



- 1) SIGN UP FOR GITHUB**
- 2) GIVE USERNAME TO ERIC**
- 3) HAVE SLACK OPEN**

**TODAY!**

**KIMBERLY BAIRD & KATIE ALLYN**

*“What is the end goal of the class?”*

*“Is there a special way to organize HTML using tabs/spaces or does it vary by developer?”*

*“What are all of the HTML tags?”*

<https://developer.mozilla.org/en-US/docs/Web/HTML/Element>

*“On a link, how do you open the page in a new tab?”*

*“How do front-end frameworks interact with embedding from back-end frameworks without conflicting?”*

*“What should I do with Github now?”*



*“Sublime wizardry! (Namely, typing on multiple lines at once)”*

*“Are we going to talk about frameworks like angular or ember?”*

*“When did the DOCTYPE! become simplified?”*

*“How do I use the <img> tag?”*

*“What does the ‘design inspiration’ activity entail?”*

# CSS BASICS

*Eric Boyer*

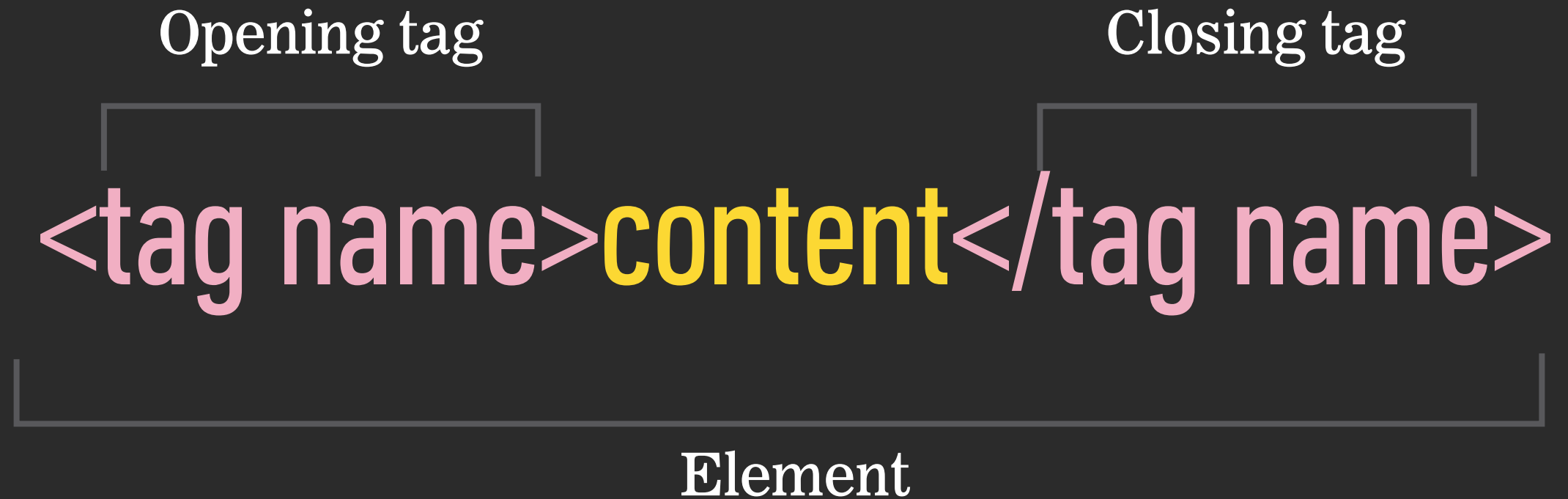
---

**FEWD**

---

**REVIEW**

## HTML SYNTAX — TAGS





---

## HTML SYNTAX — ATTRIBUTES

---

Attribute  
Name

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

Attribute  
Value

# ACTIVITY

---



## EXERCISE

### KEY OBJECTIVE

---

- ▶ Review HTML Tags

### TIMING

---

*15 min*

1. Look at review.png (in starter code folder)
2. Add markup to the text in starter\_code > [0] - HTML Review Activity > index.html
3. Google and implement <nav> <img> and <hr> tags.

---

## NAVIGATION AND HR

---

- ▶ Used to contain a set of navigational links.
- ▶ Used to wrap **major navigational blocks**

`<nav> </nav>`

- ▶ Used to create a horizontal rule, or line, across the page

`<hr>`

---

# HTML ENTITY CODES

---

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

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



**GRAB AN INDEX CARD  
AND DRAW YOUR FAMILY  
TREE ON ONE SIDE**

# LEARNING OBJECTIVES

- Describe the DOM and draw a simple DOM tree.
- Predict image paths and apply relative paths to `<img>` and `<a>` tags.
- Differentiate between basic web color principles: RGB, RGBA, hexadecimal color.
- Use CSS to add basic styles to an HTML page.

---

# AGENDA

---



- More HTML Basics
- Intro to CSS
- Lab

---

**FEWD**

---

# MORE HTML BASICS



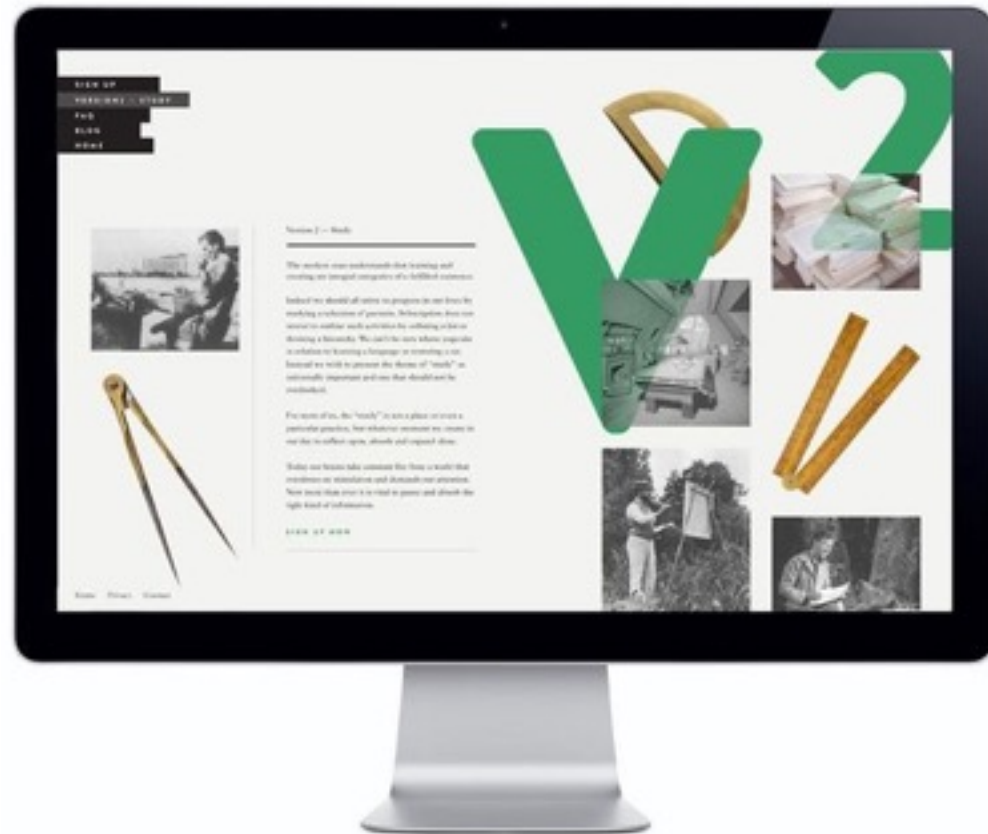
---

**MORE HTML BASICS**

---

# HTML STRUCTURE

# WHAT IS THE DOM?



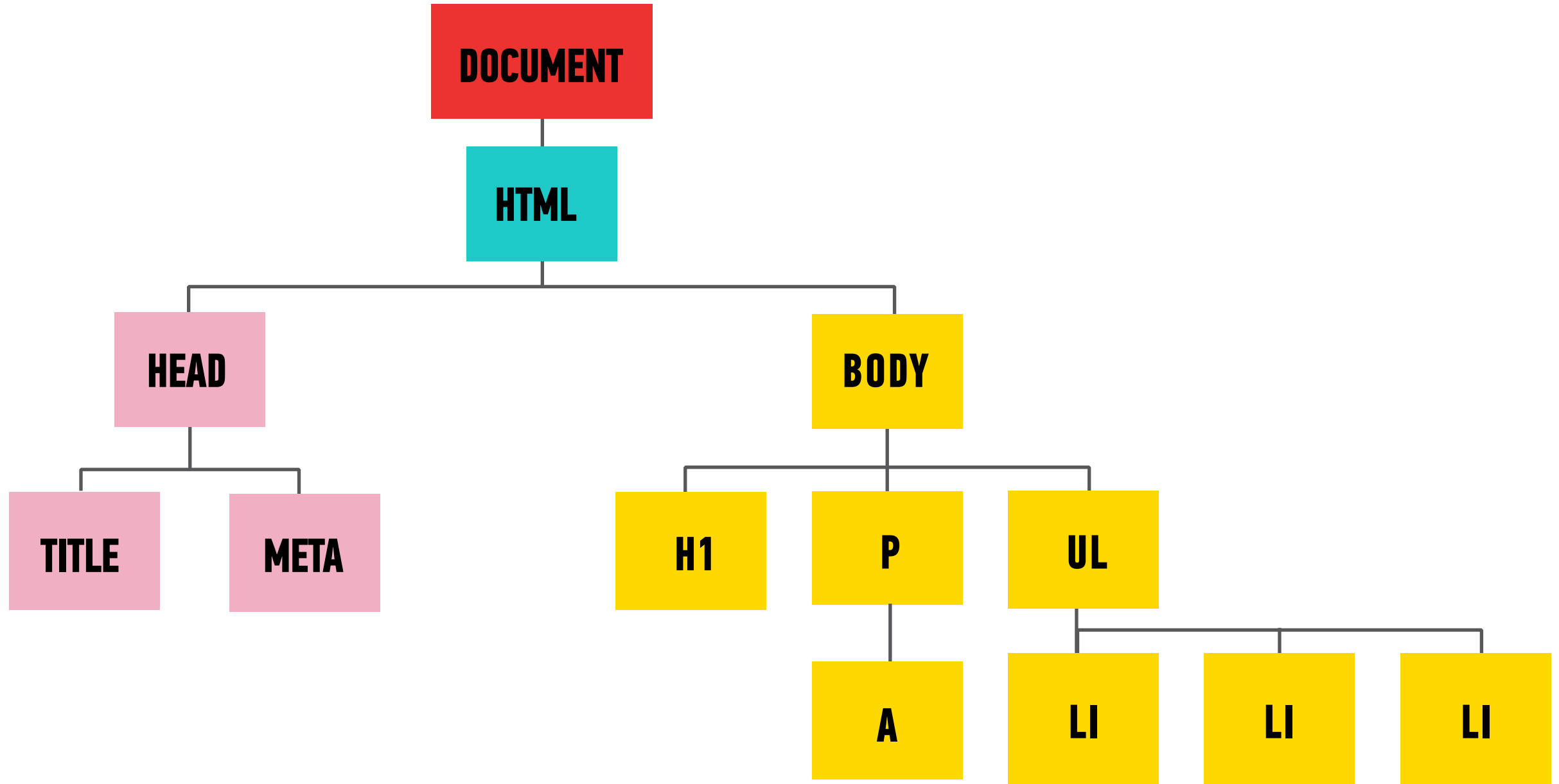
---

## WHAT IS THE DOM?

---



# DOM TREE

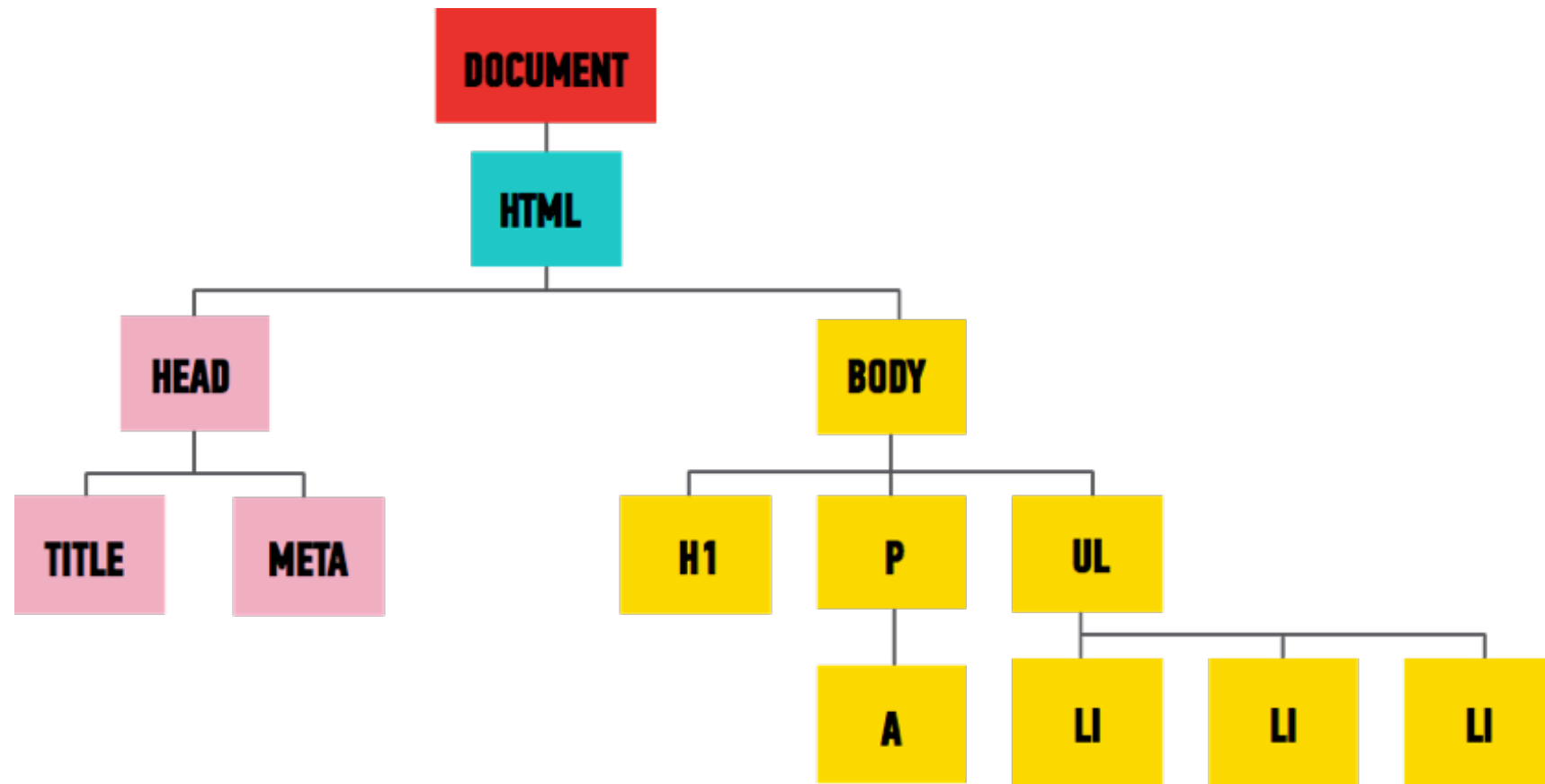


---

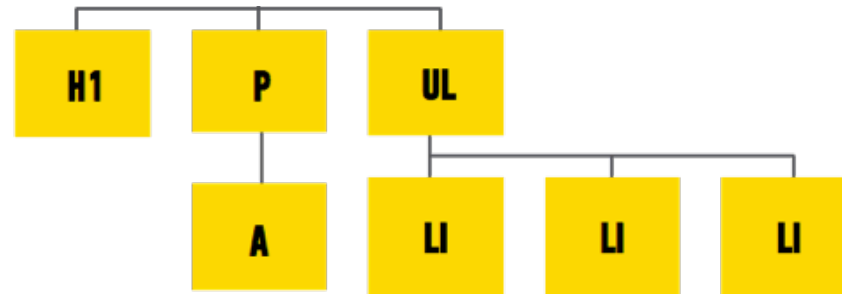
## DOM TREE

---

- The Document Object Model (DOM) is the model that **describes how all elements in an HTML page, like headers, images, paragraphs etc., are related to the topmost structure: the document itself.**



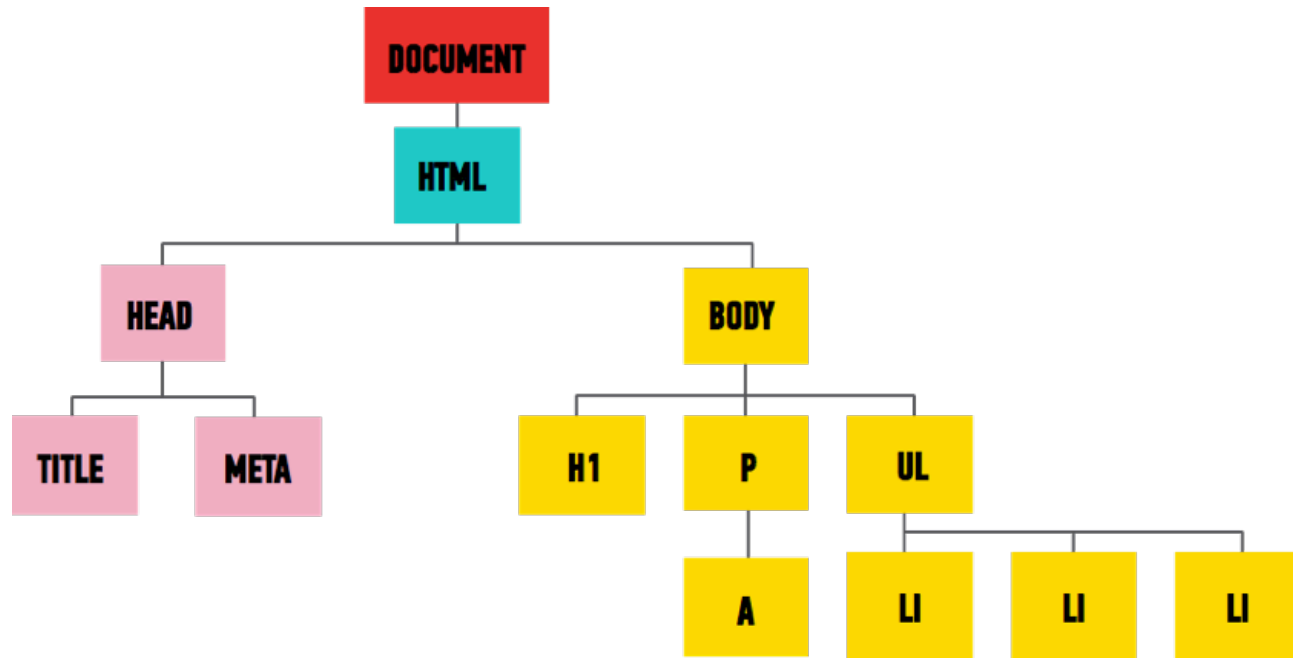
# DOM TREE



```
<h1>Site title</h1>
<p>Bacon ipsum dolor amet brisket tail frankfurter cupim pig salami.
Biltong beef ribs ribeye short loin flank corned beef tri-tip beef ball tip
sausage jowl swine. Fatback porchetta strip steak doner chicken <a href="
http://www.jamieoliver.com/recipes/pork-recipes/pork-belly-roast/">pork
belly</a>. Rump picanha jowl ground round, pancetta doner boudin landjaeger
prosciutto meatloaf.</p>
<ul>
  <li>Bacon</li>
  <li>Chicken</li>
  <li>Meatloaf</li>
</ul>
```



# HTML STRUCTURE



```
index.html
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="UTF-8">
5     <title>Document</title>
6   </head>
7   <body>
8
9     <h1>Site title</h1>
10    <p>Bacon ipsum dolor amet brisket tail
    frankfurter cupim pig salami. Fatback
    porchetta strip steak doner chicken <a href="
    http://www.jamieoliver.com/recipes/pork-recipes
    /pork-belly-roast/">pork belly</a></p>
11    <ul>
12      <li>Bacon</li>
13      <li>Chicken</li>
14      <li>Meatloaf</li>
15    </ul>
16
17  </body>
18 </html>
```

# ACTIVITY

---



## EXERCISE

### KEY OBJECTIVE

---

- ▶ Draw out a simple DOM tree

### TIMING

---

*2 min*

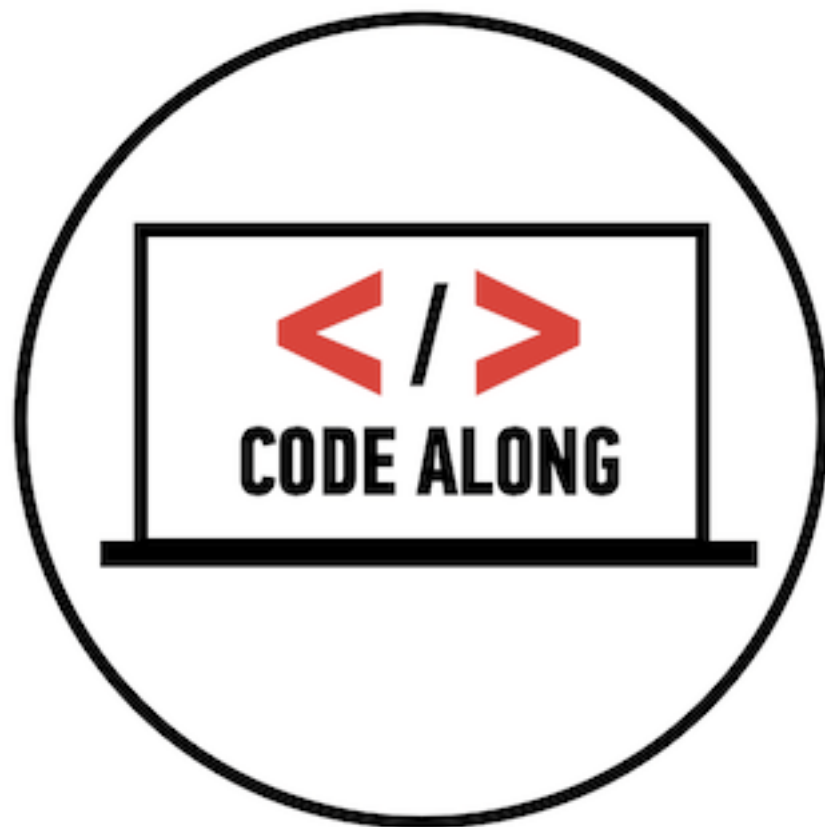
1. Review starter\_code > [1] - DOM Tree > index.html
2. On the back of your index card, draw out a simple DOM tree for the HTML



---

## CODE ALONG

---



---

**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

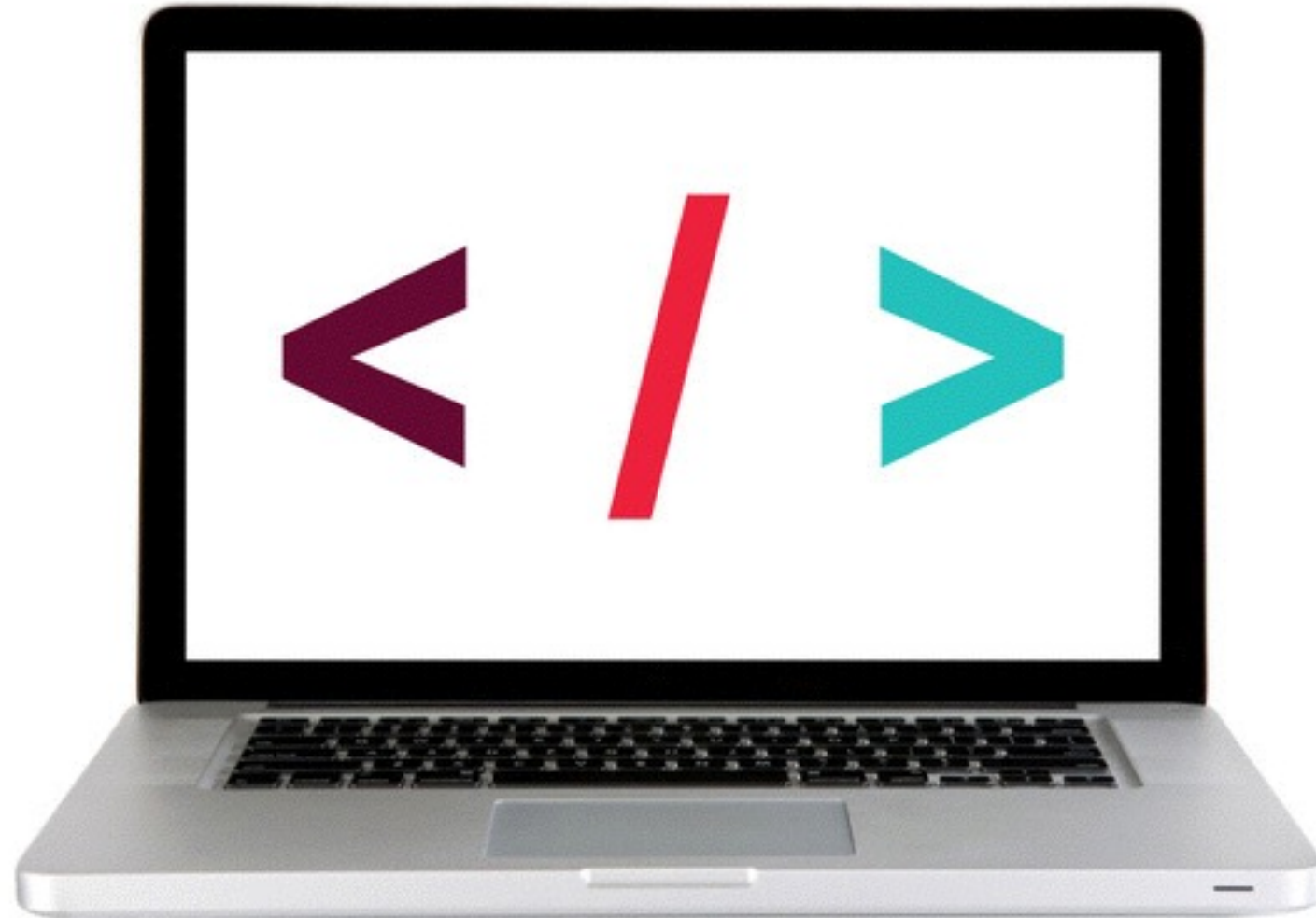
```

```

---

**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

`<a href="about.html">About Page</a>`

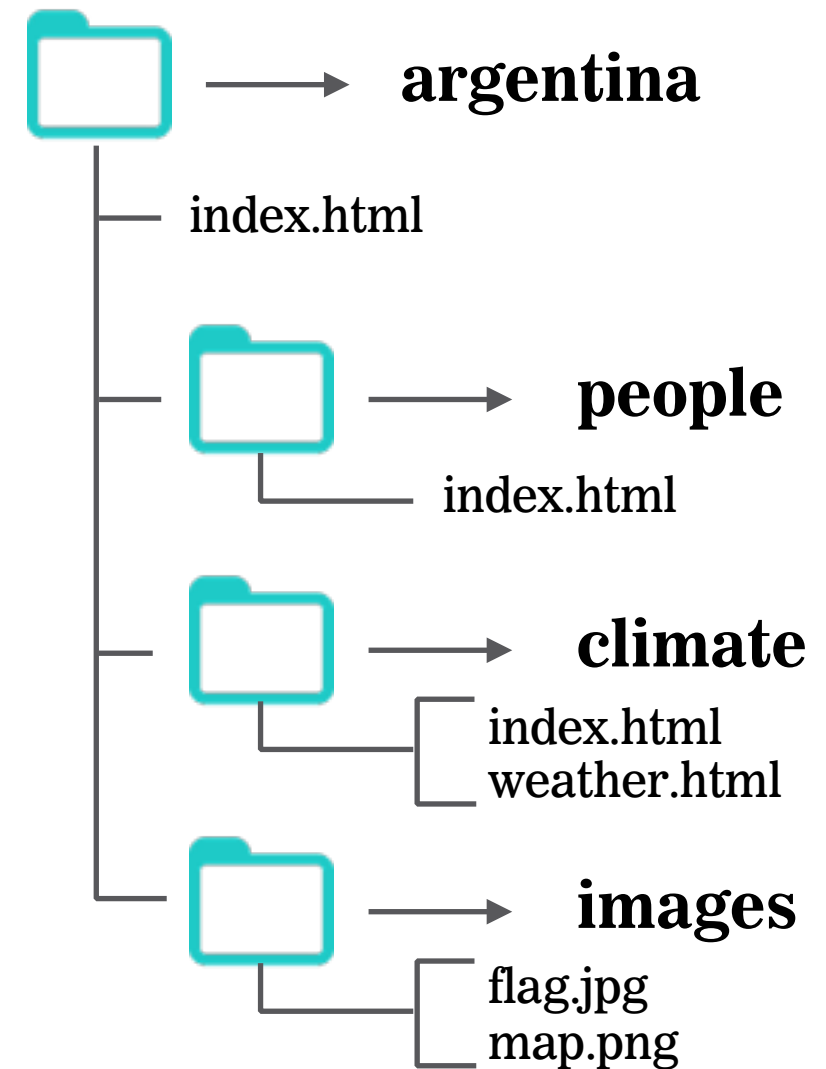
``

---

## DIRECTORY STRUCTURE

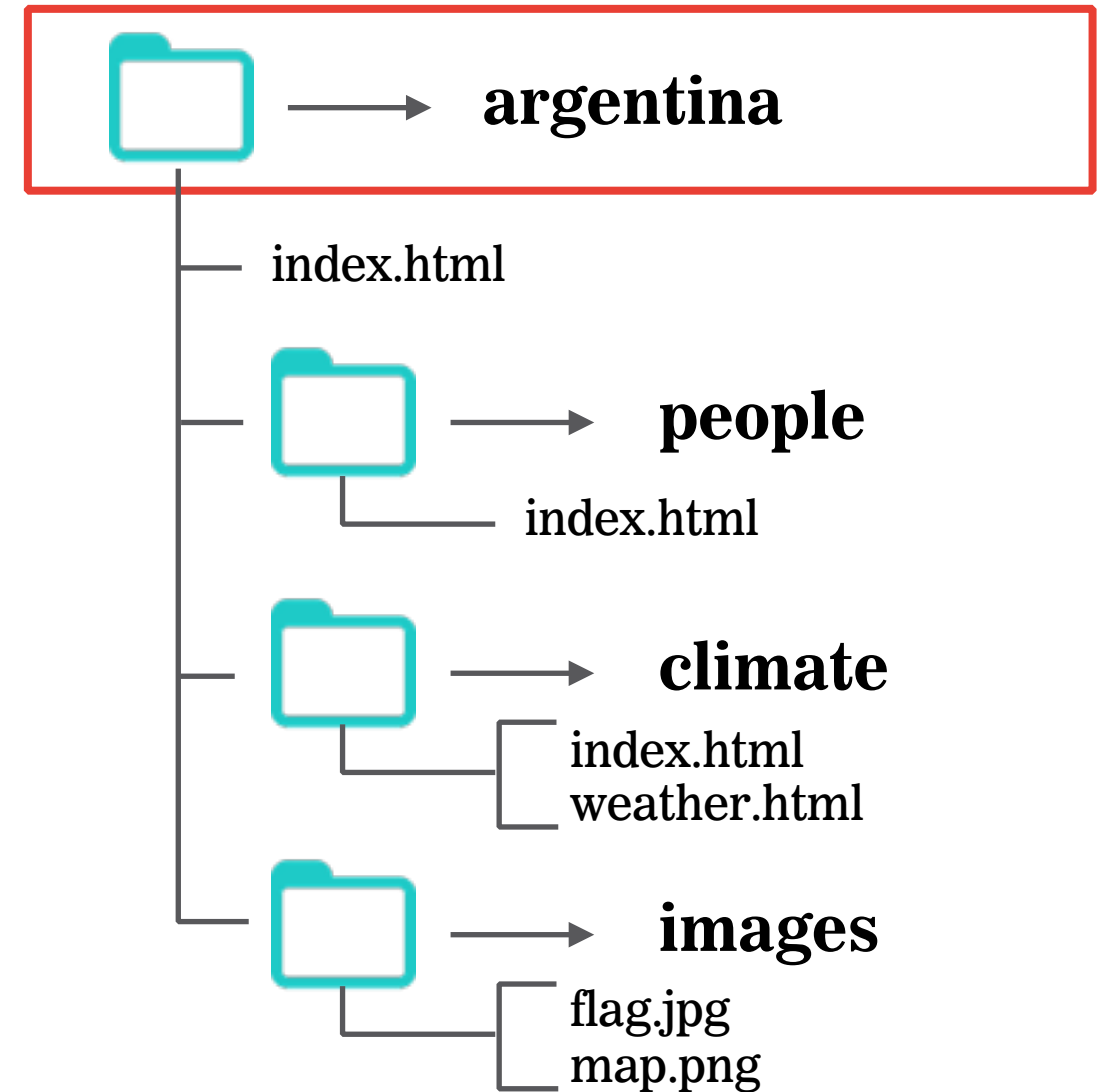
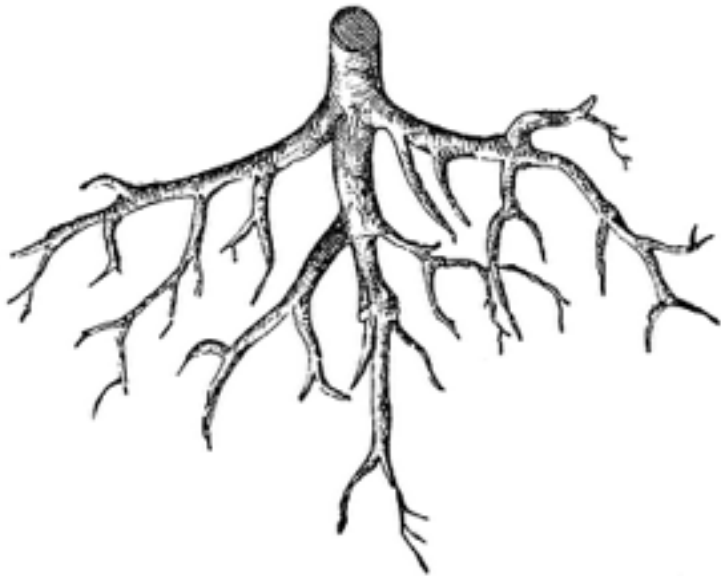
---

- 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

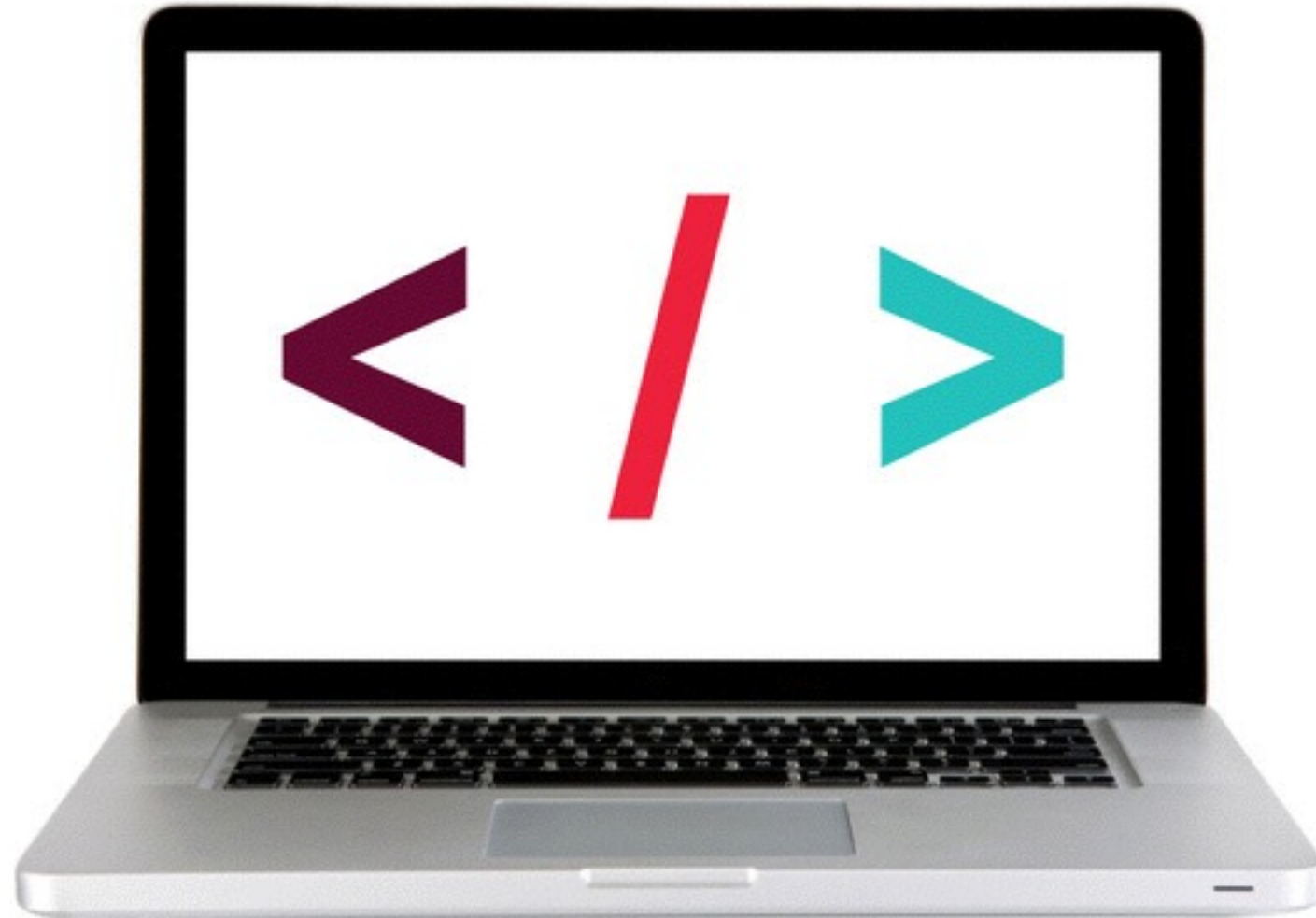
- The topmost folder is called the **root** folder. It contains all other files and folders for a website.



---

## LET'S TAKE A CLOSER LOOK

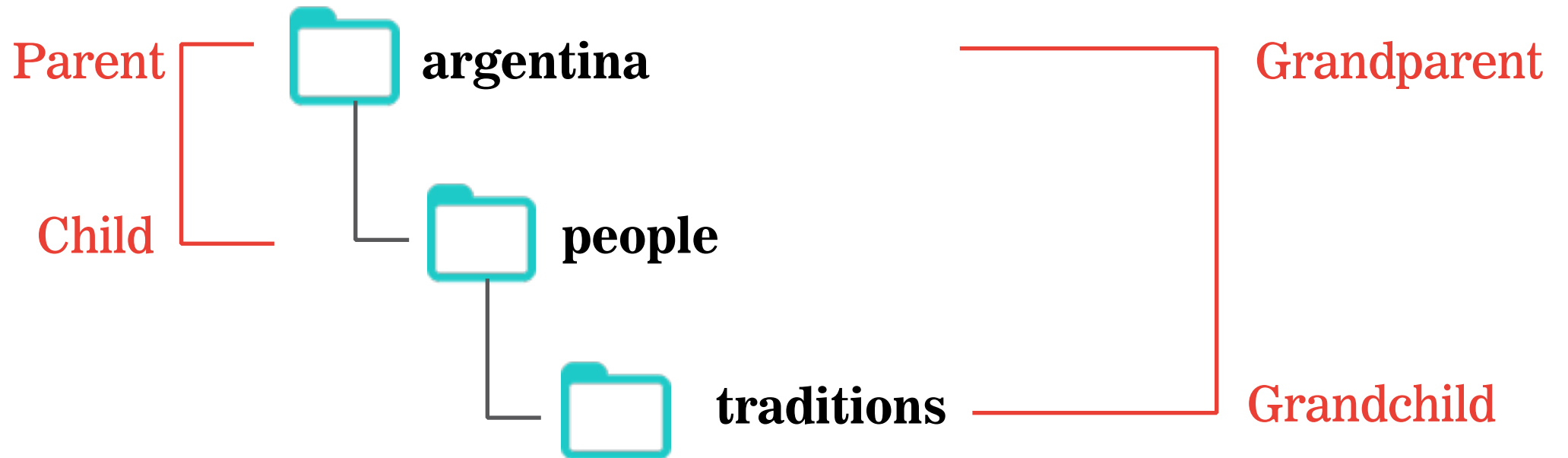
---



## DIRECTORY STRUCTURE

---

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



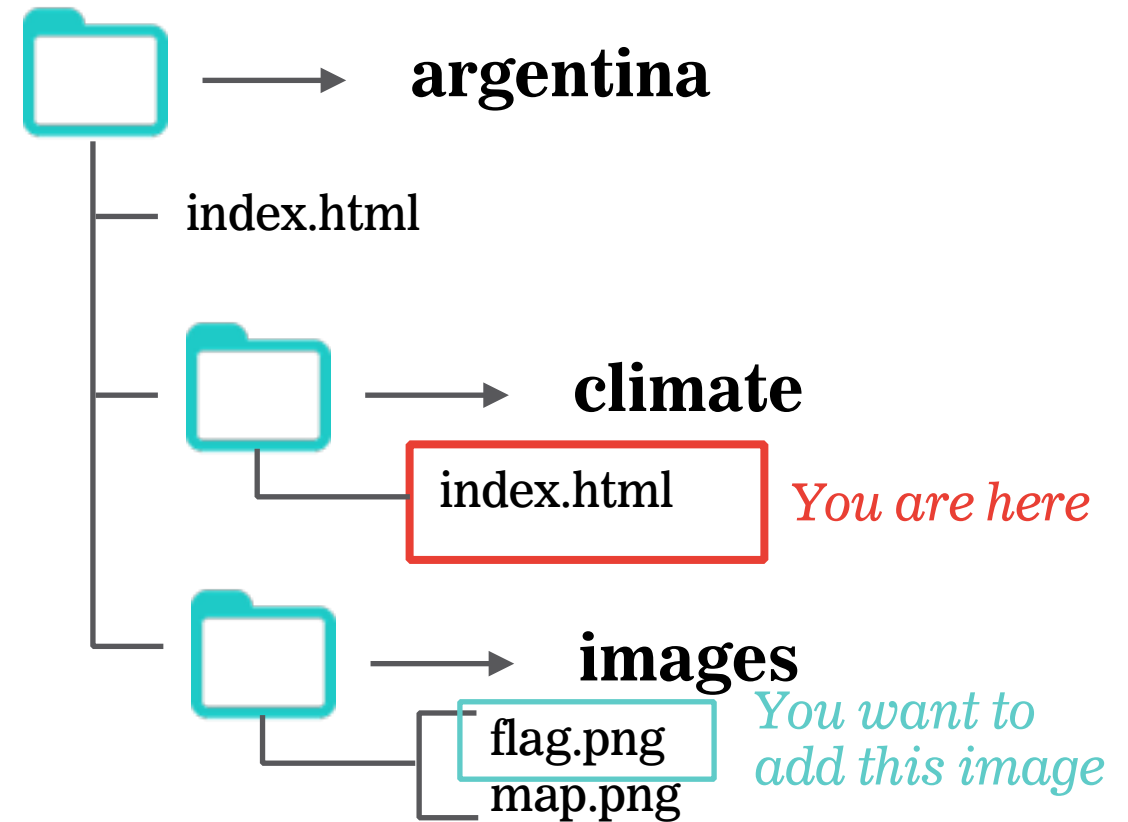
# RELATIVE URLS

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.

## RELATIVE URLS

1. First we use ../ to go back to the parent directory
2. Then we add the path to the image



```
img src="../../images/flag.png" alt="Flag">
```



---

# ACTIVITY

---



## EXERCISE

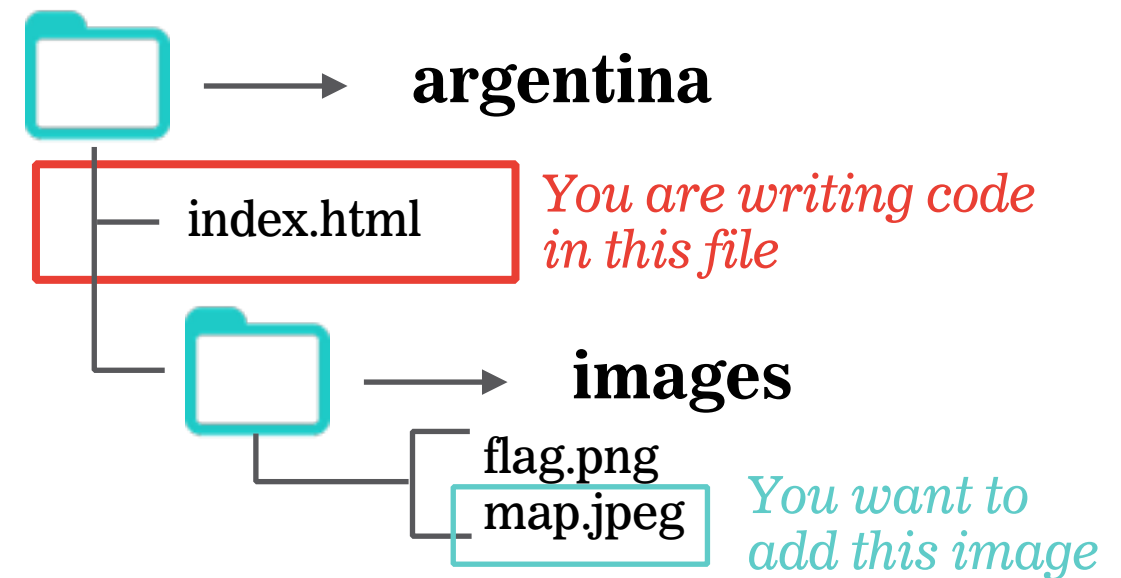
### **KEY OBJECTIVE**

---

- Practice writing absolute and relative paths

## ACTIVITY

- ▶ We want to add the image map.jpeg to our homepage (index.html)
- ▶ Write out the image tag (include src and alt attributes)



**Answer: ``**

	DESCRIPTION	PATH
CHILD	Name of child folder + / + file name	images/map.jpeg

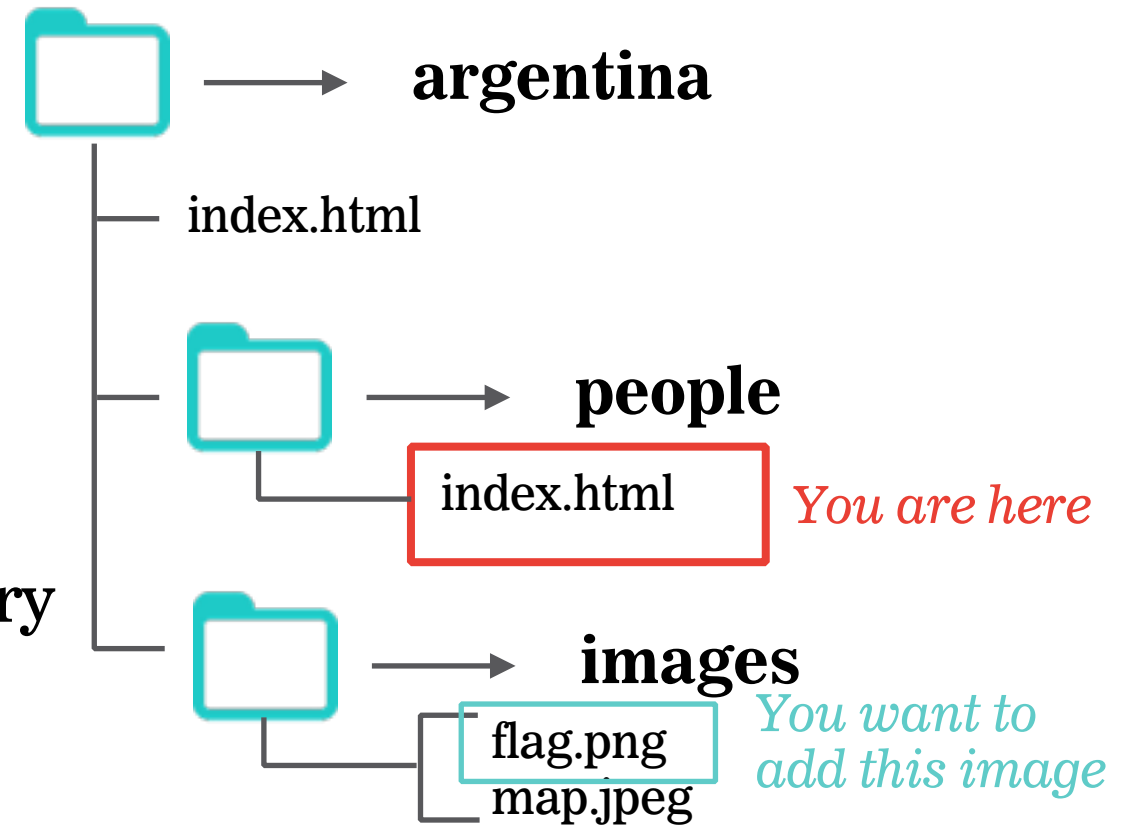
## ACTIVITY

- ▶ We want to add flag.png to the homepage of the "people" section of our site.
- ▶ Write out the image tag (include src and alt attributes)

### HINT:

1. First we use ../ to go back to the parent directory
2. Then we add a path to the image

**`img src="../../images/flag.png" alt="Flag">`**



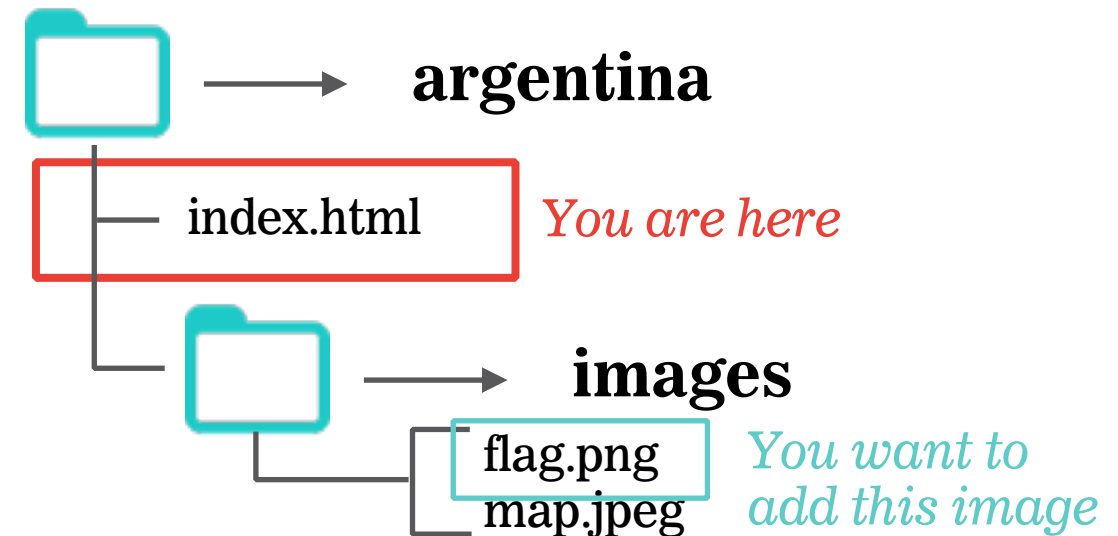
---

## ACTIVITY

---

- ▶ We want to add the image flag.png to our homepage (index.html)
- ▶ Write out the image tag (include src and alt attributes)

****



---

## ACTIVITY

---

- ▶ We want to add a link from our homepage (index.html) to our about page (about.html)
- ▶ Write out the anchor tag (including href attribute)



**`<a href="about.html">About Page</a>`**

---

**HTML BASICS**

---

# INTRO TO 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.



plain ol' HTML



HTML + CSS

---

## CSS SYNTAX

---

Selector



The diagram illustrates the components of a CSS rule. A bracket above the text 'h1' identifies it as the 'Selector'. A bracket below the text 'color: yellow;' identifies it as the 'Declaration'. The entire rule is enclosed in curly braces.

```
h1 { color: yellow; }
```

Declaration

## CSS SYNTAX

---

h1, h2 {

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 — :)

---

- Use the `<link>` element to tell the browser where to find the CSS file that should be used to style the page.
- This should live inside the `<head>` of the document.

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

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

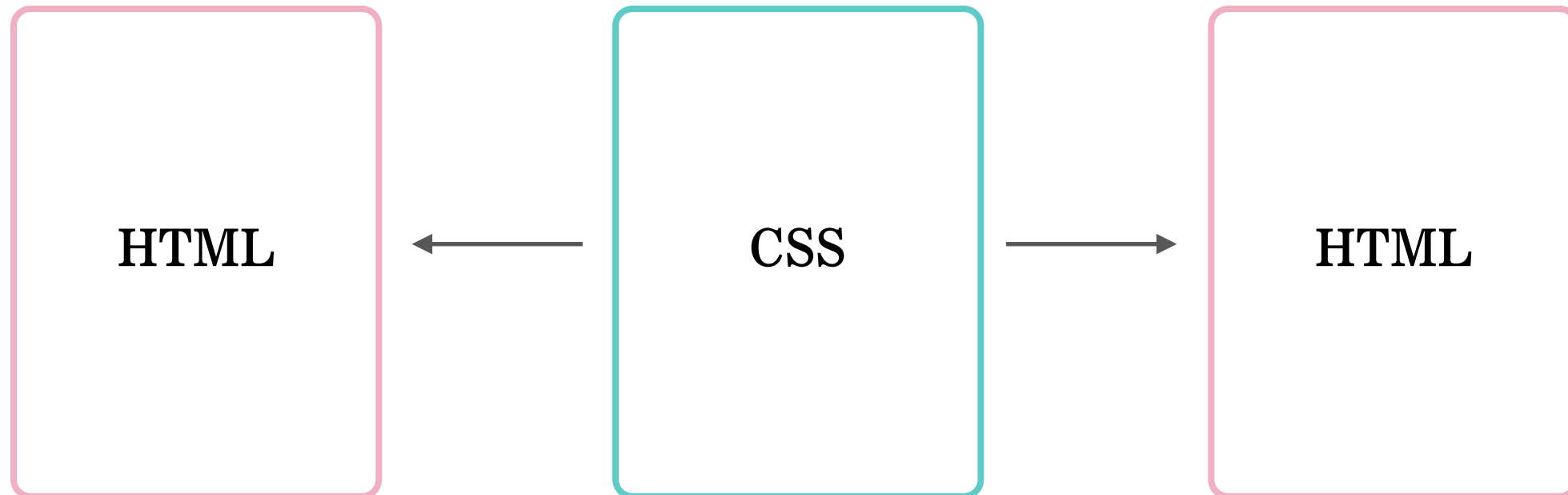
---

## INTERNAL VS. EXTERNAL CSS

---

### **BENEFITS OF USING AN EXTERNAL STYLE SHEET:**

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



---

**INTRO TO CSS**

---

**COLOR**

---

## COLOR

---

### TEXT COLOR:

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

### BACKGROUND COLOR:

#### 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: blue;  
}
```

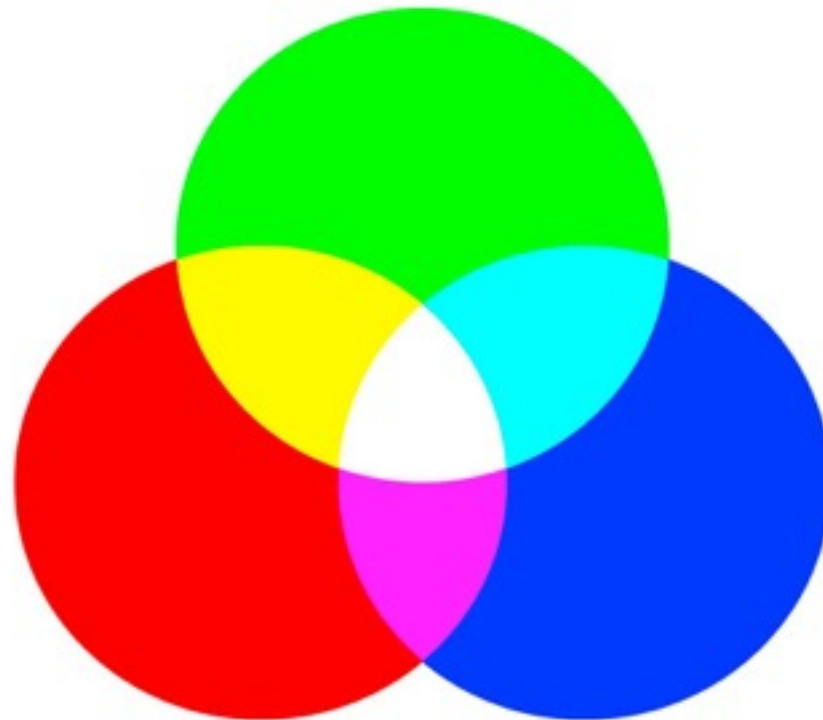


---

## COLOR

---

- Every color on a computer screen is created by mixing amounts of **red**, **green**, and **blue**
- To find the color you want, you can use a color picker
- Mac: [Sip](#) (free)
- Windows: [Just Color Pic](#) (free)
- Photoshop



---

## COLOR

---

### RGB VALUES

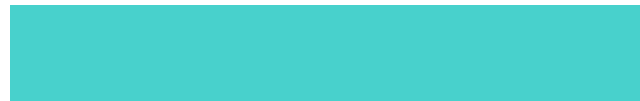
- Values for red, green and blue are expressed as numbers between 0 and 255



**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](#)

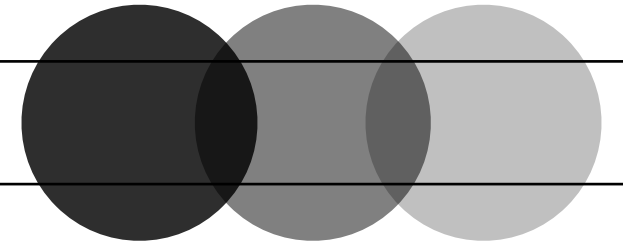


**MediumTurquoise**

---

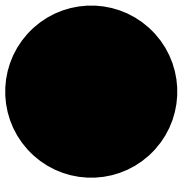
## OPACITY

---

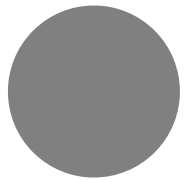


## RGBA

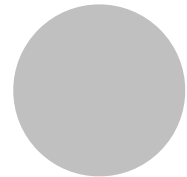
- ▶ RGBA works the same as RGB, except that it takes a 4th value called 'alpha'.
- ▶ This is a value between 0 and 1 which can be used to determine a color's opacity on the page.



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



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

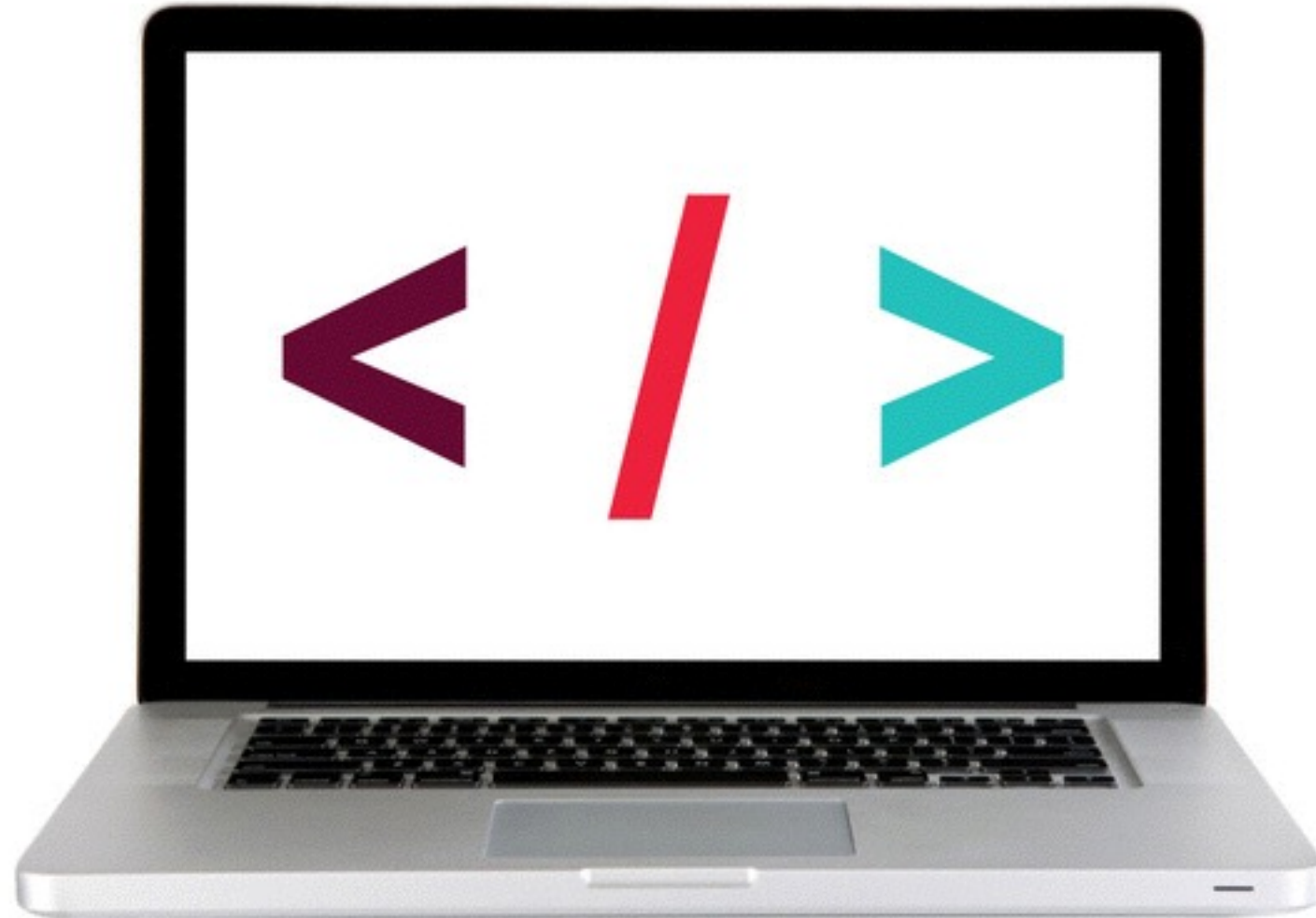


`rgba(0, 0, 0, 0.25)`

---

## LET'S TAKE A CLOSER LOOK

---



# ACTIVITY

---



## EXERCISE

### KEY OBJECTIVE

---

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

### TIMING

---

*3 min*

1. Open starter\_code > [2] - CSS Practice > index.html
2. (together) Add a link to the style.css file
3. Follow the instructions under Part 1

---

## INTRO TO CSS

---

# TYPE

---

# TYPEFACE TERMINOLOGY

---

## TEXT-ALIGN

left  
center  
right  
justify

## TEXT-TRANSFORM

UPPERCASE  
lowercase  
Capitalize

## TEXT-DECORATION

none  
underline

## LINE-HEIGHT

20px

## FONT-WEIGHT

Normal  
**Bold**

## FONT-STYLE

Regular  
*Italic*

## FONT-FAMILY

sans-serif  
serif

## FONT-SIZE

12px  
**26px**

---

## FONT-FAMILY

---

### serif

- Georgia
- Times
- Times New Roman

### sans-serif

- Arial
- Verdana
- Helvetica



# ACTIVITY

---



## EXERCISE

### KEY OBJECTIVE

---

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

### TIMING

---

*3 min*

1. Refer back to starter\_code > [2] - 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.
- You can force a lot of properties to inherit values from their parent elements by using 'inherit' for the value of the properties.

# ACTIVITY

---



## EXERCISE

### KEY OBJECTIVE

---

- ▶ Practice using CSS by styling Wendy Bite's Resume page

### TIMING

---

- Until 8:45*
1. Review supplied .pngs starting with Lab...
  2. (together) project set up
  3. Style Wendy's About Me and Resume pages

# LEARNING OBJECTIVES

- Describe the DOM and draw simple DOM tree.
- Predict image paths and apply relative paths to `<img>` and `<a>` tags.
- Differentiate between basic web color principles: RGB, RGBA, hexadecimal color.
- Use CSS to add basic styles to an HTML page.

---

**CSS BASICS**

---

# HOMEWORK

---

## **HOMEWORK**

---

- Continue working through the lab we started on today
- Make your own version the “About me” page!

### **HIGHLY RECOMMENDED COURSE MATERIALS:**

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





---

## **HOMEWORK**

---

# **FEEDBACK GROUPS**

---

## FEEDBACK GROUPS

---

### IF YOU ARE STUCK...

- 1) Use the “Chrome Inspector” to look at your code
  - 2) Ask your question in the Slack channel and see if any fellow students might know the answer
  - 3) Ask Eric & Adriana
- \* When using your fellow students and instructors, pushing your code to Github is a great place to share where all of your code is currently at.

---

## **FEEDBACK GROUPS**

---

### **GROUP 1 (ERIC)**

Kimberly Baird  
Tom Bunting  
Michael Fischer  
Jim Howes  
Jon Iler  
Peyton Lee  
Blanca Leon-Carter  
Anna Matras  
Christopher Zalek

**TO: EBOYER@GMAIL.COM**

### **GROUP 2 (ADRIANA)**

Corin Menuge  
Colin Quirk  
Allison Schaffer  
Kevin Sella  
Michael Sena  
Nicholas Skeba  
Ashley Sullivan  
Lisa Vasquez  
Katrina Wang  
Xi Yue Li

**TO: ADRIANA.LCS316@GMAIL.COM**

---

## FEEDBACK GROUPS

---

### **DUE 11:59PM, SUNDAY FEB 29TH**

- Continue working through the lab we started on today
- Make your own version the “About me” page!
- \* Familiarize yourself with the Github resources. Try to turn homework in and share code with the class using it.

\*Feedback given between Monday AM - Tuesday PM

**FRONT-END WEB DEVELOPMENT**

---

**SNACKS & DESIGN**

**THURSDAY FEB 25TH**

**CORIN MENUGE & PEYTON LEE**

**(GOOGLE SHEET IS PINNED IN SLACK)**

# EXIT TICKETS