GENERAL ASSEMBLY

CSS BASICS

Jonathan Cheng

LEARNING OBJECTIVES

- Apply and explain CSS including: how to use selectors, specificity and inheritance
- Describe the DOM and draw a simple DOM tree
- Predict image paths and apply relative paths to <img> and <a> tags
- Experiment with margin, padding, and border

HTML REVIEW

PREFERRED TAGS

What if the content I want to wrap doesn't fall in any of the purpose built tags we've gone over?

<DIV>

If you want a generic box of content, use a tag

<SPAN>

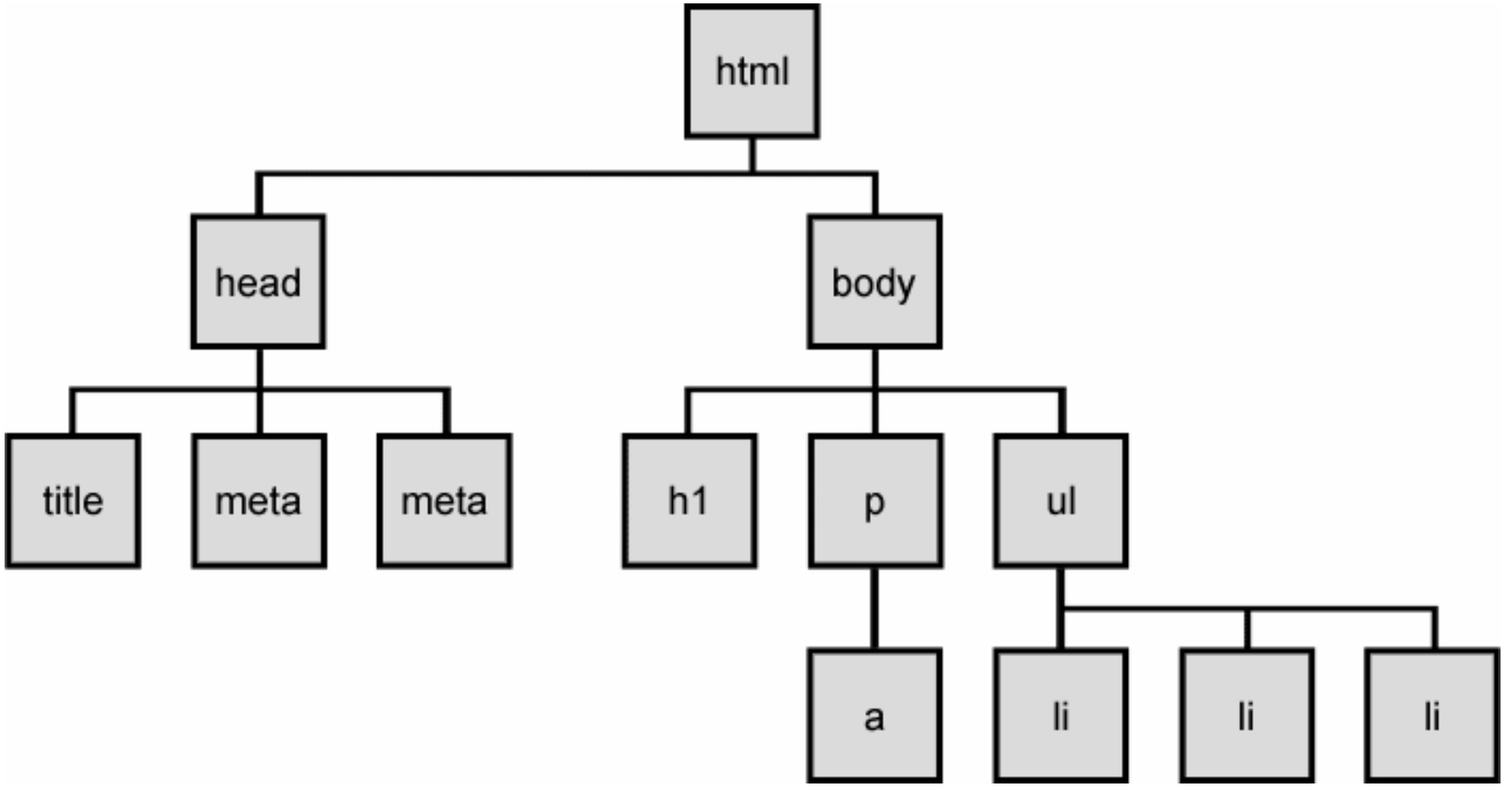
If you just need to wrap a word, phrase, or single line of text, use a tag

DOM

THE DOCUMENT OBJECT MODEL

- The Document Object Model (DOM) is a convention for representing and interacting with objects in HTML
- The nodes (represented by tags) of every document are organized in a tree structure, called the DOM tree, with topmost node named "Document object”
- When an HTML page is rendered in browsers, the browser downloads the HTML into local memory and automatically parses it to display the page on screen.
- The DOM is also the way JavaScript transmits the state of the browser in HTML pages

THE DOCUMENT OBJECT MODEL



CSS SELECTORS

There are three ways to target elements that will handle 95% of your css selecting needs

```
h1{
    color: red;
    font-size: 40px;
}
```

```
<h1>My Blog</h1>
```

```
.sidebar{
    color: blue;
    font-weight: bold;
}
```

```
<a class="sidebar" href="/about.html">
    About
</a>
<a class="sidebar" href="/contact.html">
    Contact
</a>
```

```
#popup{
    visibility: hidden;
}
```

```
<div id="popup">
    50% off winter sale!
</div>
```

CSS SELECTORS: BREAD AND BUTTER

- **Tag names:** should only be used for site wide defaults
  example: we want all h1 tags across the site to be 50px tall, so we'll use a tag name selector

- **Classes:** The most flexible selector. Use this for 90% of your css targeting. Each class can be applied to multiple elements on the page
  example: our sidebar links should look different than other links on our blog. A class is perfect for this

- **Ids:** The most powerful selector. Each id can only be applied to one element on the page. Best reserved for special javascript interactions.
  example: We want to target a hidden modal to popup using javascript if the user clicks a button

Selectors can be strung together in a few ways to increase specificity

```
<ul>
  <li class="highlight">Item 1</li>
  <li>Item 2</li>
  <li class="emphasis">Item 3</li>
  <li class="sub_item">Item 4</li>
  <li>
      <div class="emphasis">Item 5</div>
     <div class="sub_item">part 1</div>
  </li>
</ul>
```

Item 1
Item 2
Item 3
Item 4
Item 5
part 1

```
li.highlight {
    color: red;
}

ul > .emphasis {
    font-weight: bold;
}

ul .sub_item{
    color: green;
}
```

CSS SELECTORS: STRUNG TOGETHER

- Adding two selectors together without space between them will target a dom element with both selectors

- Adding two selectors together separated by a space will target elements that match selector 2 inside elements that match selector 1, regardless of how many levels deep

- Adding two selectors together separated by a > will target elements that match selector 2 inside elements that match selector 1 only one level deep

```
li.highlight {
    color: red;
}

ul .sub_item{
    color: green;
}

ul > .emphasis {
    font-weight: bold;
}
```

CSS SPECIFICITY

CSS SPECIFICITY: DECIDING WHAT RULES GET APPLIED

CSS rules follow a cascade algorithm. If multiple rules apply the same attribute to a dom element, the browser applies the most specific rule

- Universal selectors `* {color: red;}` (least specific)
- Type (tag name) selectors `span {color: red;}`
- Class selectors `.highlight {color: red;}`
- Attributes selectors `[type="text"] {color:red;}`
- Pseudo-classes `a:hover {color: red;}`
- ID selectors `#logo {color: red;}`
- Inline style `<span style="color: red;">some text</span>` (most specific)

ORDER MATTERS

If two css rules have the same specificity as calculated by the cascade, the rule that came last is applied. This applies within files and across files.

CSS INHERITANCE

Certain attributes will pass their values down to children of the element selected

- For example, if you set the body tag of a page to a specific font, that font will be inherited by other elements inside it, such as headers and paragraphs
- This allows us to style a single parent element without having to style all the children elements separately
- Not all properties are inherited, but you can force ones to be by using the inherit value.

CSS PARTING WORDS

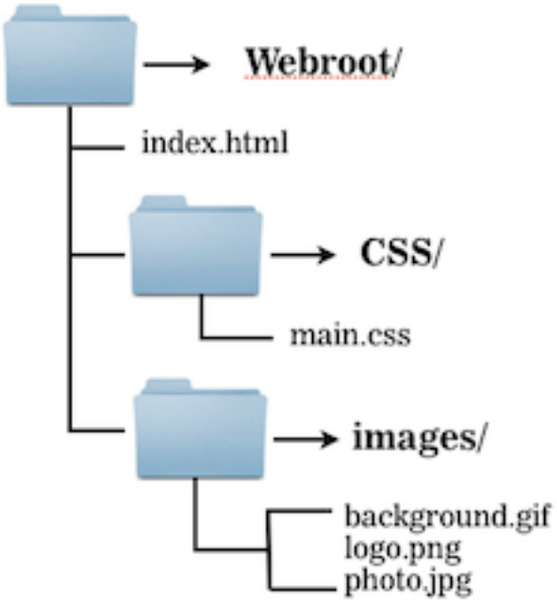
PITFALLS

- Relying on increasing specificity to apply rules leads to selector Cold War
- Relying on the order of your rules without specificity can break your css if you start moving rules into different files or different locations in the same file
- Naming classes by their appearance will make them irrelevant as soon as you decide to change the look
- Not annotating your css with comments will make it difficult to section off what rules belong to areas of the page

BEST PRACTICES

- **Rule 1:** When in doubt, use a single class
- **Rule 2:** Only be as specific as you need to be
- **Rule 3:** Use descriptive unique class names
- **Rule 4:** /* Comment your css to label sections */

PATHING

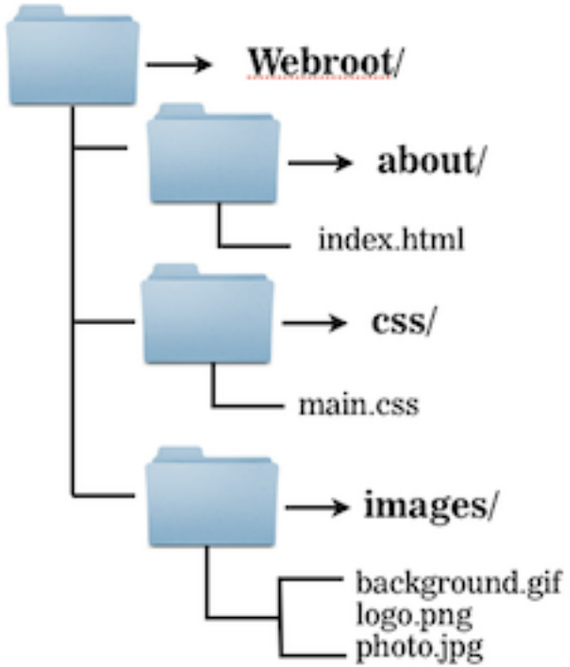


RELATIVE PATHS

```
<!- - in index.html - ->

<nav>
    <img src="images/logo.png" alt="site logo">
</nav>
```

We can navigate our file structure by naming the folders we need to navigate into to find logo.png. Since this img tag is in index.html, this relative path starts at the same directory level as index.html

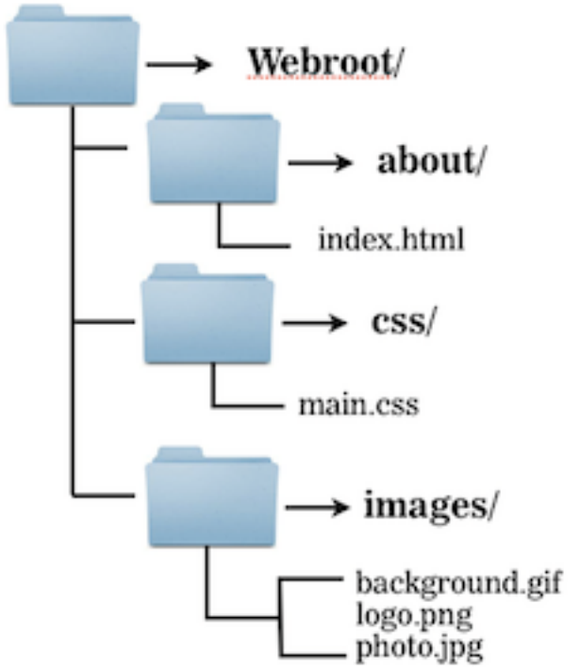


UP A LEVEL

```
<!- - in index.html - ->

<nav>
    <img src="../images/logo.png" alt="site logo">
</nav>
```

If we need to, we can navigate upward using ../
This can be chained together to move up multiple levels like so: ../../somepath/image.jpg



USING WEBROOT

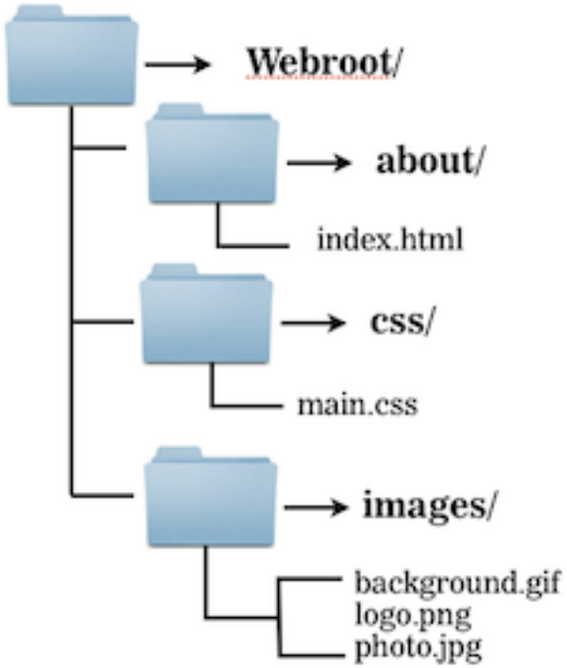
```
<!- - in index.html - ->

<nav>
    <img src="/images/logo.png" alt="site logo">
</nav>
```

Webroot is a folder your web server chooses as the root of all files it makes available to the web (often called public). If our files are hosted on a web server, we can refer to the webroot for all our paths by starting our path with /

This will not work on your local file system.

ABSOLUTE PATHS



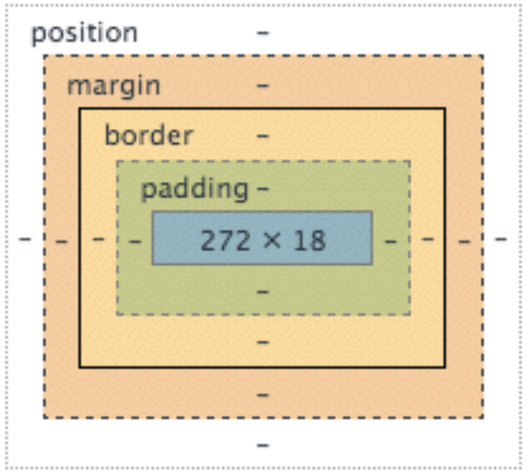

```
<!- - in index.html - ->

<div>
    <a href="http://someothersite.com">Go to SomeOtherSite</a>
</div>
```

If we'd like to refer to web pages outside our site, we can use an absolute path by giving a full url. This will work on your local file system.

MARGIN
PADDING
BORDER

MARGIN



Margin represents the outer layer of space that separates a DOM element from other elements. we can change this a few ways:

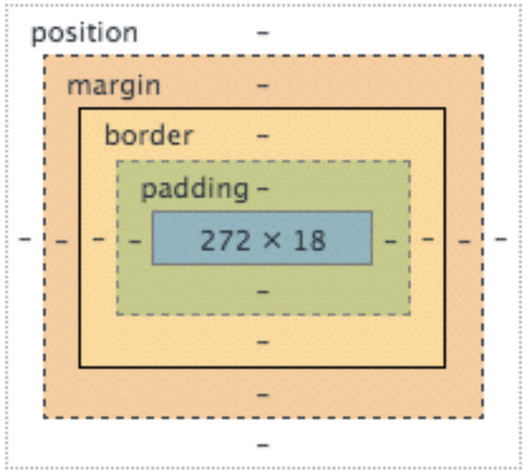
```
margin: 5px;                    /*add 5px of margin around all sides */

margin: 5px 10px;               /*add 5px on top/bottom and 10px left/right */

margin: 5px  0px 10px;          /*add 5px on top, 0px left/right, 10px bottom */

margin: 5px 0 10px 7px;         /*add 5px on top, 0px on right, 10px on the
                                bottom, and 7px on the left  (clockwise from top) */
```

PADDING



Padding represents the inner layer of space that separates the content of an element with its border.

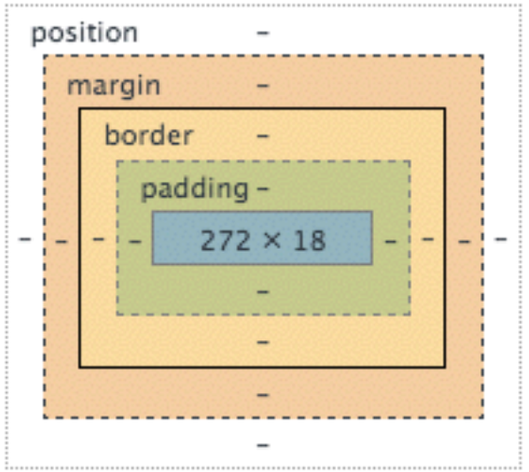
```
padding: 5px;                  /*add 5px of padding around all sides */

padding: 5px 10px;             /*add 5px on top/bottom and 10px left/right */

padding: 5px  0px 10px;        /*add 5px on top, 0px left/right, 10px bottom */

padding: 5px 0 10px 7px;       /*add 5px on top, 0px on right, 10px on the
                                bottom, and 7px on the left (clockwise from top)
*/
```

BORDER



Border represents a (usually) colored layer separating the element from surrounding content.

```
border: 2px solid red;              /*add a 2px red border to all sides */

border-bottom: 1px solid #ccc;      /*add a 1px grey border to the bottom*/
```

LAB

WRAP UP

LEARNING OBJECTIVES

- Apply and explain CSS including: how to use selectors, specificity and inheritance
- Describe the DOM and draw simple DOM tree
- Predict image paths and apply relative paths to <img> and <a> tags
- Experiment with margin, padding, and border

HOMEWORK

1. CREATE A RESUME
2. POST TO CODEPEN.IO
3. LINK IT TO US VIA SLACK

HOMEWORK

- Your assignment this week is to create an online resume
- You can write a resume for anyone fictional, historical, or real
- Create an about me and resume page and style it with CSS
- Link the two pages together
- Use classes to target elements you'd like to style
- Link to at least one outside resource for more information (like a wikipedia entry)
- Link to at least one image (can be relative or absolute)
- **Bonus points:** make the style fit the theme of the person you're using