LET'S GET EVERYTHING SET UP!

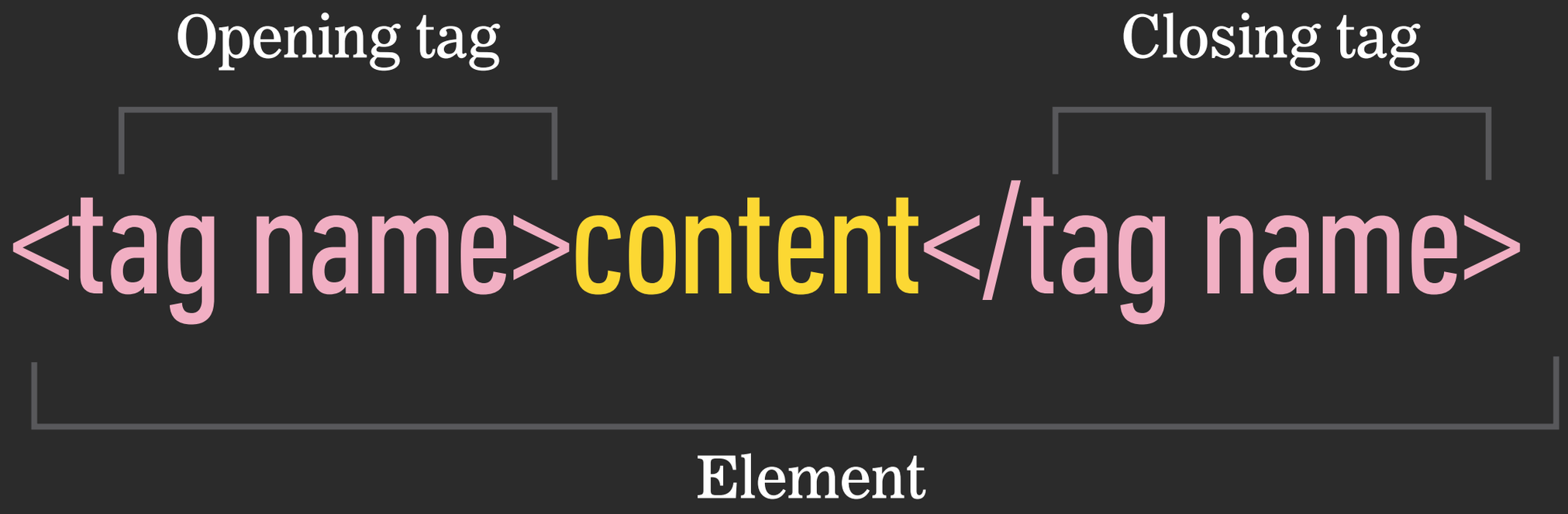
1. Open your Terminal
2. Go to your FEWD-52 folder: `cd ~/Desktop/fewd/FEWD-52`
3. Download the new material: `git pull`
4. Open the FEWD-52 folder in finder
5. Drag it to your Sublime Text icon to open it, also open your homeowork in Sublime and in Chrome
6. Also... Please sign the attendence sheet.

GENERAL ASSEMBLY

CSS BASICS

Alex White

HTML SYNTAX — TAGS



SEMANTIC MARKUP

SEMANTIC TAGS — STRONG, EM

ELEMENT	DESCRIPTION	EXAMPLE
strong	Strong Importance	Do not walk
em	Emphasis	I think Bill went.

- Default browser styles: **strong will be bold** and *em will be italic*
- indicates emphasis that may *subtly change* the meaning of a sentence:

I *think* John was there

I think *John* was there

HTML STRUCTURE

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>
```

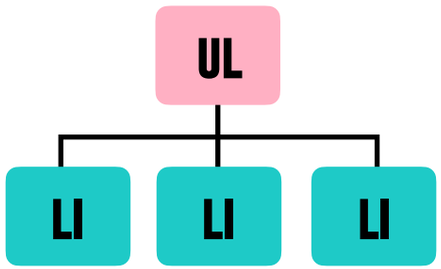
PAGE STRUCTURE — NESTING

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

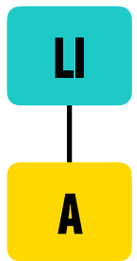
- Here we can say that our <ul> is the **parent** of our s
- We can also say that our s are **children** of the




```
<!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 <li> element.

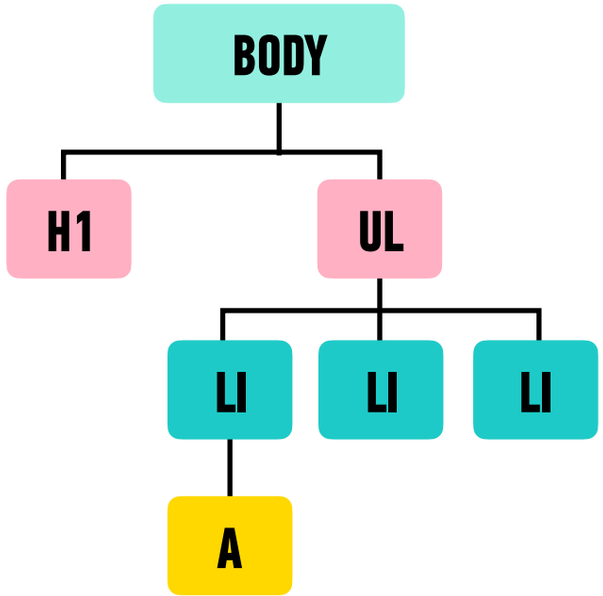



```
<!DOCTYPE html>
<html lang="en">
<head>
     <meta charset="UTF-8">
     <title>The Evolution of Denim</title>
</head>
<body>
    <h1>The Evolution of Denim</h1>
    <ul>
        <li>Dark Wash <a href="linkgoeshere">Jeans</a></li>
        <li>Stone Wash</li>
        <li>Chambray</li>
    </ul>
</body>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

<a>

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

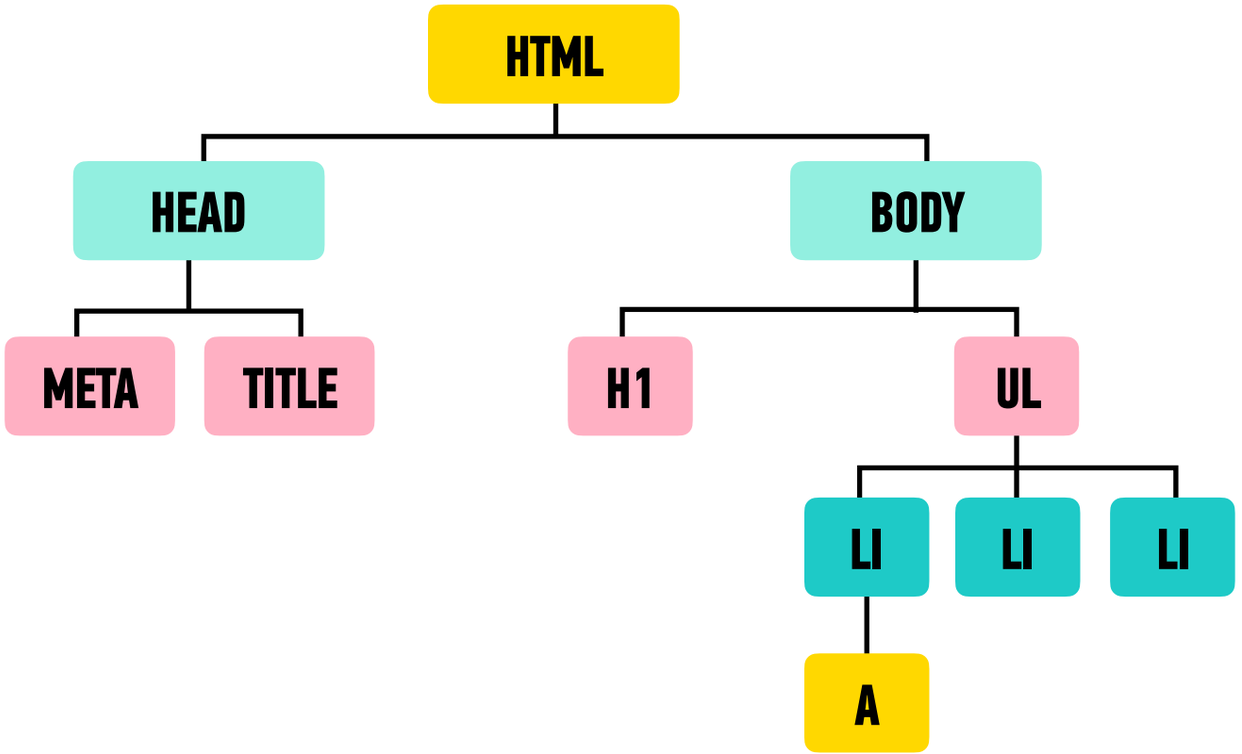


body

```
<!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




```html
<!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.

KEY OBJECTIVE

- 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?

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

```
<img src="images/camera.jpg" alt="Girl holding a camera" />
```

IMAGES

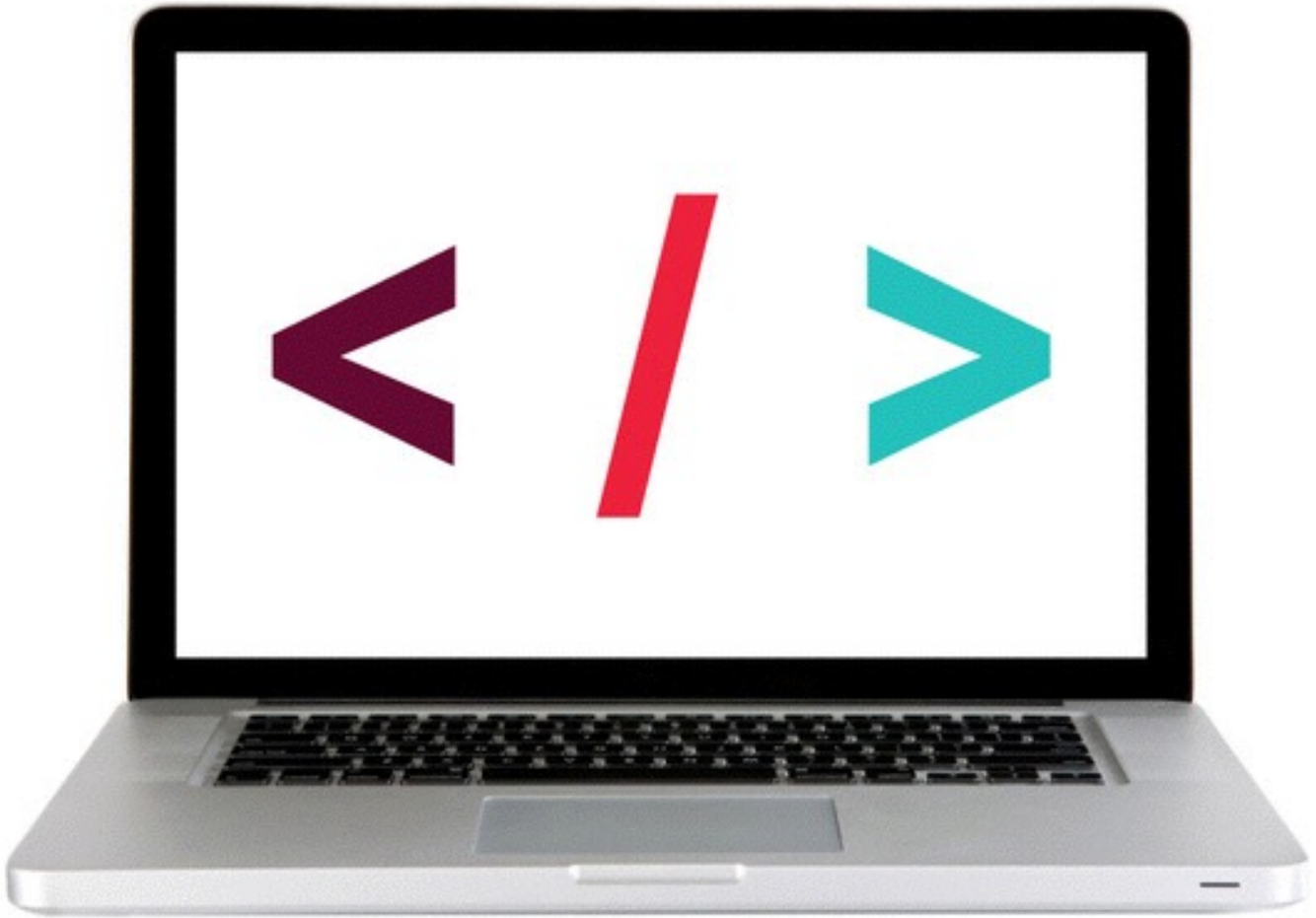
The **alt** attribute provides a text description of the image that:

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

Text description

```
<img src="images/camera.jpg" alt="Girl holding a camera" />
```

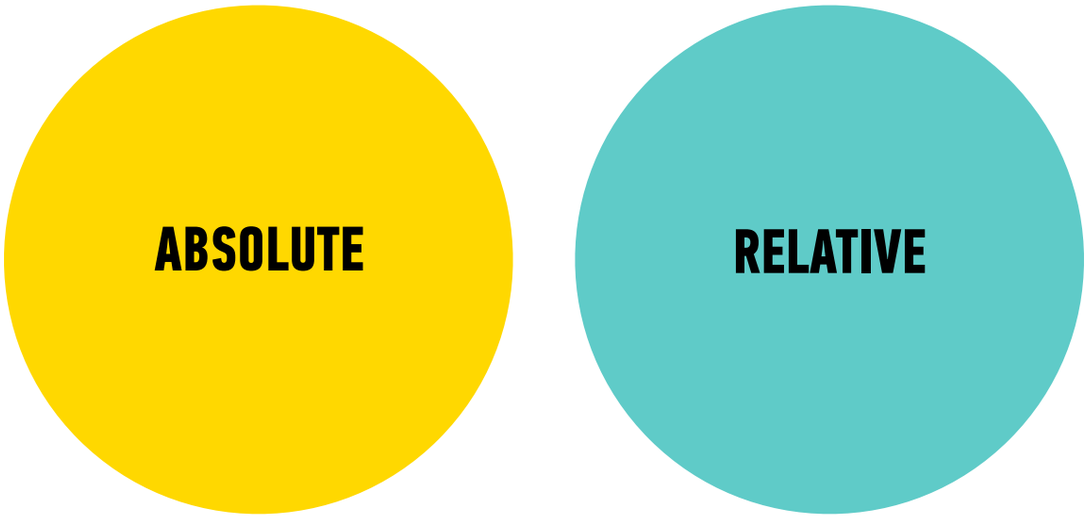
LET'S TAKE A CLOSER LOOK



URLS

TYPES OF URLS

- There are two main types of URLs:



LINKING TO OTHER SITES – ABSOLUTE URLS

ABSOLUTE

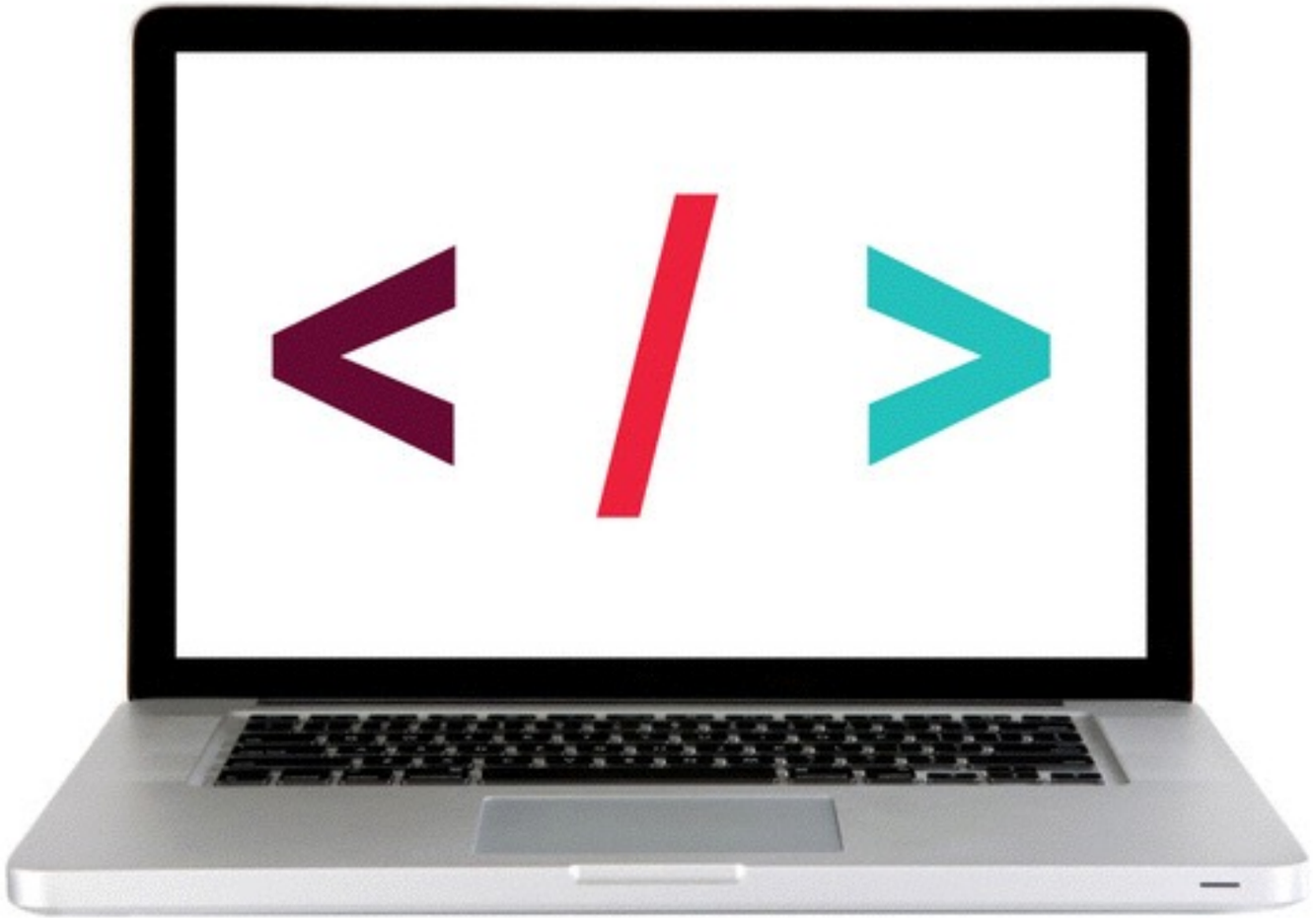
WHEN YOU LINK TO ANOTHER SITE:

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

Absolute URL

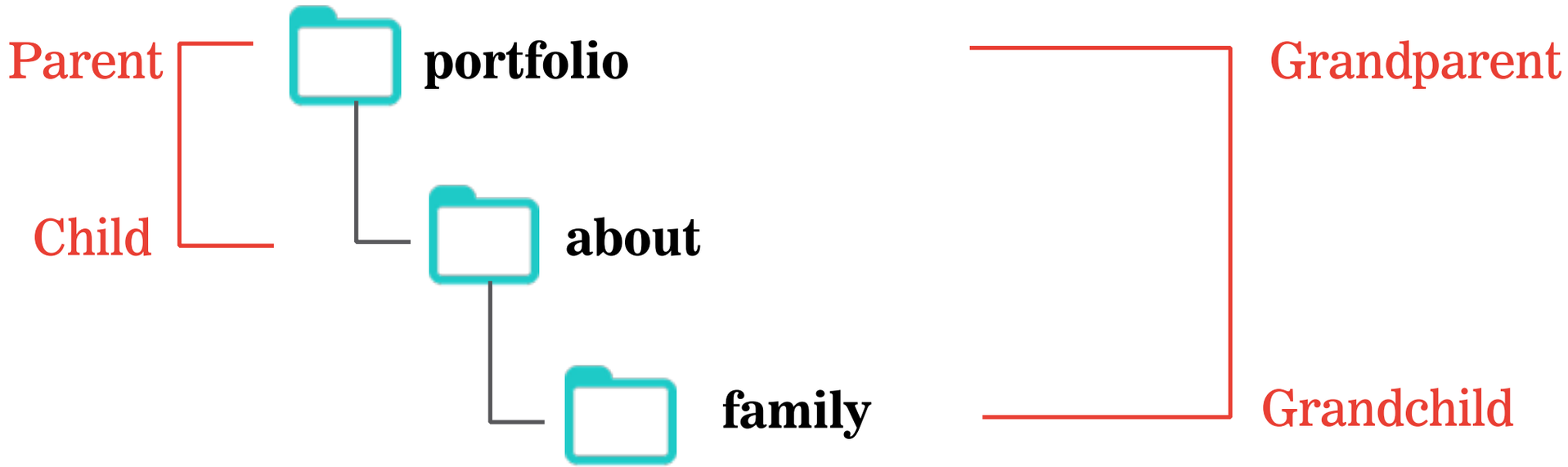
```
<a href="http://www.amazon.com">Amazon</a>

<img src="http://saraheholden.com/img/aboutPhoto.jpg" alt="Profile Pic>
```

LET'S TAKE A CLOSER LOOK

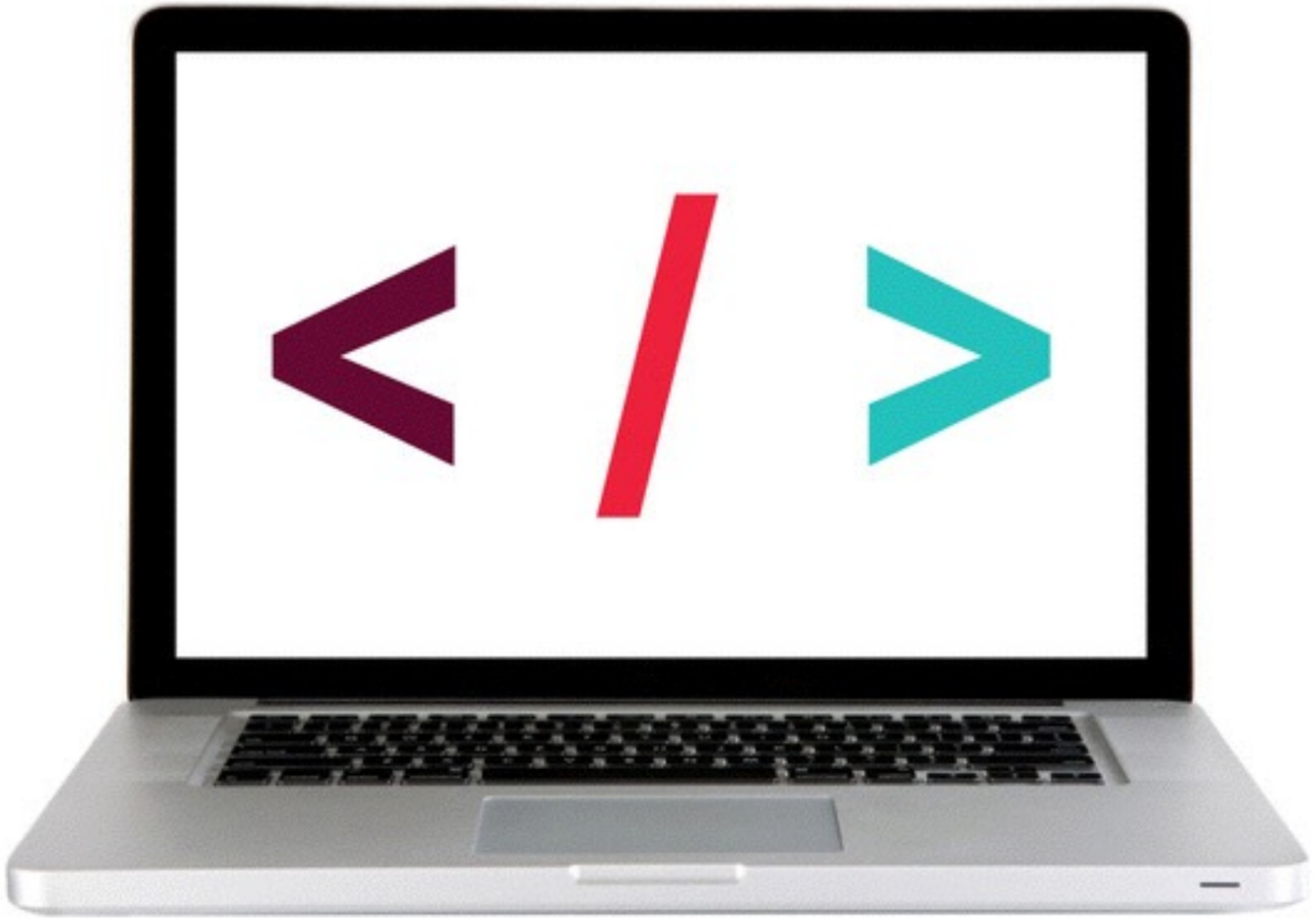


DIRECTORY STRUCTURE

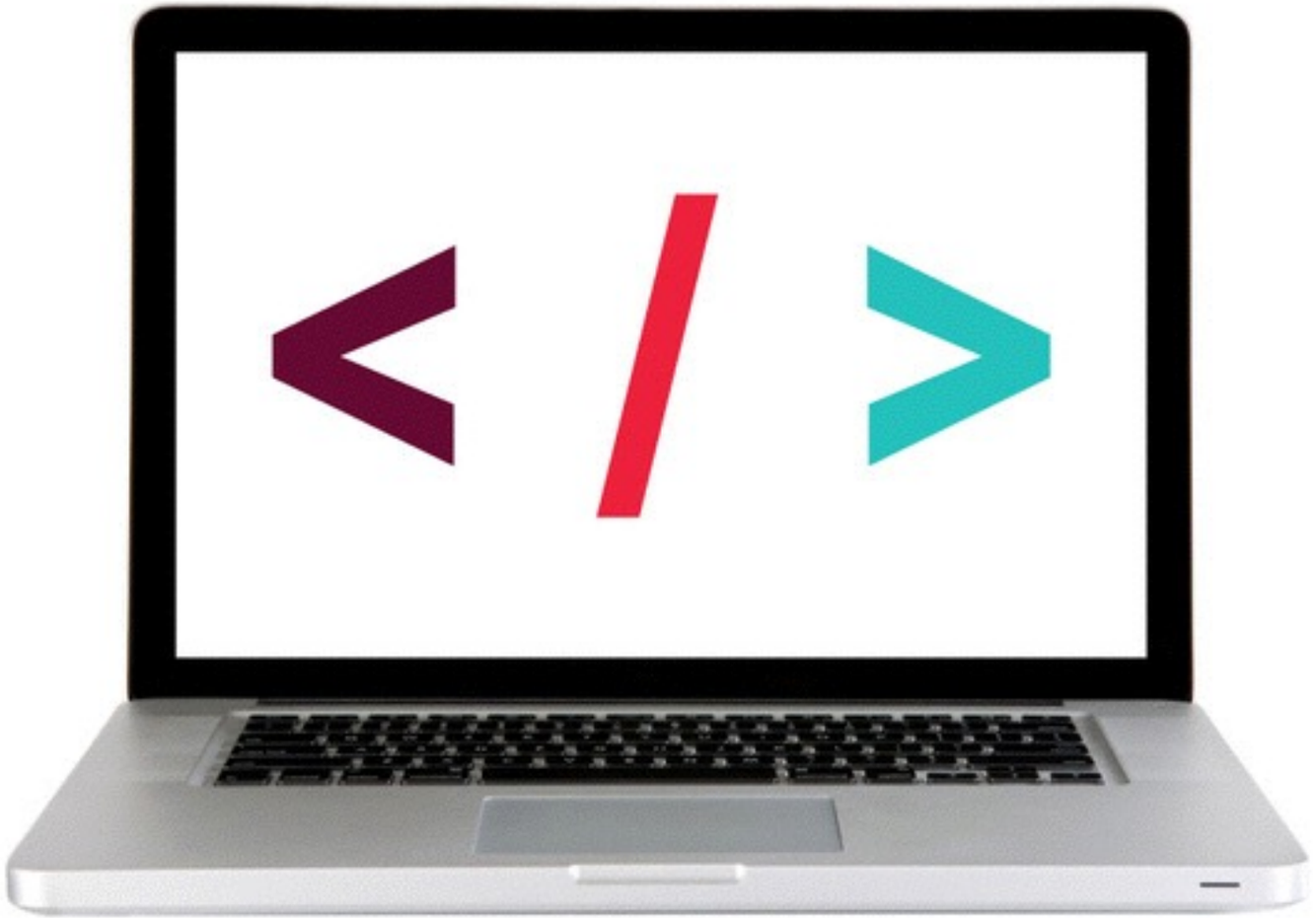
- Relationships between folders can be described using similar language to that of a family tree



LET'S TAKE A CLOSER LOOK



LET'S TAKE A CLOSER LOOK



ACTIVITY

EXERCISE

KEY OBJECTIVE

- Practice adding relative URLs to a project

KEY OBJECTIVE

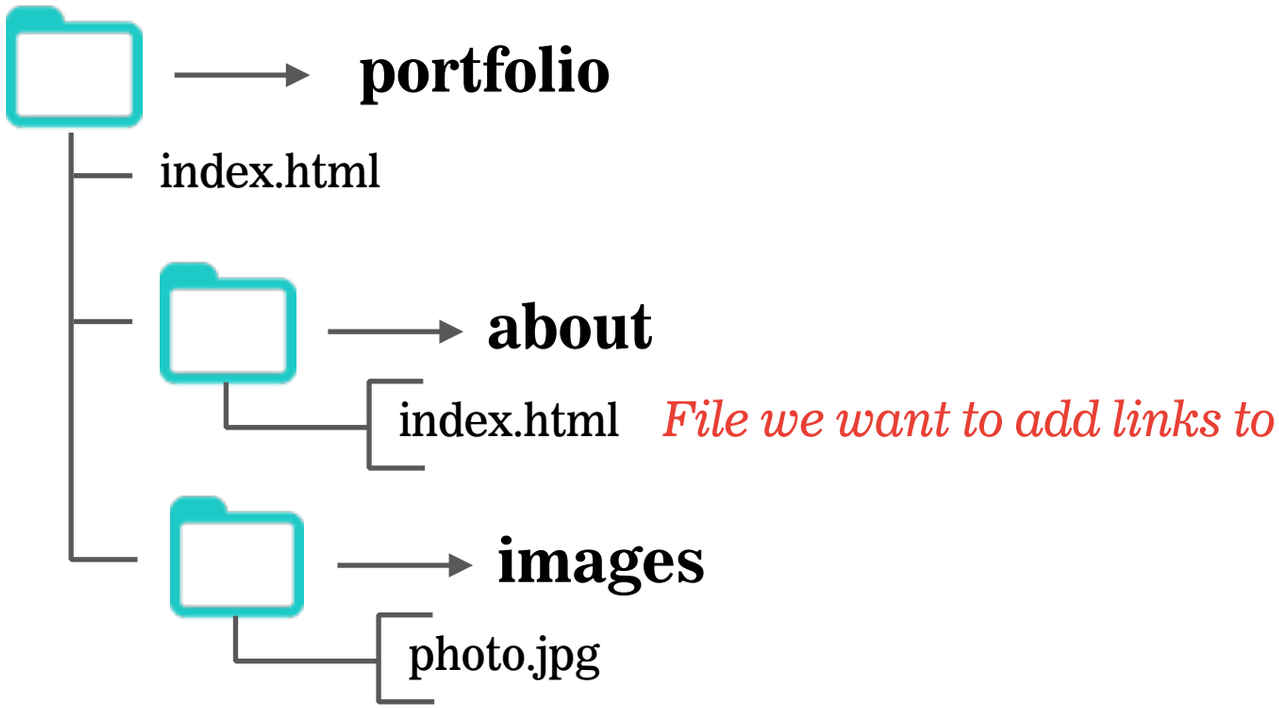
- Starter code > Portfolio Folder

TIMING

4 min

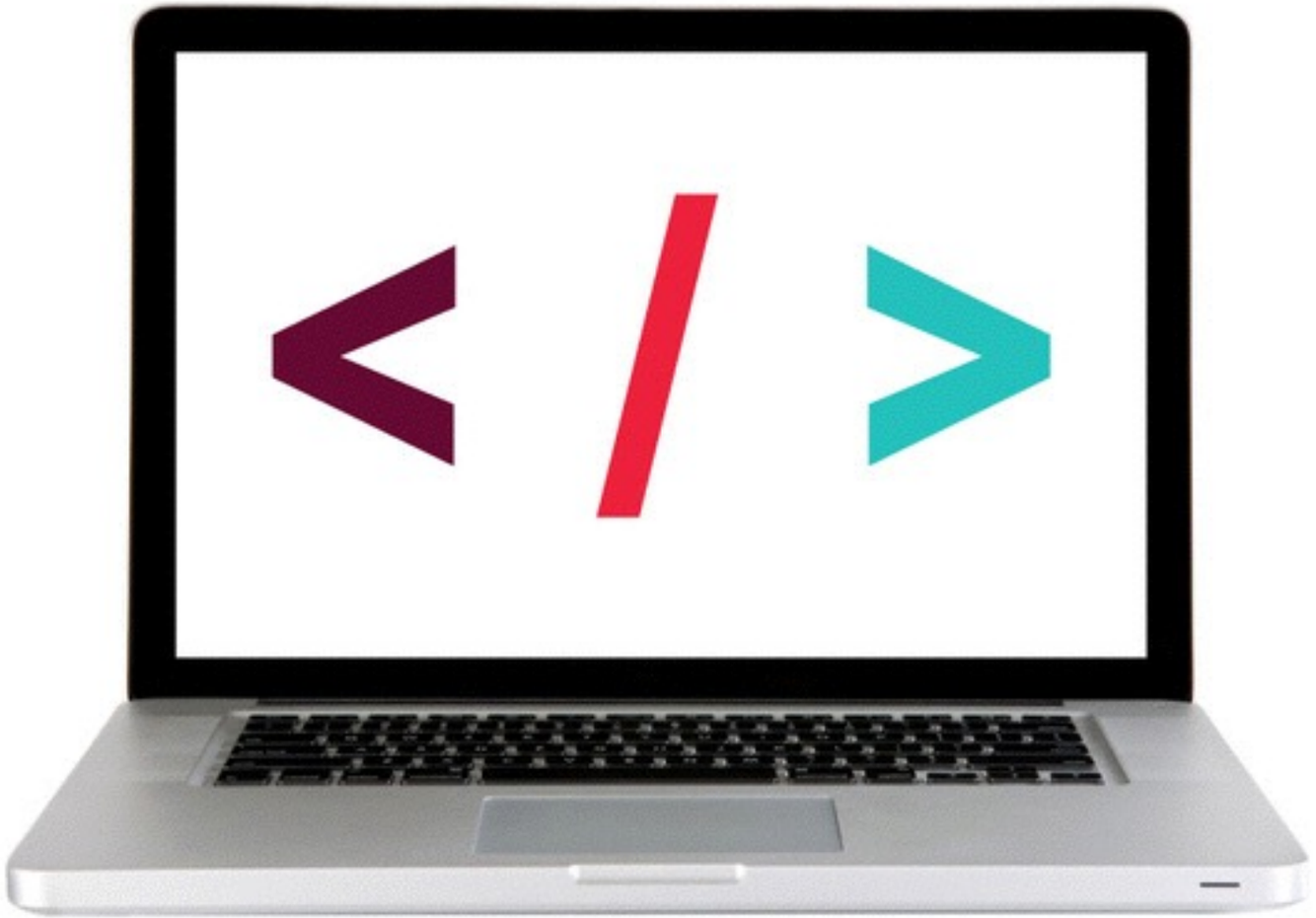
1. With a partner — Follow steps 1 - 3 in resume.html
2. Test in browser!

RELATIVE URLS — CHILD FOLDER



	DESCRIPTION	EXAMPLE
PARENT	../ + path	`<a href="../index.html">Home</a>`
		`<img src="../images/photo.jpg">`

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

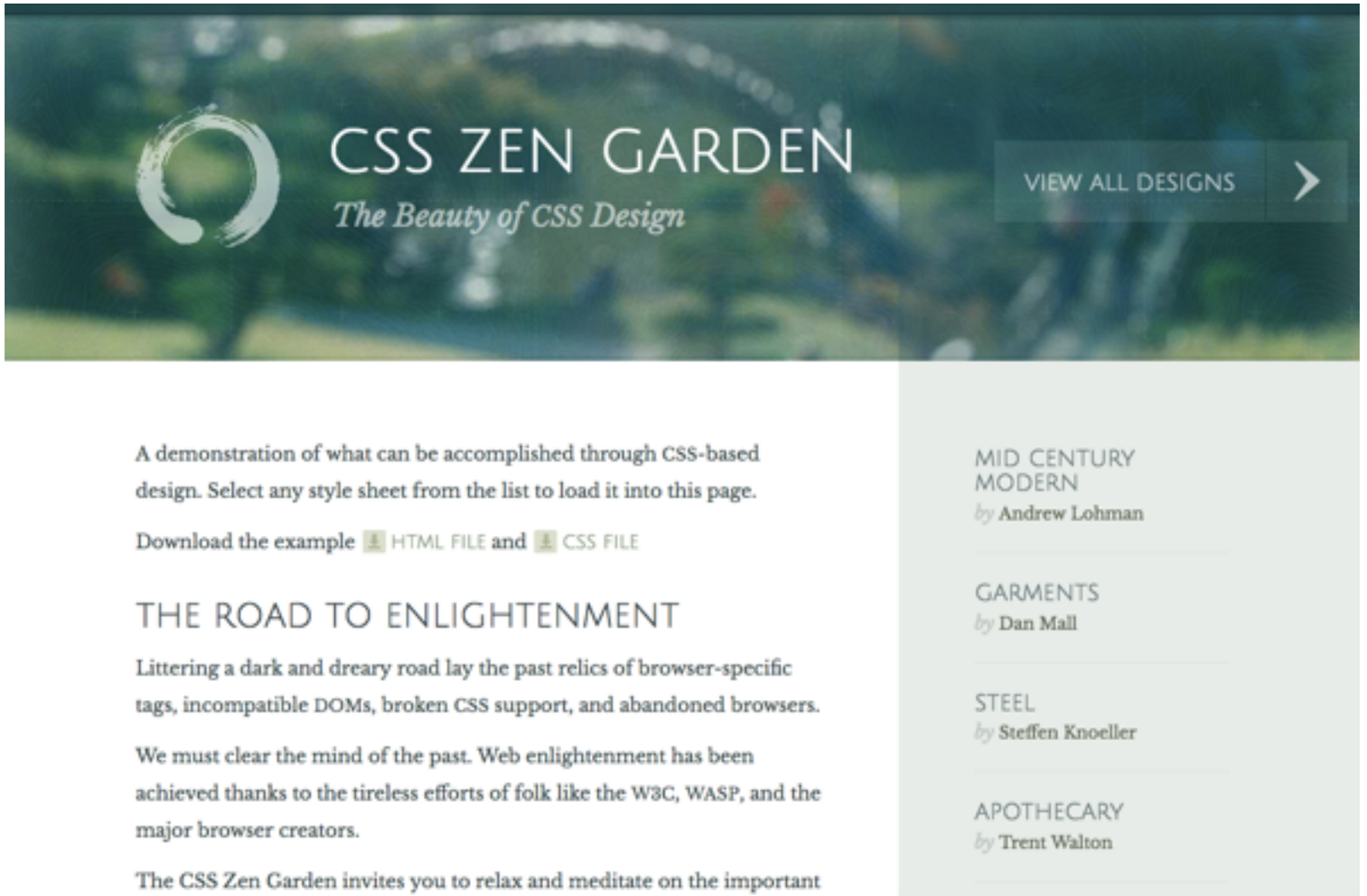
1. With a partner — Follow steps 1 - 3 in resume.html
2. Test in browser!

RELATIVE URLS

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.

WHAT IS CSS?


CSS ZEN GARDEN
The Beauty of CSS Design
VIEW ALL DESIGNS
A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.
Download the example HTML FILE and CSS FILE
THE ROAD TO ENLIGHTENMENT
Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.
We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WASP, and the major browser creators.
The CSS Zen Garden invites you to relax and meditate on the important
MID CENTURY MODERN
by Andrew Lohman
GARMENTS
by Dan Mall
STEEL
by Steffen Knoeller
APOTHECARY
by Trent Walton

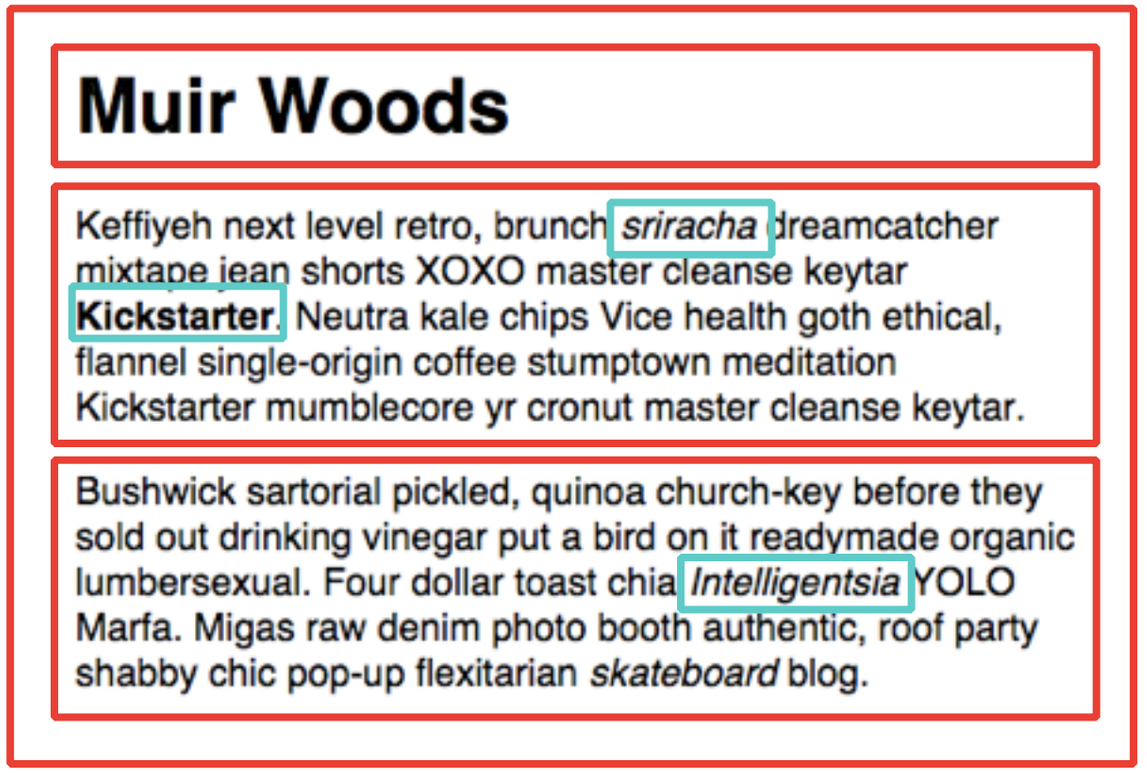
THE BASICS

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.



CSS SYNTAX

```
h1 {
    color: yellow;
    font-size: 16px;
}
```

Property

Value

USING INTERNAL CSS — :(

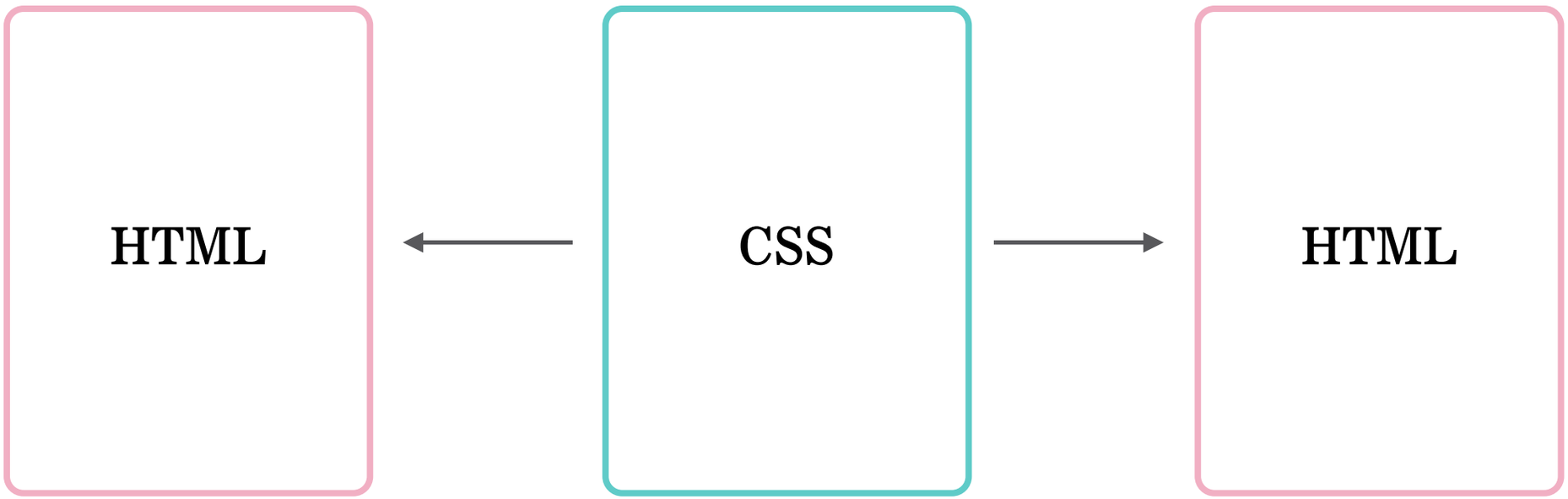
- You can include CSS rules by placing them inside a <style> element, which usually sits inside the <head>.

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

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

</head>
```

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

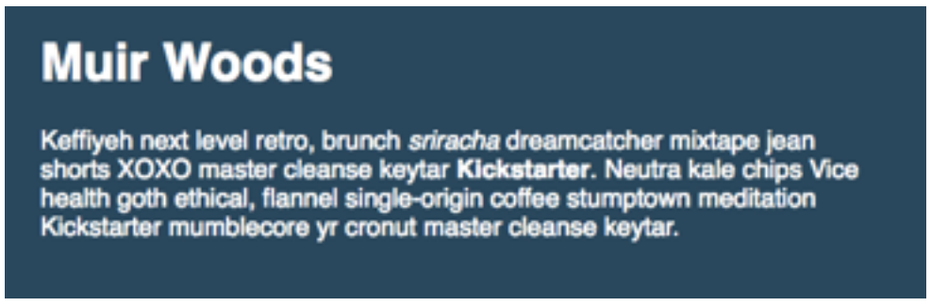
4 min

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

COLOR

COLOR

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

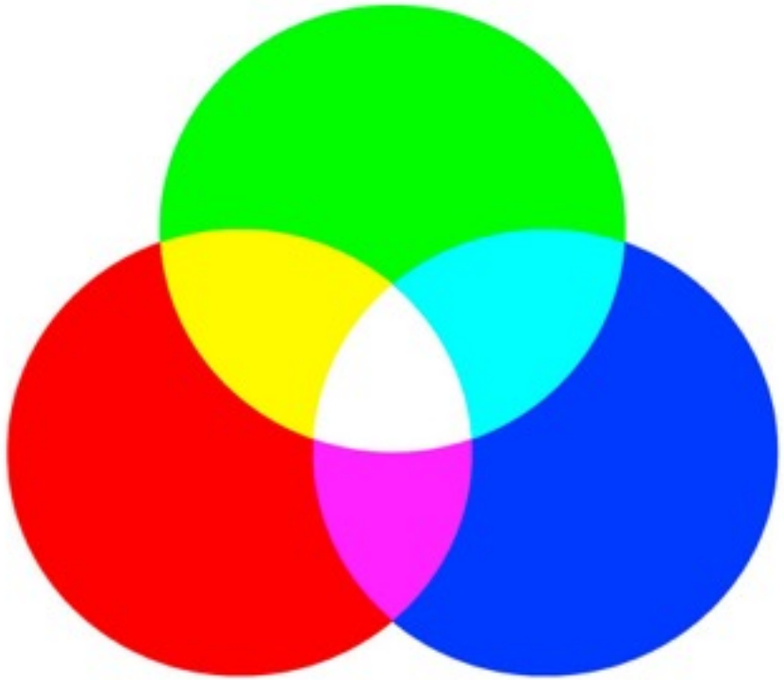



```
body {
  background-color: #22475E;
}
```

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

COLOR

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



RGB VALUES

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

rgb(72, 209, 204)

HEX CODES

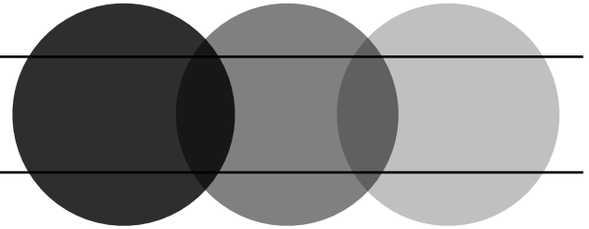
- Represent values for red, green and blue in hexadecimal (base 16) code

#48D1CC

COLOR NAMES

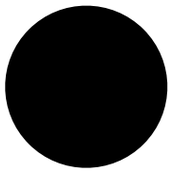
- Colors are represented by predefined names. They are not used very much but are helpful for basic colors such as black and white. Full list of color names

MediumTurquoise

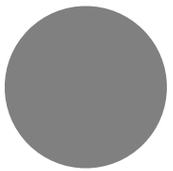


RGBA

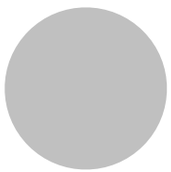
- RGBA works the same as RGB, except that it takes a 4th value called 'alpha'.
- This is a value between 0 and 1 which can be used to determine a color's opacity on the page.



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

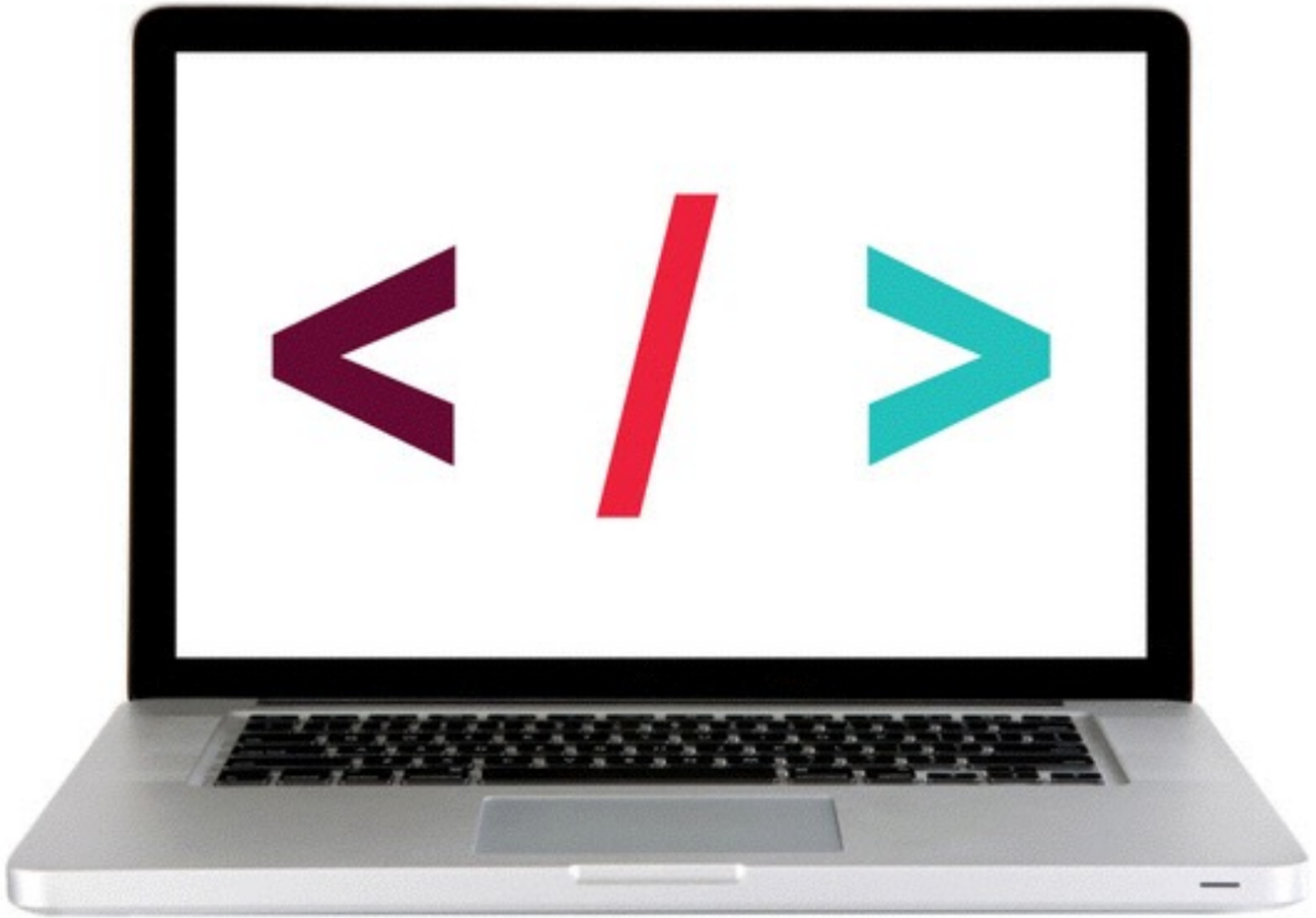


`rgba(0, 0, 0, 0.5)`



`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

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, underline	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
- Times
- Times New Roman
- Arial
- Verdana
- 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

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 9:00

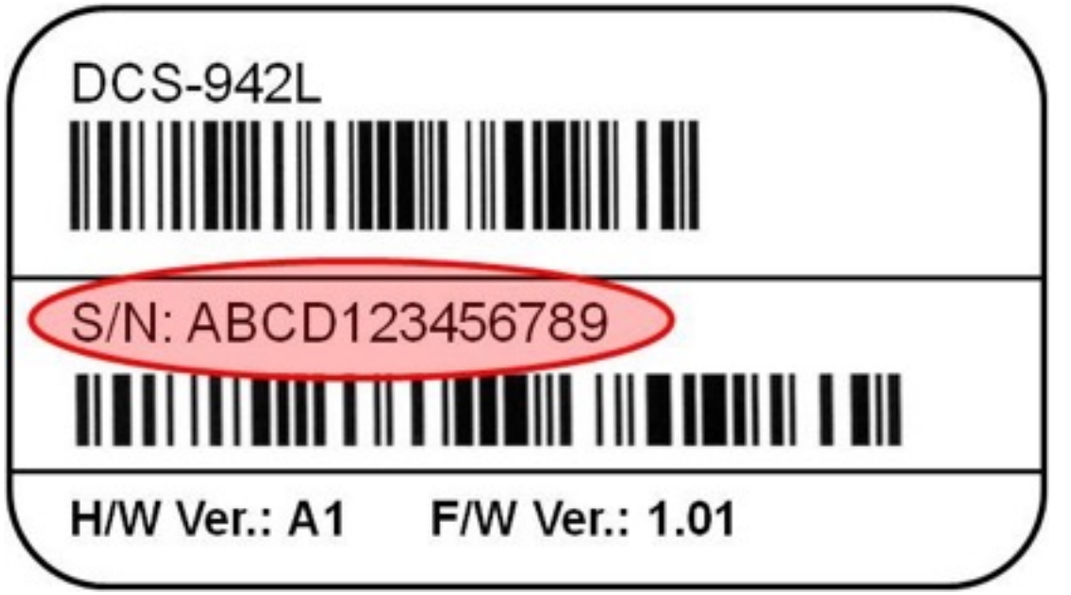
1. Review supplied .pngs starting with Lab…
2. (together) project set up
3. Style Wendy's About Me and Resume pages

CLASSES AND IDS

TARGETING SPECIFIC ELEMENTS

- Classes & IDs allow us to add ‘labels’ to elements so we can target them in our CSS.

TARGETING SPECIFIC ELEMENTS



CLASSES

- Classes are used to group elements together
- Elements can have multiple classes

```
<li class="emphasis">Content</li>
```

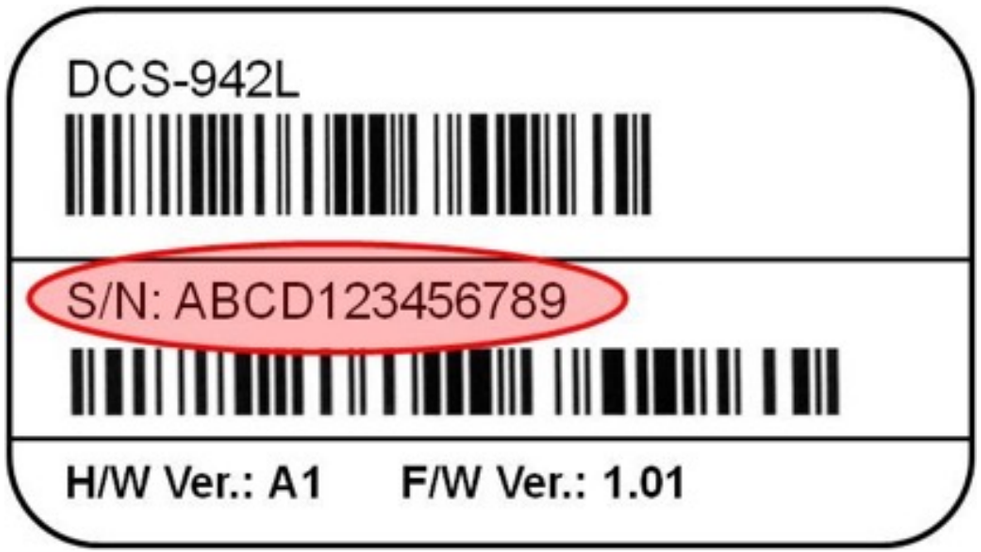
```
.emphasis {
    color: red;
    font-size: 20px;
}
```

IDS

- Ids are used to target *one specific element*
- Each element can only have one id
- **Important**: two elements on the same page cannot have the same id

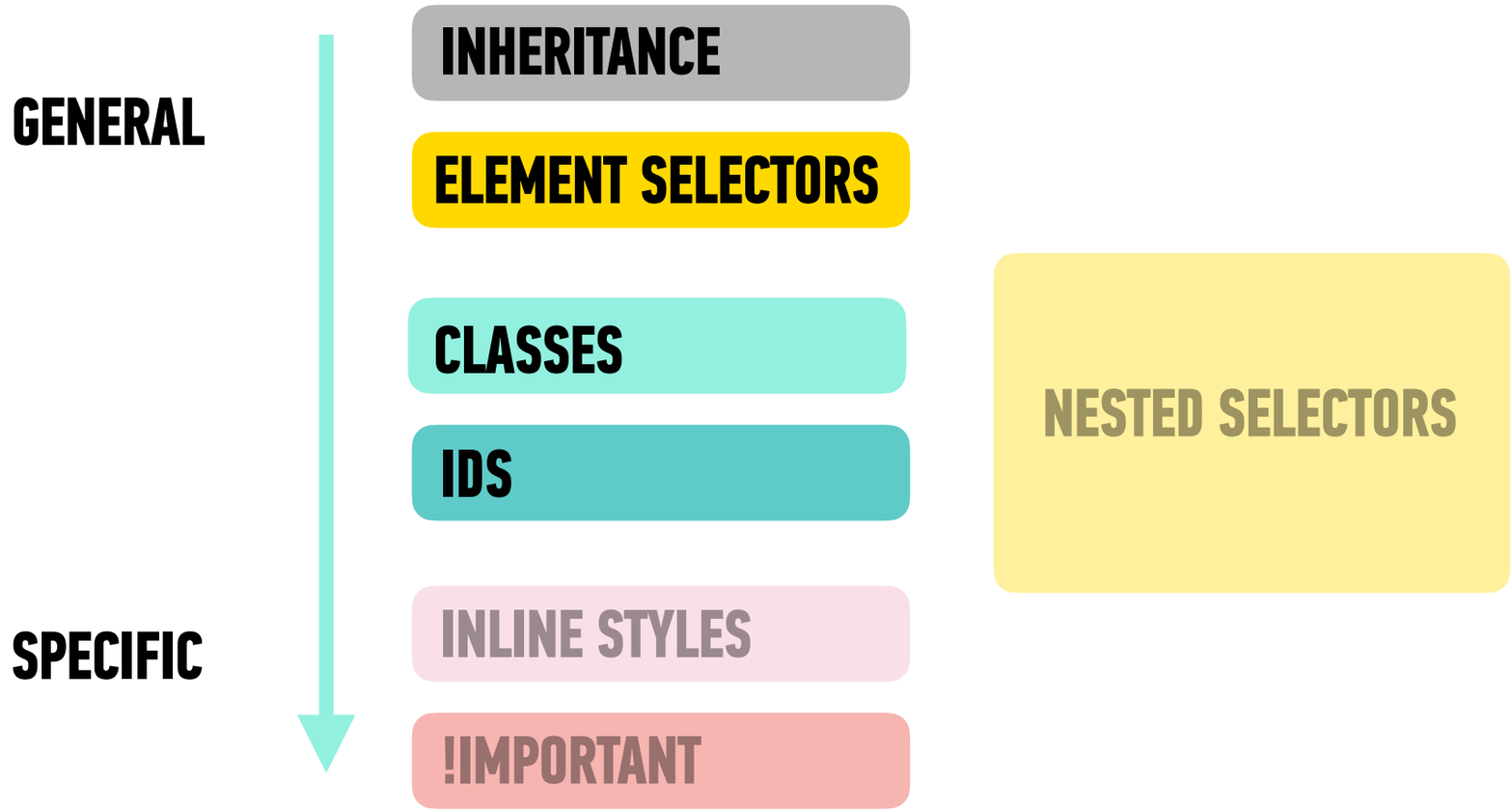
```
<nav id="main-nav">Content</nav>
```

```
#main-nav {
    text-align: center;
}
```



MORE ABOUT CASCADING — GENERAL TO MORE SPECIFIC

- CSS rules cascade downward until they are canceled out by another rule.



- CSS rules are able to override one another and cancel each other out, depending on their order.

ACTIVITY

EXERCISE

LOCATION

- starter_code folder > wendy_bite

KEY OBJECTIVE

- Use classes and IDs to target elements

TIMING

5 min

1. Select add an ID to one element
2. Add a class to two or more elements that make sense to style the same way
3. add CSS to style those elements by ID and class