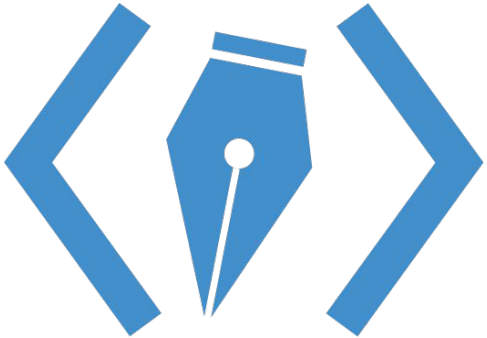
Week 2

CSS Selectors, Pseudo-selectors



Announcements

Add Deadline Tomorrow

Please use your enrollment code; expires after tomorrow.

Homework 1

Technically due right now, please don't do your homework last minute!

Piazza

Check Portal Submission Debugging on Piazza 📍
We try our best, but sometimes we don't get to every question.

OH/Email

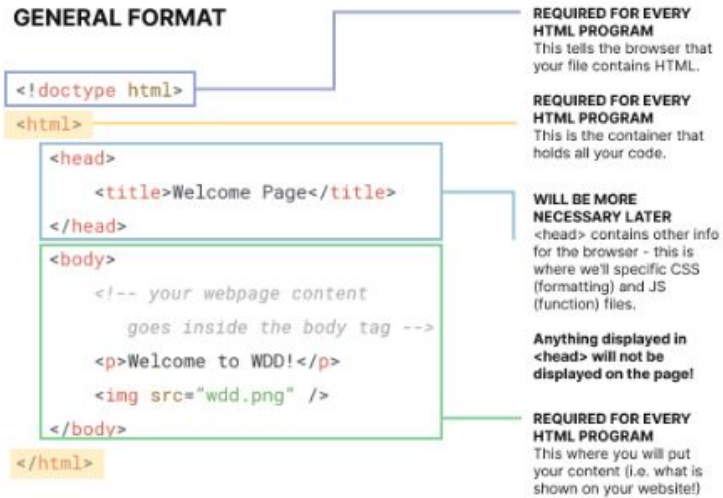
For more instant, in-depth help, visit us at office hours.
Email us if you have **logistical** questions.

Give us anonymous feedback at wdd.io/go/feedback

Add songs to our class playlist: wdd.io/go/spotify

HTML CHEAT SHEET

GENERAL FORMAT



GOOD HABITS

1. Keep your tags lined up, especially starting and ending tags. This will help you keep track of whether or not you forgot to add your closing tags.

2. Use comments to break up what your sections mean. Use alt-text to describe images for accessiblity features.

3. Strategize how you want to group your code before you write it! (Mainly relevant for projects and writing code from scratch!

IMPORTANT + USEFUL TAGS

** ** This denotes a paragraph of text. Separate your paragraphs with another line of paragraph tags. ** **

**
** This denotes a line break.

<!-- This is a comment and will be ignored by the browser. It helps to make notes about what you're doing and how. **-- >**

**** This is where the clickable text goes!****

****

**** This makes text bold but still treated as normal text. ****
**** This italicizes text! ****

<h1> This creates a header, which is treated to be bold large text (i.e. differently than). h1 is the largest version and as the number increases, the size of the text decreases. **</h1>**

<ul>
  **
** This is an item in a bullet point list, also known as an "unordered list." **
**
  **
** This is another item in the list. Use <ol> tag instead of <ul> for an ordered list. **
**
**
**

** ** This denotes a container for you to organize your code into groups (more important later with CSS. The "<section>" tag does the same thing. ** **

<div class="class name"**>** Everything in this div belongs to the specified class. More on this in CSS Cheat Sheet. ** **

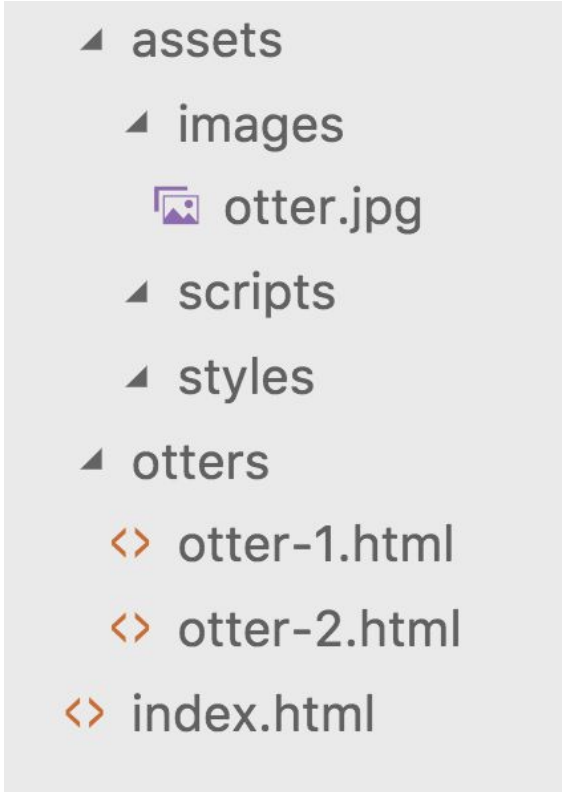
<div id="id name"**>** Everything in this div belongs to the specified ID. More on this in CSS Cheat Sheet. ** **

<title> This goes in the "<head>" tag and specifies the title of the page that shows up on a tab of a web browser, at the top of a window, and on a Google result. **</title>**

Now that we've learned about HTML

How do we organize our project workspace?

(The following may/may not apply to industry standards)




```
assets      - Where all static resources go
    images    - .png, .jpg, .gif, etc
        otter.jpg
    scripts   - Where JavaScript files are
    styles    - Where CSS files are
otters      - Sub-directory here for better structure
    otter-1.html
    otter-2.html
index.html - The entry HTML file
```

Example project (follow link to view/download)

Relative path

- Path to another file relative to the current file
- It may point to different files depending on the current location

Examples:

- `assets/images/otter.jpg`
- `./assets/images/otter.jpg`
- `../assets/images/otter.jpg`

dot-dot-slash (. . /) to go up to parent directory

dot-slash (. /) stays in the same directory

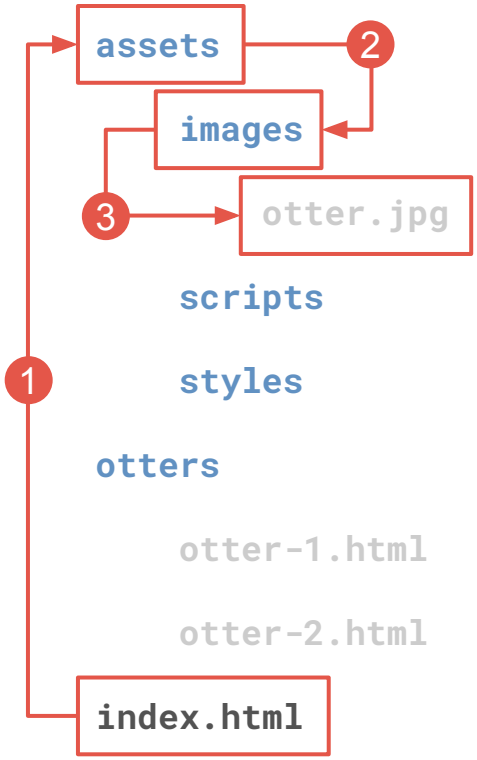
Absolute path

- Complete path to another file
- Since it's not relative, the absolute path always points to the same exact file

Begins with a slash

Examples:

- `/assets/images/otter.jpg`
- `https://wdd.io/assets/images/otter.jpg`
- `file:///Users/slu/Pictures/otter.jpg`

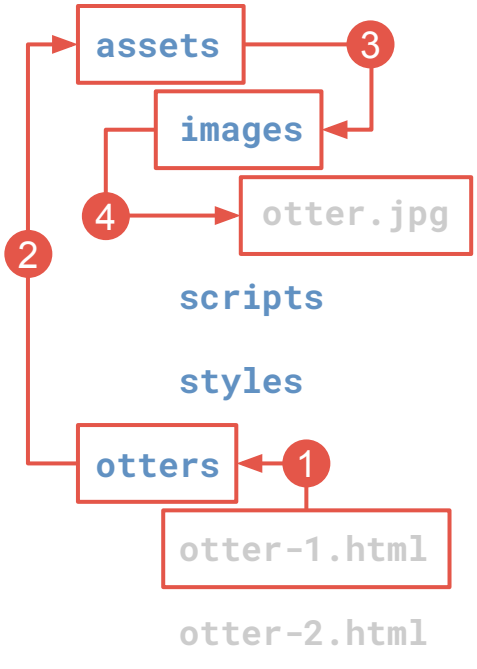



```
<!doctype html>
<html>
   <head></head>
   <body>
       <h1>Monterey Bay Museum</h1>
       <p>Come to see the otters!</p>
       <ul>
           <li><a href="otters/otter-1.html">Otter 1</a></li>
           <li><a href="otters/otter-2.html">Otter 2</a></li>
       </ul>
       <img src="assets/images/otter.jpg" width="200" />
   </body>
</html>
```

1 2 3

index.html

Relative path from **index.html** to **assets/images/otter.jpg**



index.html

```
<!doctype html>
<html>
   <head></head>
   <body>
       <a href="../index.html">Back to Home</a>
       <h1>Monterey Bay Museum</h1>
       <h2>Otter 1</h2>
       <p>Otter 1 [1] [2] just [3] [4] 2.</p>
       <img src="../assets/images/otter.jpg" width="200" />
   </body>
</html>
```

otters/otter-1.html

Relative path from `otters/otter-1.html` to `assets/images/otter.jpg`

We usually prefer relative paths to absolute paths

HTML

Structure

CSS

Design

JavaScript

Function

Files and Linking CSS

CSS Syntax

CSS Selectors

Pseudo-Selectors

Files and Linking CSS

CSS Syntax

CSS Selectors

Pseudo-Selectors

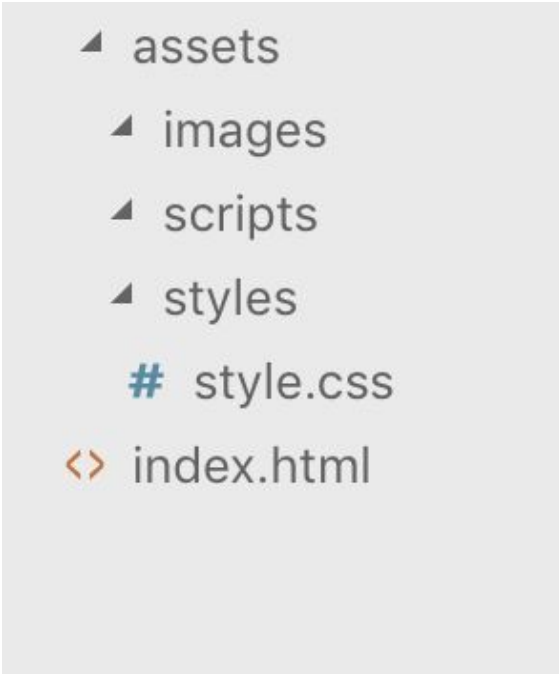
Linking CSS in HTML

How does the HTML file know where to find its css?!

Inside the `<head>` tag, add this line:

```
<link rel="stylesheet" type="text/css"
      href="assets/styles/style.css" />
```

*No need to memorize this — you can just copy and paste this or something



Where our CSS file will live (for this class): `assets/styles/style.css`

Files and Linking CSS

CSS Syntax

CSS Selectors

Pseudo-Selectors


h1 {
color: blue;
background: gray;
}
HTML element(s)
semicolon
CSS property
property value

HTML and CSS: At a Glance (check it out)

index.html

```
<!doctype html>
<html>
    <head>
        <link rel="stylesheet" type="text/css"
          href="style.css">
    </head>
    <body>
        <div>
            <h1>Welcome to WDD!</h1>
            <p>Let's learn CSS.</p>
        </div>
    </body>
</html>
```

style.css

```
div {
    width: 200px;
    background: beige;
    color: darkblue;
}

h1 {
    font-size: 32px;
    font-family: Arial;
}
```

HTML and CSS: At a Glance (check it out)

index.html

```
<!doctype html>
<html>
    <head>
        <link rel="stylesheet" type="text/css"
          href="style.css">
    </head>
    <body>
        <div>
            <h1>Welcome to WDD!</h1>
            <p>Let's learn CSS.</p>
        </div>
    </body>
</html>
```

style.css

```
div {
    width: 200px;
    background: beige;
    color: darkblue;
}

h1 {
    font-size: 32px;
    font-family: Arial;
}
```

HTML and CSS: At a Glance (check it out)

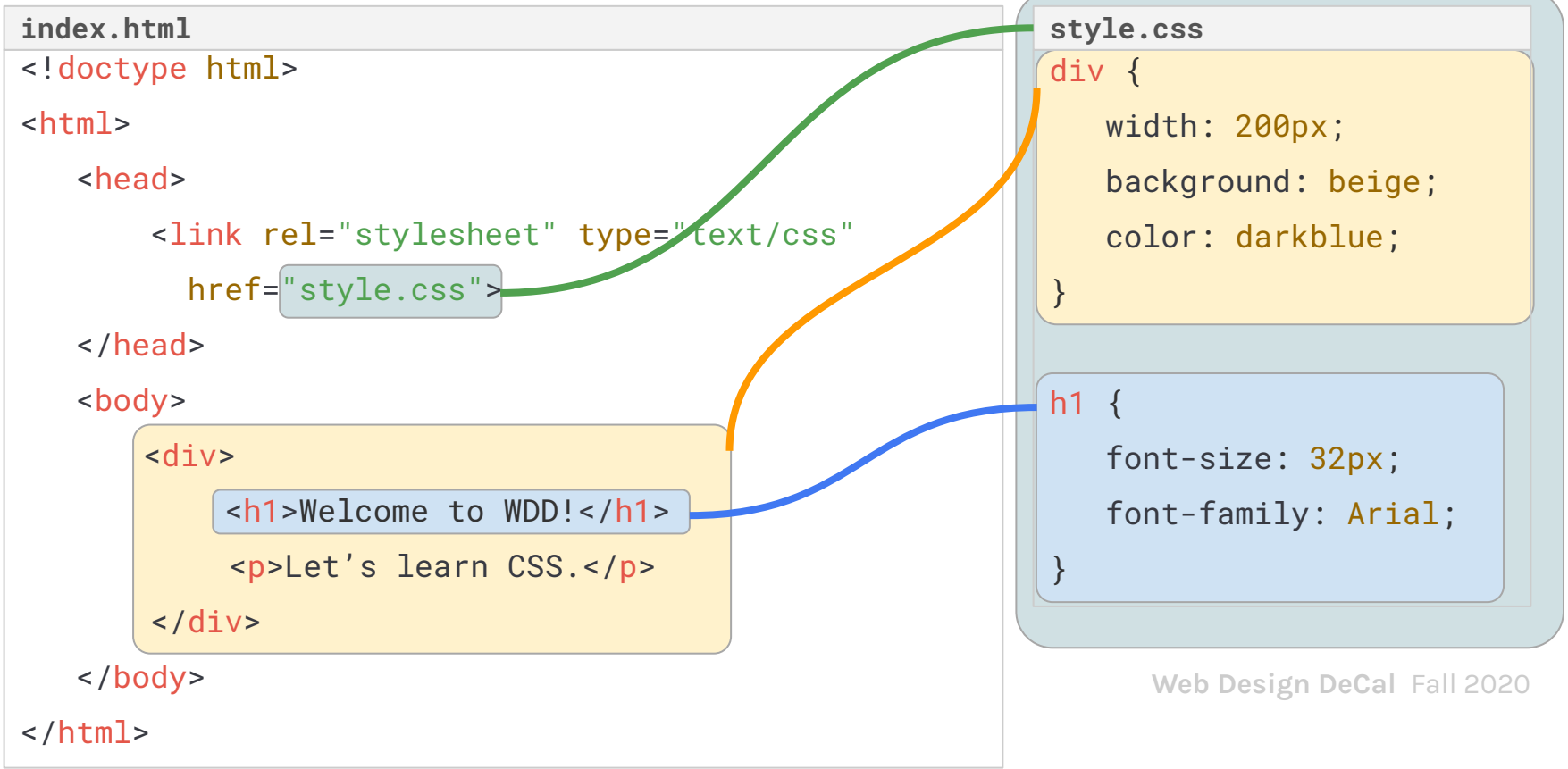
index.html

```
<!doctype html>
<html>
    <head>
        <link rel="stylesheet" type="text/css"
          href="style.css">
    </head>
    <body>
        <div>
            <h1>Welcome to WDD!</h1>
            <p>Let's learn CSS.</p>
        </div>
    </body>
</html>
```

style.css

```
div {
    width: 200px;
    background: beige;
    color: darkblue;
}

h1 {
    font-size: 32px;
    font-family: Arial;
}
```



Selectors

Tells us what elements in the HTML to “select”, or apply these properties to.

- In this case, we are selecting all the `div` elements.
- Lots of ways to specify what to select! Very powerful

```css
div {
    font-size: 32px;
    color: green;
}
```

CSS

Property and Value

Property
Defines **what** we are changing about the selected element

Value
Specifies **how** we are changing it, or what it should be changed to

Together: Declaration (or rule)

```css
div {
    font-size: 32px;
    color: green;
}
```

Miscellaneous, Very Small Things

- Rules (property and attribute pairs) for a selector fall between **curly braces**
- Rules with curly braces are separated by a **semicolon**
- Comments surrounded by **/*** and ***/**

```css
div {
    font-size: 32px;
    color: green;
}

img {
    /* This is a comment */
    width: 50%;
}
```

CSS

Files and Linking CSS

CSS Syntax

CSS Selectors

Pseudo-Selectors

What if I don't want *all* tags the same style?

```html
<p>Green text</p>
<p>Blue text</p>
<p>And more green text!</p>
```

HTML

```css
/* ¡DOES NOT WORK! */
p {
   color: green;
   color: blue;
}
```

CSS

An Attempt

```html
<p>Green text</p>
<div>Blue text</div>
<p>And more green text!</p>
```

HTML

```css
p {
    color: green;
}

div {
    color: blue;
}
```

CSS

Yes Good

```html
<p class="green">Green text</p>
<p id="blue">Blue text</p>
<p class="green">More green text!</p>
```

HTML

```css
.green {
    color: green;
}

#blue {
    color: blue;
}
```

CSS

Classes

- Apply a set of rules to **multiple** elements on the page

- HTML: add the `class` attribute to opening tag
- CSS: put a period in front of the class name as a selector

```html
<p class="green">Green text</p>
<p class="green">Green text 2</p>
<div class="green">Also green</div>
```

HTML

```css
.green {
   color: green;
}
```

CSS

IDs

Apply a set of rules to only **one distinct element** on the page

- HTML: add `id` attribute to opening tag
- CSS: put a hashtag in front of the class name as a selector

```html
<p id="blue">Blue text</p>
```

HTML

```css
#blue {
   color: blue;
}
```

CSS

Tip for Remembering

A (student) `id` is **unique** per student.
An student id is a **number** (#)

Many students take a `class`
Classes are in **periods** (.) in high school

Selecting Multiple Elements

- Separate each element with a comma “,”
- CSS properties and values will be applied to all selected elements.

```html
<p class="one">Lorem</p>
<p class="two">Ipsum</p>
<div id="three">Dolor</div>
```

HTML

```css
.one, .two, #three {
   font-size: 32px;
}
```

CSS

Selecting Element w/ Multiple Names

- Classes can have multiple names within one tag.

- No space between CSS selectors.
- **“Select all elements with both `.green` and `.big` set within its class.”**
- CSS in the example will only apply to the first `<p>` tag, but not the second.

```
<p class="green big">Lorem</p>
<p class="big">Ipsum</p>
```

HTML

```
.green.big {
    color: green;
    font-size: 64px;
}
```

CSS

Selecting Nested Elements

- Space between selectors.
- **“Select all elements with `.park` that are nested within `#jay`.”**
- CSS in the example will only apply to the first `<p>` tag, but not the second.

```
<div id="jay">
   <p class="park">Jay Park <3</p>
</div>
<p class="jay park">all i wanna do</p>
```

HTML

```
#jay .park {
   font-size: 32px;
}
```

CSS

Files and Linking CSS

CSS Syntax

CSS Selectors

Pseudo-Selectors

Pseudo-Selectors

Selects elements that are under a certain **state,** great for **user interaction**

General Pseudo-selector structure:
`normal-selector:pseudo-class`

Rules will be applied to the div
only when the cursor is hovering over

```css
div:hover {
    font-size: 32px;
    color: green;
}
```

CSS

Some Pseudo-Classes

`:hover`

- when mouse “hovers” over an element

`:visited`

- when a link has been visited before (default is purple hyperlink)

`:active`

- when the mouse is currently clicked on an element

… and many, many more!

Summary

- CSS is used to change the default appearance of HTML!
- Goes in a separate file, usually **style.css**
- In the HTML file, we need this line to tell it where to find its styles:

```
<link rel="stylesheet" type="text/css" href="style.css">
```

- CSS looks like this:

```
h1 {
    color: blue;
    background: gray;
}
```

Selectors Summary

1. To apply the css to **a certain html tag**:

```
h1 {
    color: blue;
    background: gray;
}
```

2. To apply the css to **a class of elements**:

```
.nav-link {
    color: green;
    font: 14px Arial;
}
```

3. To apply the css to **one specific element**:

```
#logo {
    width: 200px;
    margin: 30px;
}
```

4. To apply the css to **multiple things**:

```
h1, h2, #special-heading {
    color: blue;
    background: gray;
}
```

Pseudo-selectors Summary (let's style links!)

1. To apply the css to **a link**:

```
a {
    color: darkblue;
    text-decoration: none;
}
```

2. What should the link look like when someone **hovers** over it?

```
a:hover {
    color: blue;
    text-decoration: underline;
}
```

3. What about when the mouse is **currently clicked on it**?

```
a:active {
    background: lightgrey;
}
```

4. What about when the link has been **visited before** (defaults to purple)?

```
a:visited {
    color: darkblue;
    text-decoration: none;
}
```

Changing how text appears:

```
color: blue;
font-size: 30px;
font-family: "Comic Sans";
text-decoration: underline;
font-weight: bold;
line-height: 1.5;
```

Changing size and spacing:

```
width: 200px;
height: 100px;
margin: 30px;
padding: 30px;
```

Make the generic “box” nicer:

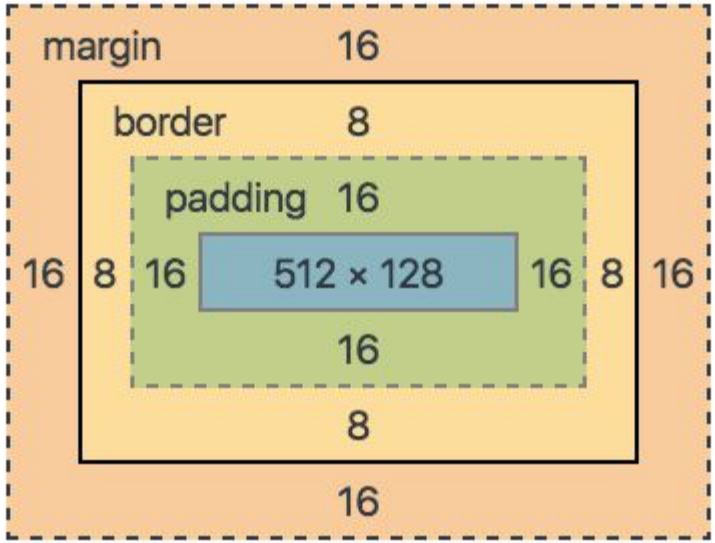
```
background-color: lightblue;
border: 10px solid black;
border-radius: 10px;
```

More on the **importance of spacing** in the design lecture!

```
width: 512px;
height: 128px;
```

```
padding-top: 16px;
padding-right: 16px;
padding-left: 16px;
padding-bottom: 16px;
```

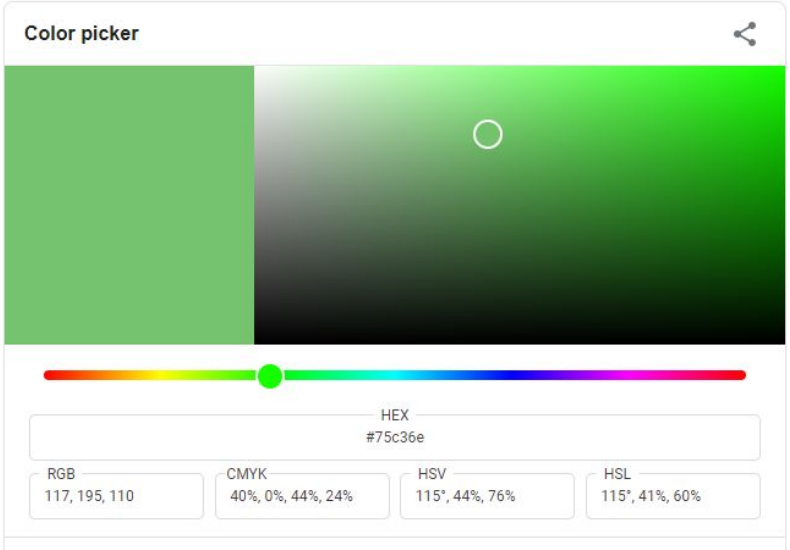
```
margin-top: 16px;
margin-right: 16px;
margin-left: 16px;
margin-bottom: 16px;
```



Note: colors

Throughout these slides, we've only used very basic, convenient colors (that the browser understands), like `blue`, `green`, `lightblue`, `beige`, etc

To use a specific color, Google 'color picker', use it to pick the color you want, then copy the **HEX code** (the one that starts with #)



For example, **WDD green** is **#75c36e**

Demo: Let’s make hw1 look nice!

wdd.io/go/htmlcss-demo

Solution: wdd.io/go/htmlcss-demo-sol

Helpful Links

- More practice: tinyurl.com/wdd-cssbasics
- CSS selectors: https://www.w3schools.com/cssref/css_selectors.asp
- Fun game with sushi: https://flukeout.github.io/
- CSS properties: https://www.w3schools.com/cssref/default.asp

Questions?