FEWD Class 3

Review

- Basic structure of an html5 page
- Linking files
- Block vs. Inline elements
- CSS Box Model
- Fixed vs. Relative Units of Measurement
- Colors
- Cascading and weighting for CSS rules
- CSS Combinators

Code Along

- Reproducing the About Me page

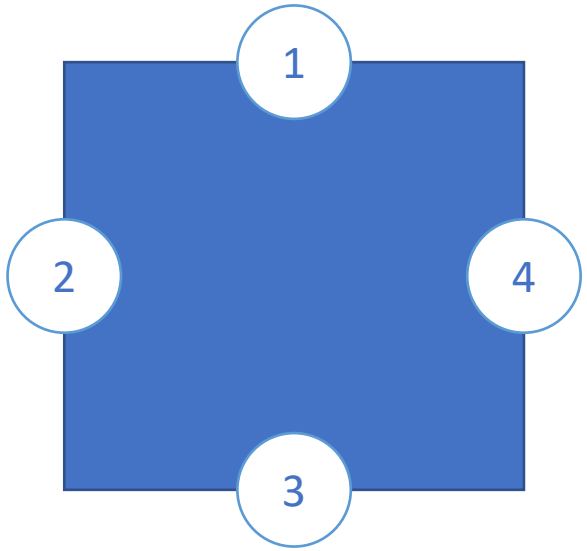
Objectives

- Shorthand Properties
- Classes and IDs
- Floats
- Centering

Shorthand Properties

- Many CSS properties have single properties that combine several properties into one.
  - Margin & Padding
  - Border
  - Background
  - Font

The **TROUBLE** with Padding & Margin



When 4 values are supplied they are applied in a counterclockwise direction from the top:

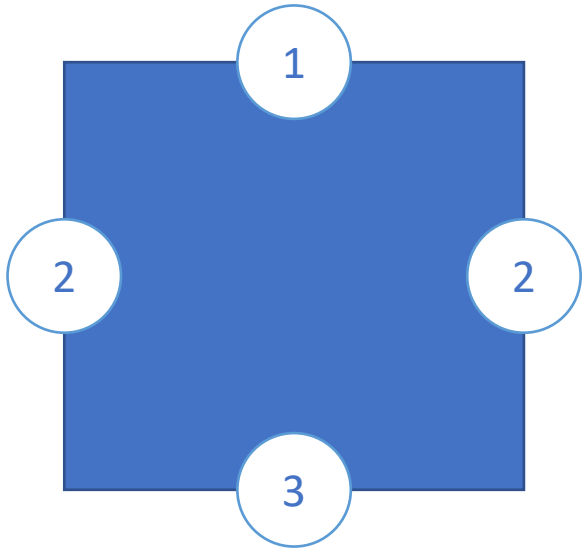
```
.box {
        margin: 4px 2px 10px 0;
}
```

4px on the top

0px on the left

Remember the mnemonic TROUBLE => **T** (top) – **R** (right) – **B** (bottom) – **L** (left)

Padding & Margin



When 3 values are supplied they are applied to the top, left and right, bottom:

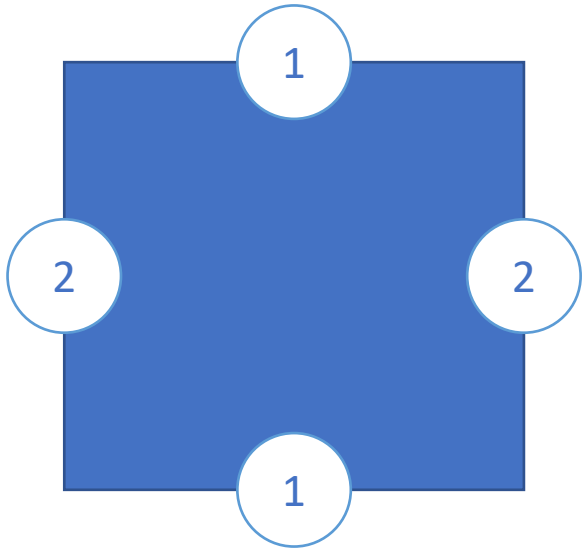
```
.box {
        margin: 4px 10px 8px;
}
```

4px on the top

10px on the left and right

8px on the bottom

Padding & Margin



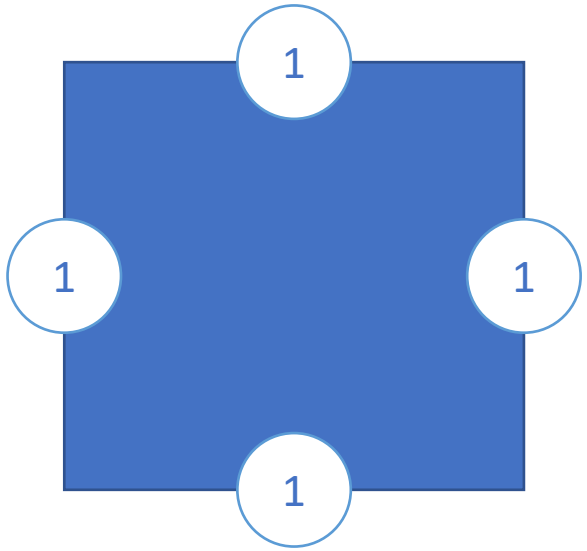
When 2 values are supplied they are applied to the top and bottom, left and right:

```
.box {
        margin: 4px 10px;
}
```

4px on the top and bottom

10px on the left and right

Padding & Margin



One value is applied to all sides:

```
.box {
        margin: 10px;
}
```

10px on all sides

Border

- With borders, the width, color, and style can be combined into one declaration
- Unlike margin and padding, these are uniformly applied

Instead of:

```
.box {
        border-width: 1px;
        border-style: solid;
        border-color: #000;
}
```

You can write

```
.box {
        border: 1px solid #000;
}
```

Background

Instead of:

```
.box {
  background-color: #000;
  background-image: url(images/bg.gif);
  background-repeat: no-repeat;
  background-position: left top;
  background-size: cover;
}
```

You can write

```
.box {
   background: #000 url(images/bg.gif) no-repeat left top / cover;
}
```

Background

- This works for multiple backgrounds as well…

Instead of:

```
.box {
  background-image: url(images/bg.gif), url(images/sm-logo.gif);
  background-repeat: repeat, no-repeat;
  background-position: center, bottom right;
  background-size: 100%, 30%;
}
```

You can write

```
.box {
   background: url(images/bg.gif) repeat center / 100%, url(images/sm-
logo.gif) no-repeat bottom right / 30%;
}
```

Font

Instead of:

```
.box {
   font-style: italic;
   font-weight: bold;
   font-size: .8em;
   line-height: 1.2;
   font-family: Arial, sans-serif;
}
```

You can write

```
.box {
   font: italic bold .8em/1.2 Arial, sans-serif;
}
```

Code Along

- Clean up our About Me page

Lab

- Practice on your own with shorthand properties
- Use at least one border, margin or padding, and font shorthand
- For the background, try a harder one and make a stacked background with two images

Classes and IDs

- Classes and IDs allow us to target elements without having to use the tag as a selector
- You can combine them with tag selectors

IDs

- An ID may **only** be used **once** on a page
- An element may only have one ID
- HTML:

```
<div id="extra-special"></div>
```

- CSS:

```
#extra-special {
    …
}
```

Classes

- Classes are reusable as many times as you want
- An element can have as many classes as you want
- HTML:

```
<div class="big primary"></div>
```

- CSS:

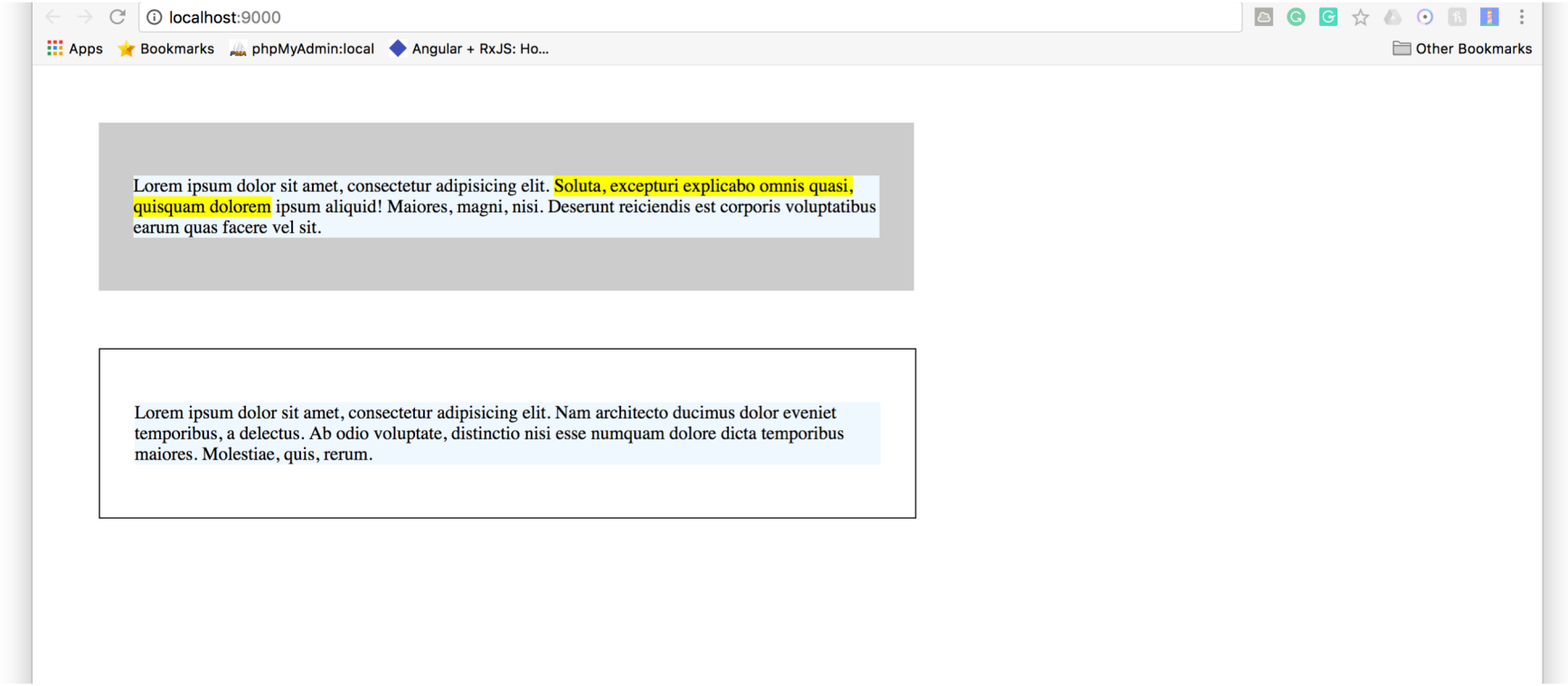
```
.big {
        …
}
.primary {
        …
}
```

Code Along

- Using classes and ids in CSS

Lab

- Practice with using CSS classes and IDs
- Reproduce the design below:



Floats

- The float property accepts the values:
  - **right**
  - **left**
  - none (default)
  - initial (resets to the default)
  - inherit (gets its value from its parent)

Clear

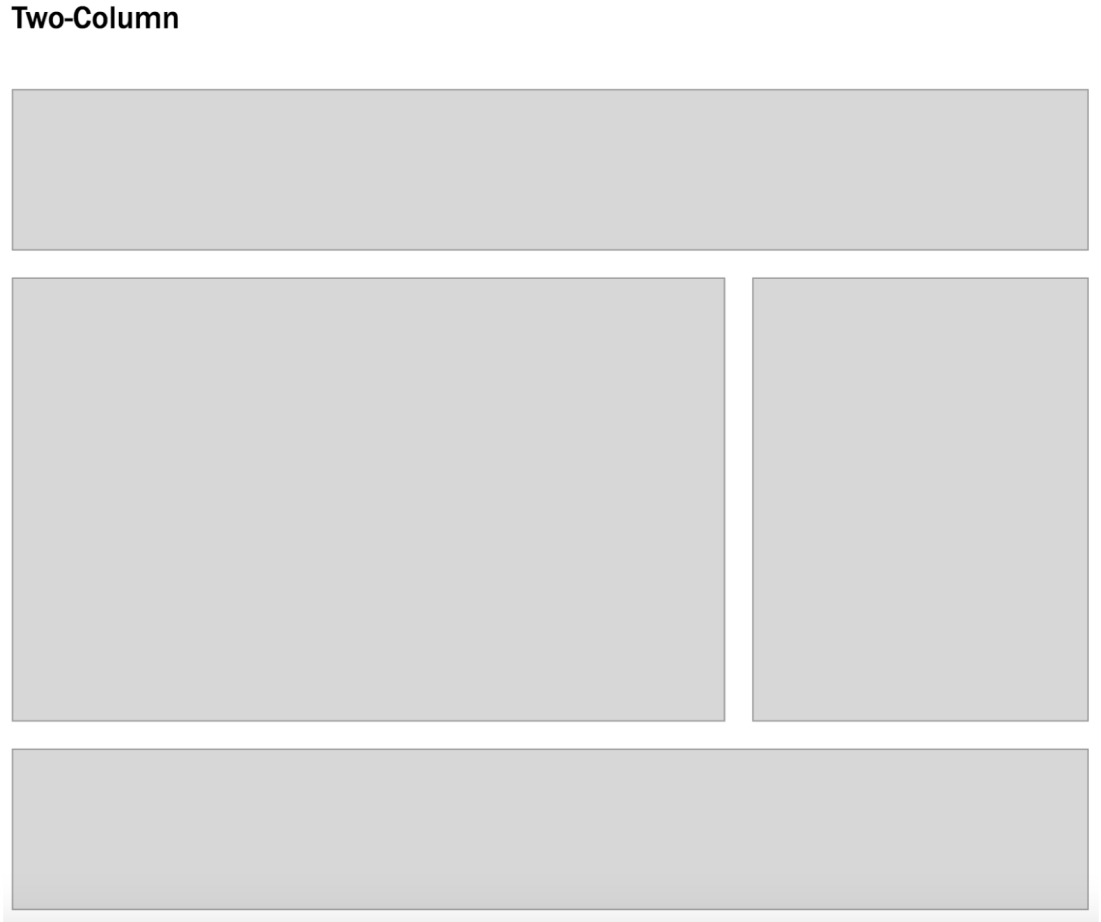
- The clear property accepts the values:
  - **right**
  - **left**
  - **both**
  - none (default)
  - inherit (gets its value from its parent)

Code Along

- Understanding floats and clears

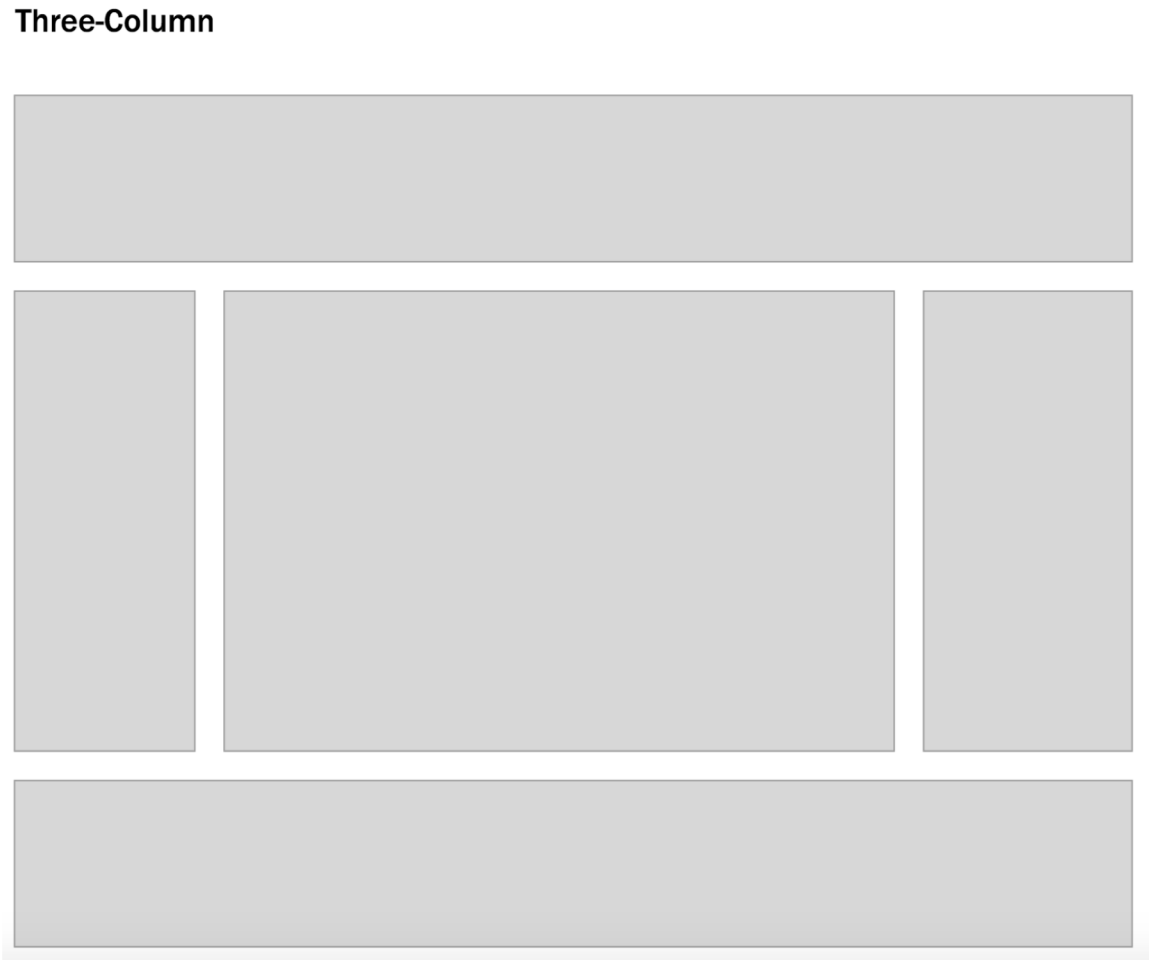
Lab

- Reproduce the follow layout using floats!



Lab

- Reproduce the follow layout using floats!



Centering Things Horizontally

- When placed on the parent element, works for **inline elements**:

```
text-align: center;
```

- When placed on the element, setting the left and right margins to auto works for **block elements**:

```
margin-left: auto;
margin-right: auto;
```

Centering Things Vertically

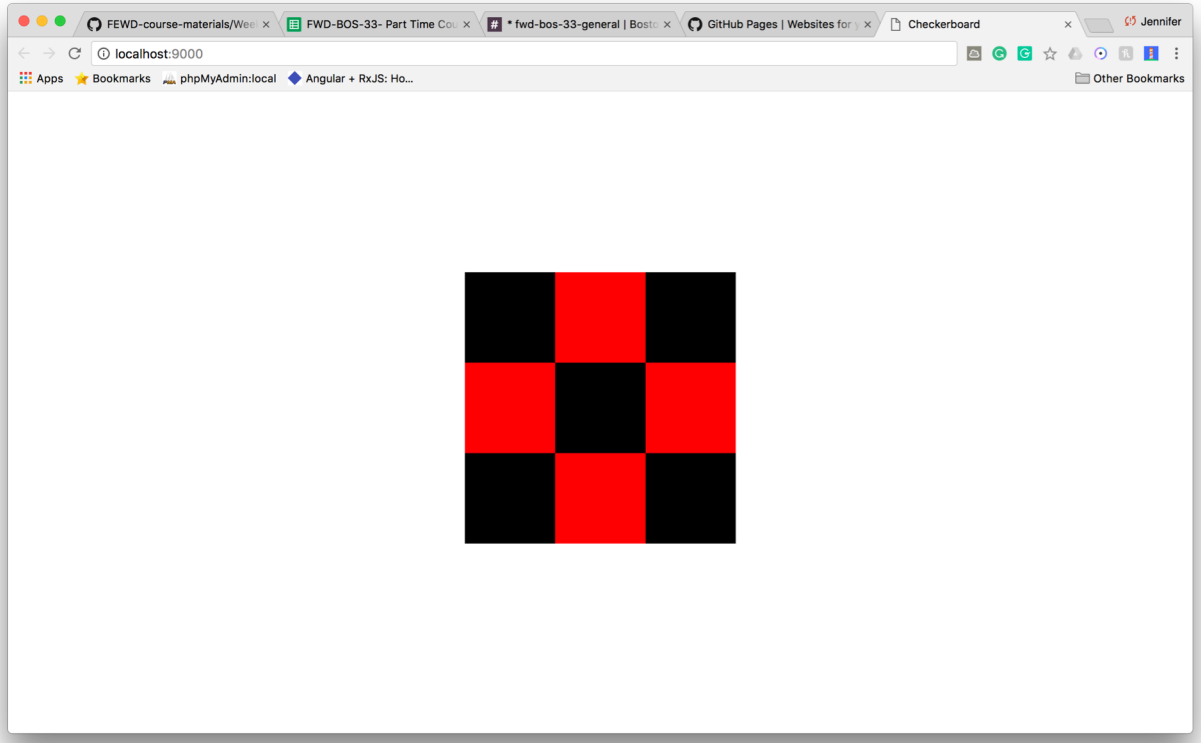
- If you know the **height** of the element you want to center, use margin or padding:

```
margin-top: calc(50vh - 150px);
```

- If you don’t know the height of the element, there are some options:
  - The transform hack…
  - The table cell hack…
  - The ghost element hack…
  - Advanced layout with flexbox YAY!

Lab

- Pulling it together
- Reproduce the following checkerbox design
- It should be 300px x 300px
- Hints:
  - You'll need to use some classes
  - You should center the board on the screen horizontally and vertically
  - You want to use floats
- **BONUS: can you do it with only five black divs on a red board?**



One possible solution

```
<div class="board">
    <div></div>
    <div class="ml"></div>
    <div class="clear ml"></div>
    <div class="clear"></div>
    <div class="ml"></div>
</div>
```

```
.board {
  background-color: red;
  height: 300px;
  width: 300px;
  float: none;
  margin: 0 auto;
  margin-top: calc(50vh - 150px);
}
div {
  background-color: black;
  height: 100px;
  width: 100px;
  float: left;
}
.ml {
  margin-left: 100px;
}
.clear {
  clear: both;
}
```