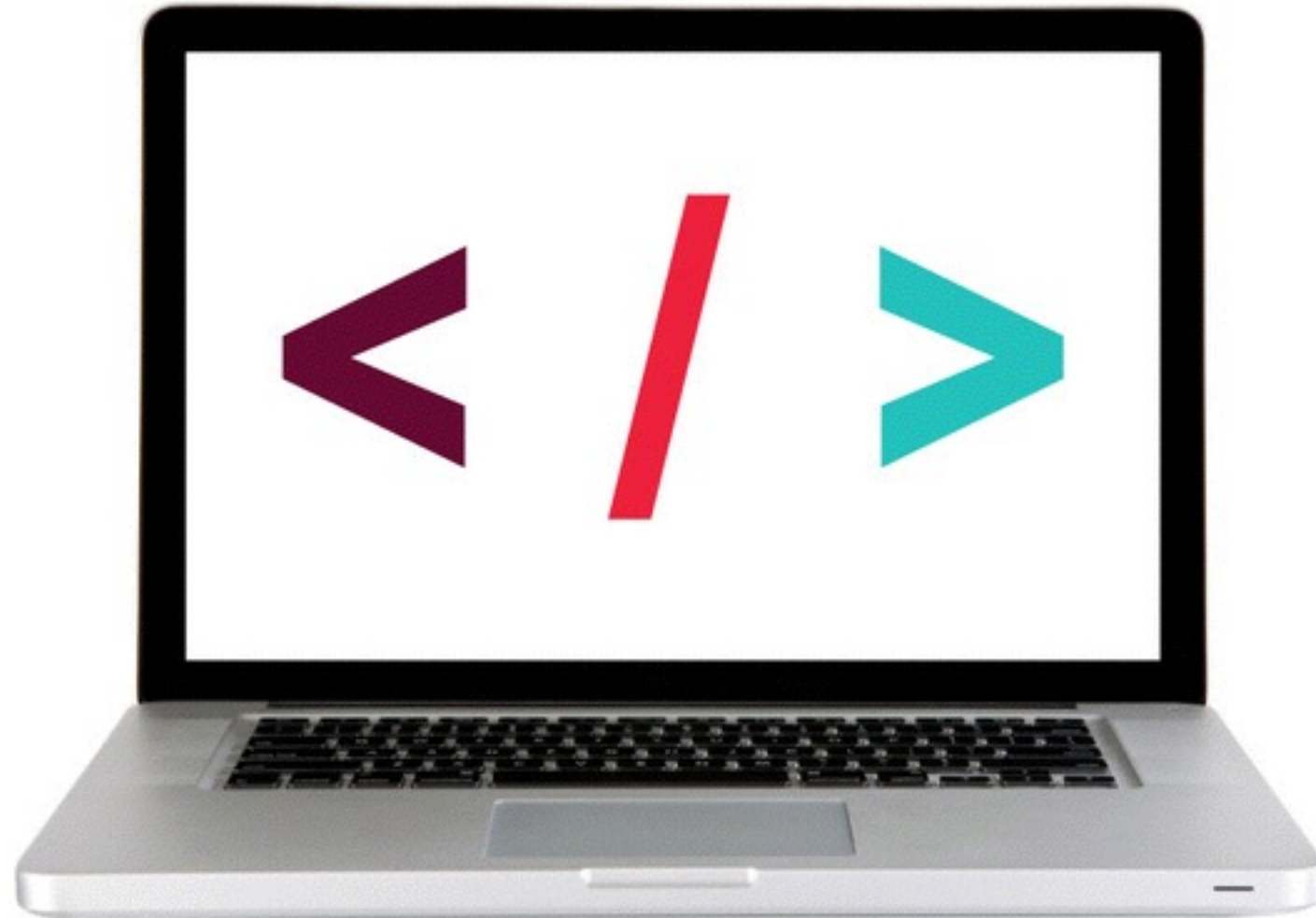
1. With a partner — Follow steps 1 - 3 in resume.html
2. Test in browser!

RELATIVE URLS — CHILD FOLDER



	DESCRIPTION	EXAMPLE
PARENT	../ + path	Home

LET'S TAKE A CLOSER LOOK



ACTIVITY



EXERCISE

KEY OBJECTIVE

- Practice adding relative URLs to a project

KEY OBJECTIVE

- Starter code > Portfolio > Folder

TIMING

4 min

1. With a partner — Follow steps 1 - 4 in About > index.html
2. Test in browser!

RELATIVE URLS

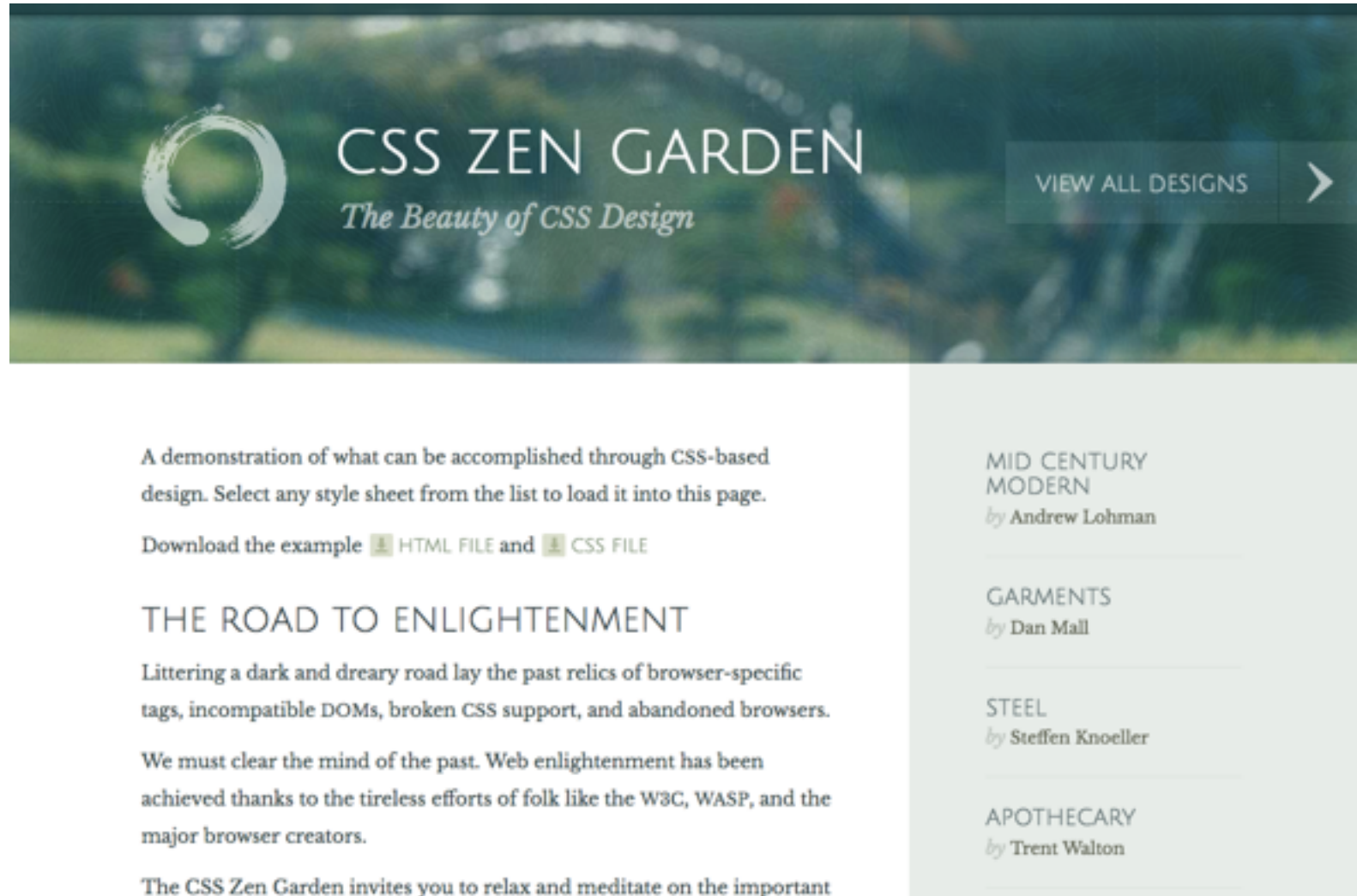
FOLDER FILE IS IN:	DESCRIPTION		EXAMPLE
	SAME	File name	info.html
	CHILD	Name of child folder + / + file name	people/index.html
	GRANDCHILD	Name of child folder + / + Name of grandchild folder + / + file name	people/culture/index.html
	PARENT	../ + path	../index.html
	GRANDPARENT	../.. / + path	../.. /index.html

Note that ../ means to go up one directory, and can be used repeatedly:
../.. / would go up two directories.

HTML BASICS

INTRO TO CSS

WHAT IS CSS?



HTML BASICS

THE BASICS

WHAT IS CSS?

Muir Woods

Keffiyeh next level retro, brunch *sriracha* dreamcatcher
mixtape jean shorts XOXO master cleanse keytar
Kickstarter. Neutra kale chips Vice health goth ethical,
flannel single-origin coffee stumptown meditation
Kickstarter mumblecore yr cronut master cleanse keytar.

Bushwick sartorial pickled, quinoa church-key before they
sold out drinking vinegar put a bird on it readymade organic
lumbersexual. Four dollar toast chia *Intelligentsia* YOLO
Marfa. Migas raw denim photo booth authentic, roof party
shabby chic pop-up flexitarian *skateboard* blog.

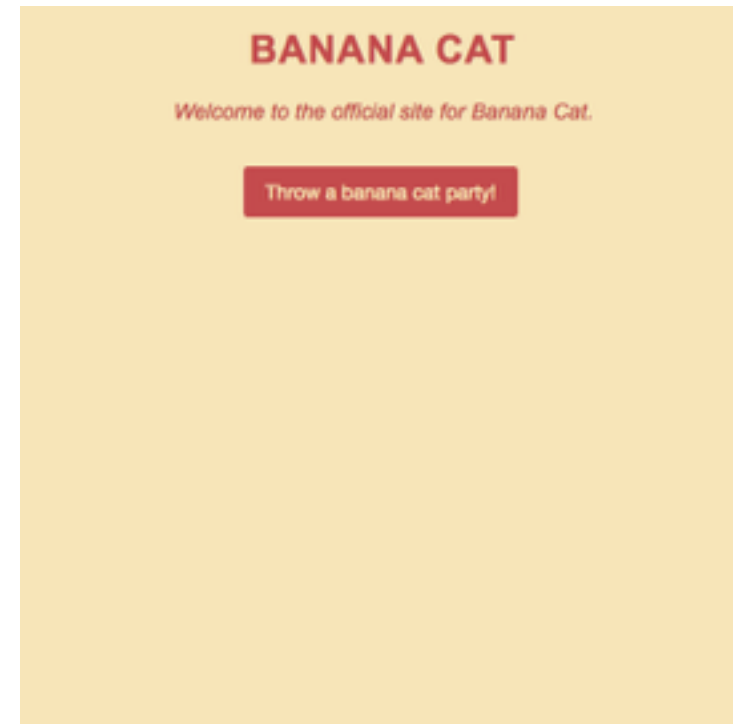
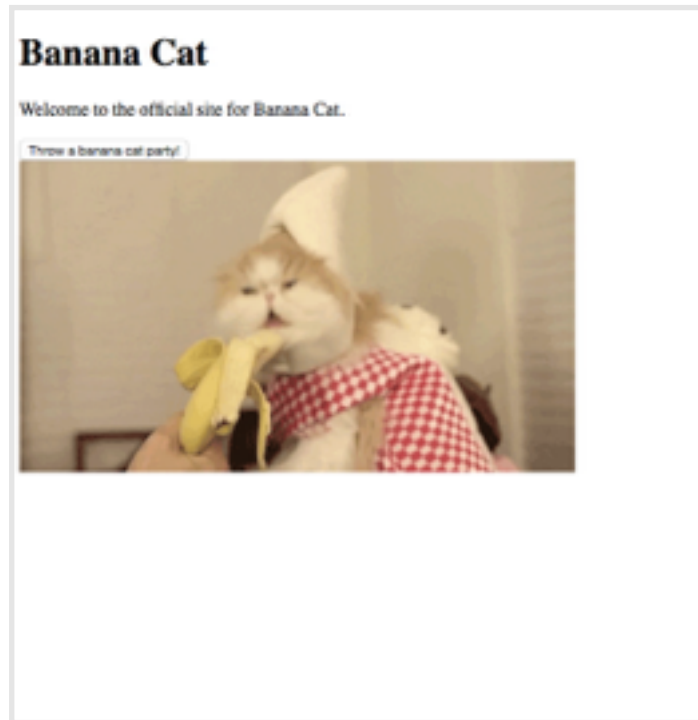
Muir Woods

Keffiyeh next level retro, brunch *sriracha* dreamcatcher
mixtape jean shorts XOXO master cleanse keytar
Kickstarter. Neutra kale chips Vice health goth ethical,
flannel single-origin coffee stumptown meditation
Kickstarter mumblecore yr cronut master cleanse keytar.

Bushwick sartorial pickled, quinoa church-key before they
sold out drinking vinegar put a bird on it readymade organic
lumbersexual. Four dollar toast chia *Intelligentsia* YOLO
Marfa. Migas raw denim photo booth authentic, roof party
shabby chic pop-up flexitarian *skateboard* blog.

WHAT IS CSS?

- CSS is what handles the presentation layer of our webpage.
- CSS allows us to associate style rules with HTML elements.



CSS SYNTAX

Selector

h1{

color: yellow;

}

Declaration

CSS SYNTAX

h1 {

color: yellow;

font-size: 16px;

}

Property

Value

USING INTERNAL CSS — :(

- ▶ You can include CSS rules by placing them inside a `<style>` element, which usually sits inside the `<head>`.

```
<head>
  <meta charset="UTF-8">
  <title>Visit Big Sur</title>

  <style>
    h1 {
      color: yellow;
    }
  </style>
</head>
```

USING EXTERNAL CSS — :)

- `<link>` — tells browser where to find the CSS file that should be used to style the page.
- Should live inside the `<head>` of the document.

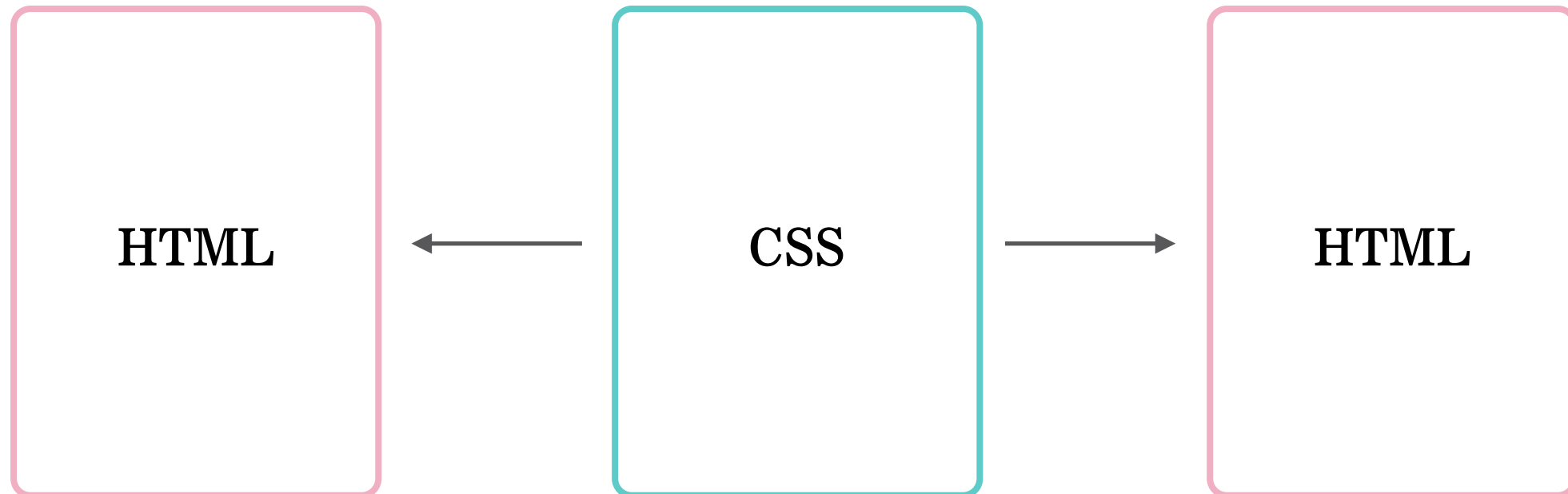
```
<head>  
  <meta charset="UTF-8">  
  <title>Document</title>  
  <link rel="stylesheet" href="css/style.css">  
</head>
```

- **href** — path to the CSS file.
- **rel** — relationship between the HTML page and the file it is linked to.

INTERNAL VS. EXTERNAL CSS

BENEFITS OF USING AN EXTERNAL STYLESHEET:

- Multiple pages can use same stylesheet (Don't repeat yourself!)
- Only have to make changes in one file
- Keep content separate from presentation



ACTIVITY



EXERCISE

KEY OBJECTIVE

- Practice adding relative URLs to a project

KEY OBJECTIVE

- Starter code > Portfolio Folder

TIMING

4 min

1. Add link to main.css file in resume.html and about > index.html

INTRO TO CSS

COLOR

COLOR

PROPERTY	VALUE	DESCRIPTION	EXAMPLE
color	color	Text color	color: #22475E;
background-color	color	Background color	background-color: green;

Muir Woods

Keffiyeh next level retro, brunch *sriracha* dreamcatcher mixtape jean shorts XOXO master cleanse keytar **Kickstarter**. Neutra kale chips Vice health goth ethical, flannel single-origin coffee stumptown meditation Kickstarter mumblecore yr cronut master cleanse keytar.

```
body {  
  background-color: #22475E;  
}
```

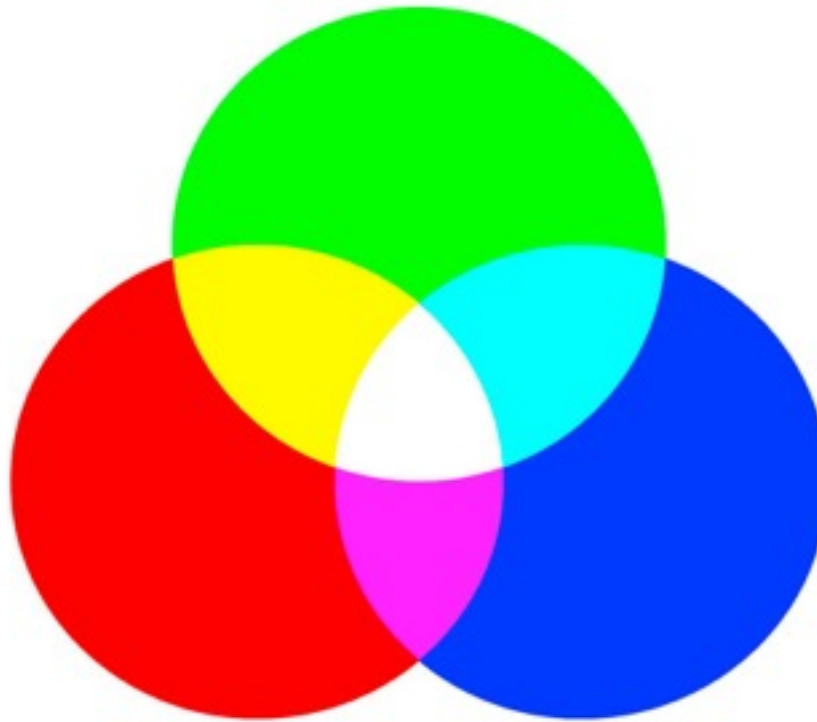
Muir Woods

Keffiyeh next level retro, brunch *sriracha* dreamcatcher mixtape jean shorts XOXO master cleanse keytar **Kickstarter**. Neutra kale chips Vice health goth ethical, flannel single-origin coffee stumptown meditation Kickstarter mumblecore yr cronut master cleanse keytar.

```
h1, p {  
  color: #98D2BF;  
}
```

COLOR

- ▶ Every color on a computer screen is created by mixing amounts of red, green, and blue



COLOR

RGB VALUES

- Values for red, green and blue are expressed as numbers between 0 and 255



rgb(72, 209, 204)

HEX CODES

- Represent values for red, green and blue in hexadecimal (base 16) code



#48D1CC

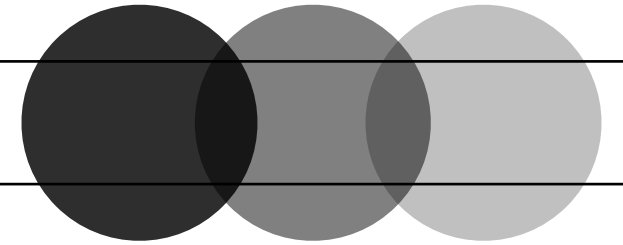
COLOR NAMES

- Colors are represented by predefined names. They are not used very much but are helpful for basic colors such as black and white. [Full list of color names](#)



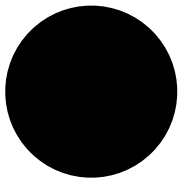
MediumTurquoise

OPACITY

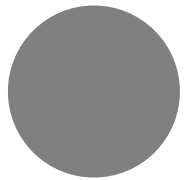


RGBA

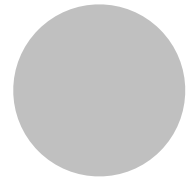
- ▶ RGBA works the same as RGB, except that it takes a 4th value called 'alpha'.
- ▶ This is a value between 0 and 1 which can be used to determine a color's opacity on the page.



`rgba(0, 0, 0, 1)`

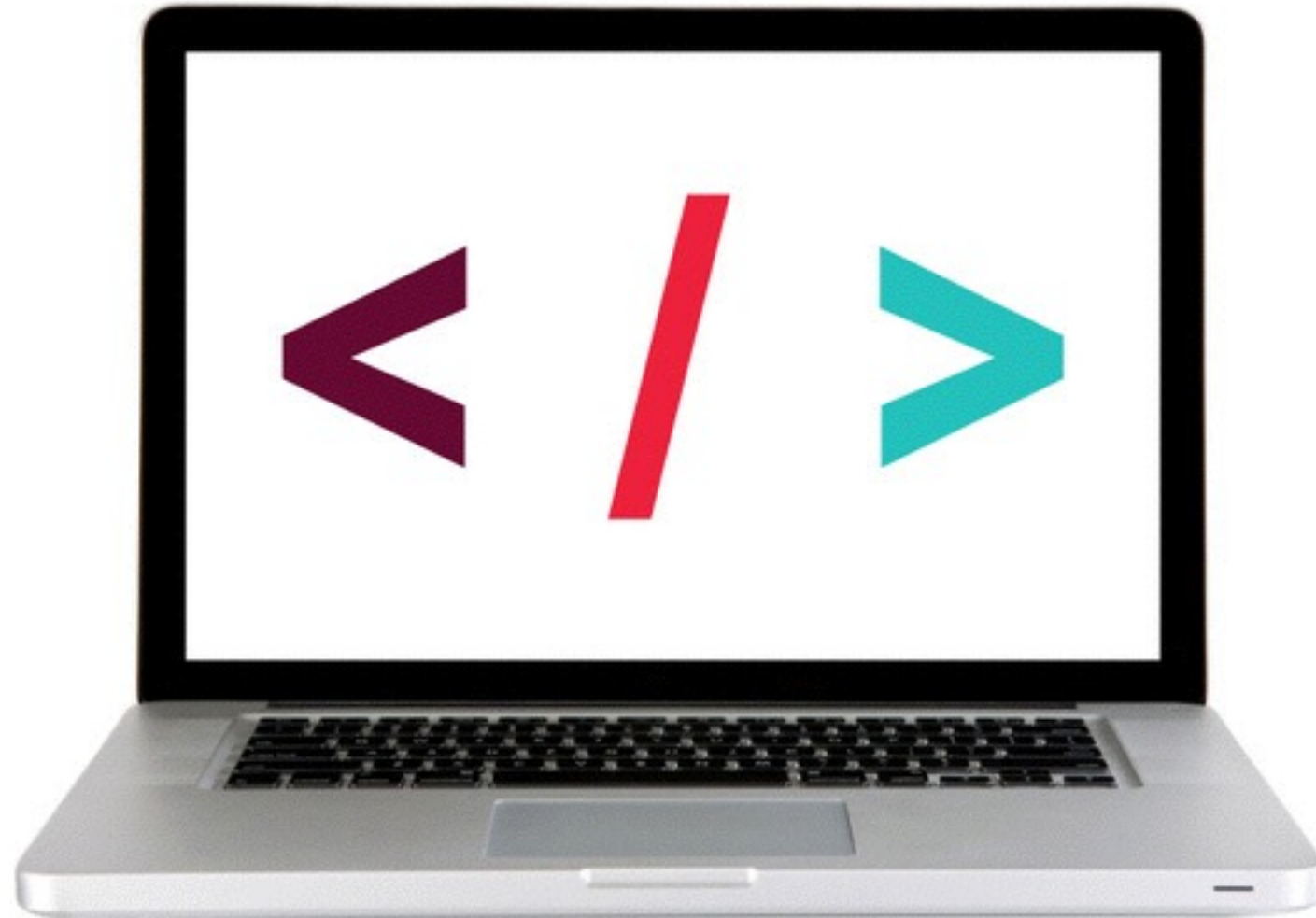


`rgba(0, 0, 0, 0.5)`



`rgba(0, 0, 0, 0.25)`

LET'S TAKE A CLOSER LOOK



ACTIVITY



EXERCISE

KEY OBJECTIVE

- ▶ Use CSS to add basic styles to an HTML page.

TIMING

3 min

1. (together) Add a link to the style.css file
2. Open starter_code > css_practice > index.html
3. Follow the instructions under Part 1

INTRO TO CSS

TYPE

TYPEFACE TERMINOLOGY — PART 1

PROPERTY	VALUES	EXAMPLE
text-align	left, center, right, justify	text-align: center;
text-transform	UPPERCASE, lowercase, Capitalize	text-transform: uppercase;
text-decoration	none, <u>underline</u>	text-decoration: underline;
line-height	number, px value	line-height: 22px;

TYPEFACE TERMINOLOGY — PART 2

PROPERTY	VALUES	EXAMPLE
font-weight	normal, bold	font-weight: bold;
font-style	regular, italic	font-style: italic;
font-family	sans-serif, serif	font-family: serif;
font-size	px value	font-size: 20px;

FONT-FAMILY

- If we want to use a specific system font, the user will [need to have it installed on their computer](#) for the font to show up.
- We can provide a comma-separated list with our preferred font-family, followed by "fallback" fonts.
- We usually want to end this list with either 'serif' or 'sans-serif'.

```
h1 {  
  font-family: Arial, Verdana, sans-serif;  
}
```

SOME COMMON SYSTEM FONTS:

- | | |
|-------------------|-------------|
| ‣ Georgia | ‣ Arial |
| ‣ Times | ‣ Verdana |
| ‣ Times New Roman | ‣ Helvetica |

**We'll take a look at how we can extend our font options by using a web font next week*

ACTIVITY



EXERCISE

KEY OBJECTIVE

- ▶ Use CSS to add basic styles to an HTML page.

TIMING

3 min

1. Refer back to `starter_code > css_practice > index.html`
2. Follow the instructions under Part 2

HTML BASICS

CASCADING STYLE SHEETS

HOW CSS RULES CASCADE

- Cascading Style Sheets
- Cascade: CSS rules are able to override one another and cancel each other out, depending on their order. In other words, the rules are able to cascade downward until they are canceled out by another rule.

LAST RULE

- If the two selectors are identical, the latter will take precedence

INHERITANCE



- Inheritance in CSS is how certain properties are passed on from a parent element down to its children
- If you specify the font-family or color properties on the <body> element, they will apply child elements. This is because the font-family property is **inherited** by child elements.
- Not all properties are inherited. For example, it wouldn't make sense for the border to be inherited since it's unlikely that a child element should need the same border as its parent.

ACTIVITY



EXERCISE

KEY OBJECTIVE

- Practice using CSS by styling Marty McFly's Resume page

TIMING

- Until 9:20*
1. Review supplied .pngs in marty_mcfly folder
 2. (together) project set up
 3. Style Marty's About Me and Resume pages

LEARNING OBJECTIVES

- Apply relative paths to `` and `<a>` tags.
- Differentiate between basic web color principles: RGB, RGBA, hex.
- Use CSS to add basic styles to an HTML page.

WEEKLY OVERVIEW

WEEK 1

HTML Basics

WEEK 2

CSS Basics / CSS Box Model

WEEK 3

CSS Layout / Lab

CSS BASICS

HOMEWORK

HOMework

- ▶ Read through the homework specs on the FEWD 47 Dashboard

HIGHLY RECOMMENDED COURSE MATERIALS:

- ▶ [Jon Duckett - Web Design with HTML, CSS, JavaScript and jQuery Set](#)



SNACK CALENDAR

[SIGN UP HERE](#)

EXIT TICKETS

Bookmark this link!

[Fill it out here](#)