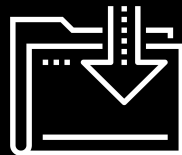




Intro to GitHub Pages and CSS

Data Boot Camp
Lesson 11.2





Activity: HTML Warm-up

In this activity, you will create a simple HTML bio page to serve as a personal info.

Suggested Time:
10 Minutes



Instructions: Activity: Inspect Hello, HTML

On your own, create a simple bio page for yourself using the following HTML elements...

- Header that will store your name inside of it
- An image that will act as a stand-in for your picture with an alt attribute which gives a very basic description of the image
- Two paragraphs that will have text describing who you are
- An unordered list of links that connect to your social media pages
- A table that will contain three columns and some data on your favorite movies, songs, books, or activities
- **Hints:**
 - Dummy images can be found at loremipsum.com and dummy text can be found at loremipsum.com. Focus on getting the entire page working before diving into more specific text and images.
- **Bonus:**
 - Look into how one might go about linking one custom-made HTML page to another. Once you have found out how, try creating a link from your bio page you have just created to one of the pages you created last class.



Time's Up! Let's Review.

CSS: Cascading Style Sheets

Introduction to Basic CSS Styling



- CSS describes how and where elements should appear on the page. It defines things such as color, placement, fonts, sizes, and more.

CSS



HTML/CSS Analogy

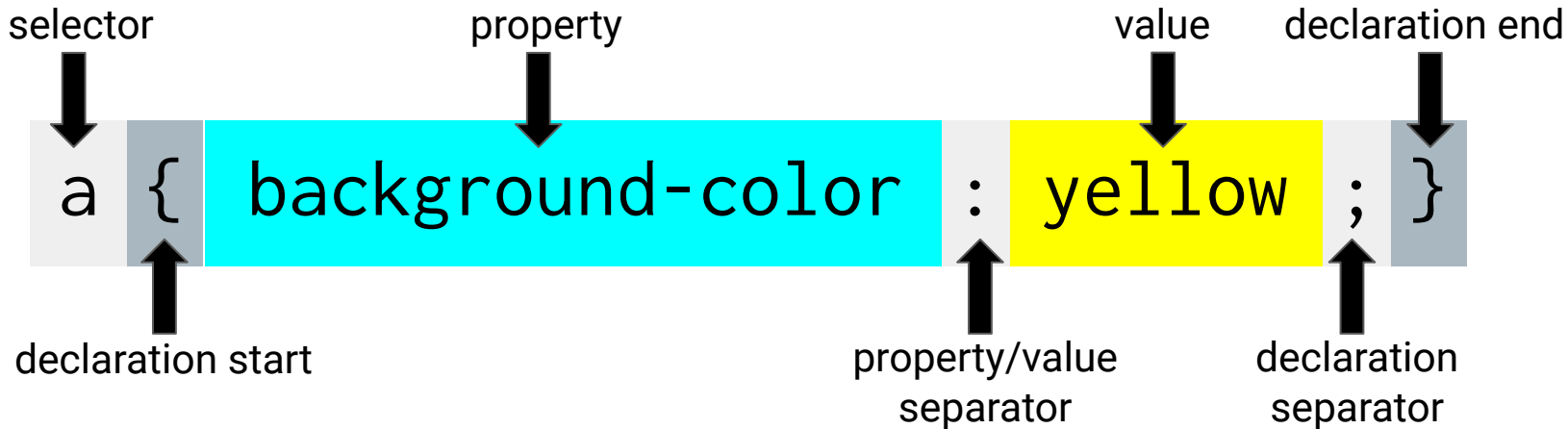
Introduction to Basic CSS Styling

HTML Alone	HTML and CSS
Like writing papers in Notepad.	Like writing papers in Microsoft Word.
Used to write unformatted text (i.e, content only).	Used both to write the content <i>and</i> format it (color, font, alignment, layout, etc.).
	

CSS Syntax

Introduction to Basic CSS Styling

- CSS works by hooking onto **selectors** added into HTML using **classes** and **identifiers**.
- Once hooked, we apply **styles** to those HTML elements using CSS.





Instructor Demonstration

Introduction to Basic CSS Styling

<Time to Code>





Activity: Dull Corp

In this activity, you will be updating the *DULL Corporation's* website so that it is not nearly so...Dull. To do so, you will be creating an external stylesheet and linking it to pre-made HTML.

Suggested Time:
10 Minutes



Instructions: Activity: Dull Corp

- Open the [Unsolved Starter](#).
- Using external CSS and your imagination, update the "DULL Corp" website to be more interesting.
 - Center the `header` element to the page.
 - Set each `h1`, `h2`, and `h3` element to be a different color.
 - Set `img` element to have a shadow.
 - Give the `section` element a background color.
 - Change the font size of both `paragraph` elements.
 - Place a border around the `ul` element.



Time's Up! Let's Review.

HTML class

Classes, IDs, and Divs

- To create an HTML class, place a `class="((className))"`

```
<style type="text/css">
    .bigHeader{
        font-size: 40px;
    }

    .smallerHeader{
        font-size: 20px;
    }

    .tinyHeader{
        font-size: 10px;
    }
</style>
</head>

<body>

    <h1 class="bigHeader">Awesome Header</h1>
    <h1 class="smallerHeader">Smaller Awesome Header</h2>
    <h1 class="tinyHeader">Even Smaller Header</h3>
```

HTML id Classes, IDs, and Divs

- To create an HTML id, place an `id="((idName))"` attribute within an HTML element.

```
#blueText {  
  color: blue;  
}  
  
#redText {  
  color: red;  
}  
  
#purpleText {  
  color: purple;  
}  
  
</style>  
</head>  
  
<body>  
  
  <h1 class="bigHeader" id="blueText">Awesome Header</h1>  
  <h1 class="smallerHeader" id="redText">Smaller Awesome Header</h2>  
  <h1 class="tinyHeader" id="purpleText">Even Smaller Header</h3>
```



Activity: Targeted CSS

In this activity, you will be given a very basic HTML file and will have to create an external CSS stylesheet changes the page's styling.

Suggested Time:
5 Minutes



Instructions:

Activity: Dull Corp

- Open the [Unsolved Starter](#).
- Using an external stylesheet, classes, and IDs accomplish the following...
 - Style the first element so that it has a color of blue
 - Style the second and third elements so that they have a color of purple
 - Style the fourth element so that it has a color of red

Bonus

- Using only additional classes and IDs, attempt to accomplish the following...
 - Set the background color of the first and last elements to black
 - Set the font-size of the first element to 40px
 - Set the font-weight of the second element to bold
 - Set the font-size of the third and last elements to 10px



Time's Up! Let's Review.

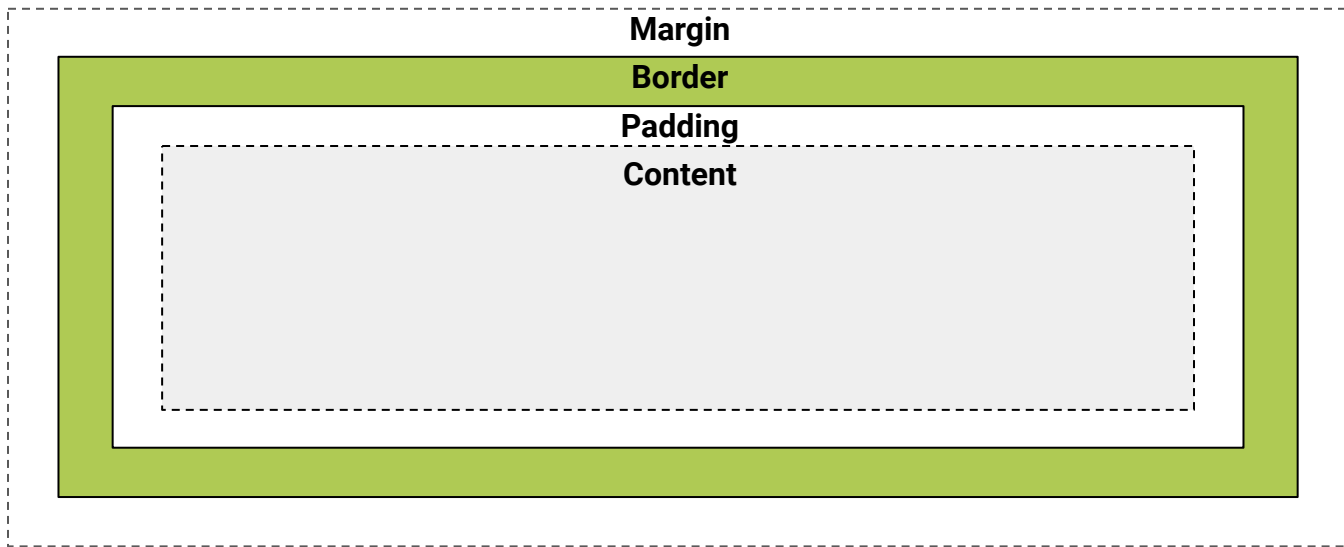


Break

Boxes Upon Boxes

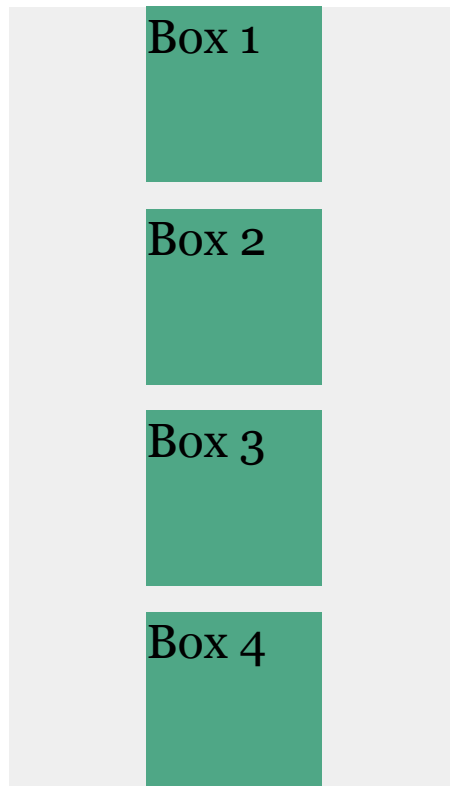
Box Model and CSS Positioning

- In CSS, every elements rests within a series of Boxes.
- Each box has customizable space properties: margin, border and padding.
- Typical spacing value: 20px 10px 10px 20px (top, right, bottom, left).



Position: Static Box Model and CSS Positioning

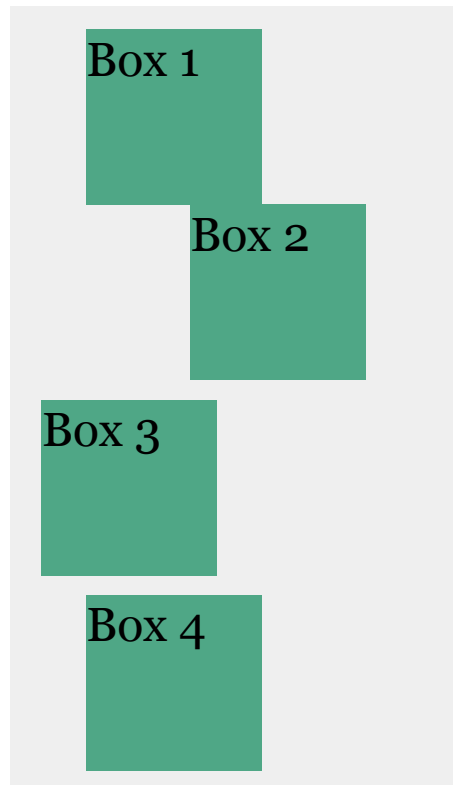
- Four boxes placed statically (default):



Position: Relative Box Model and CSS Positioning

- Switching the boxes to relative will nudge the boxes in relation to their “original” location:

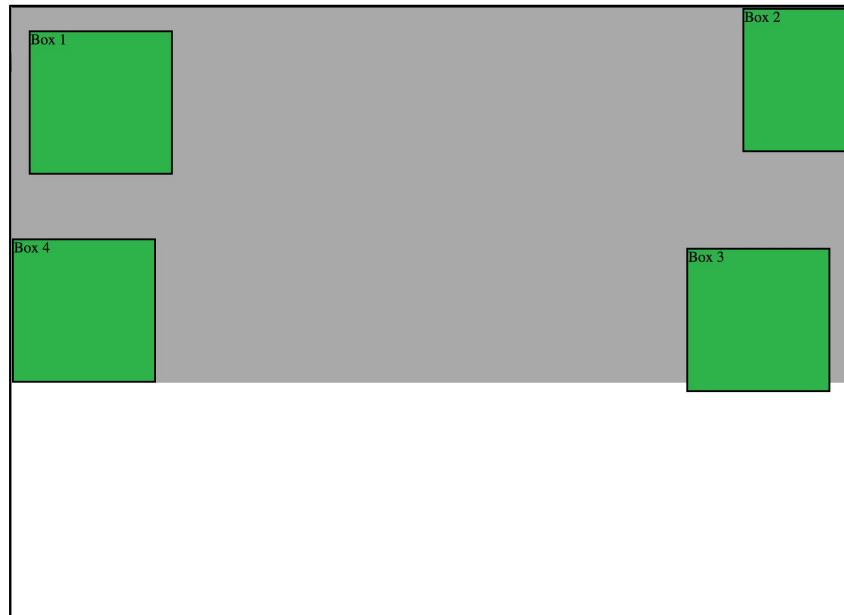
```
.box {  
  background: #2db34a;  
  height: 80px;  
  position: relative;  
  width: 80px;  
}  
.box-1 {  
  top: 20px;  
}  
.box-2 {  
  left: 40px;  
}  
.box-3 {  
  bottom: -10px;  
  right: 20px;  
}
```



Position: Absolute Box Model and CSS Positioning

- Positioned relative to nearest positioned ancestor:

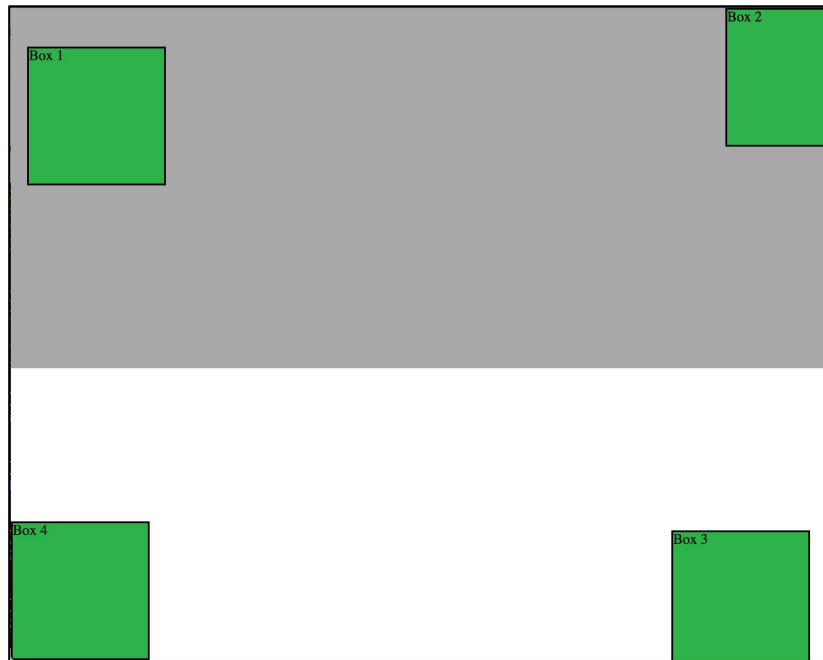
```
.box-set {  
  height: 400px;  
  background: darkgray;  
  position: relative;  
}  
.box {  
  position: absolute;  
  height: 150px;  
  width: 150px;  
  background: #2db34a;  
  border: 2px solid black;  
}  
.box-1 {  
  top: 6%;  
  left: 2%;  
}  
.box-2 {  
  top: 0;  
  right: -40px;  
}  
.box-3 {  
  bottom: -10px;  
  right: 20px;  
}  
.box-4 {  
  bottom: 0;  
}
```



Position: Fixed Box Model and CSS Positioning

- Position with exact coordinates in the browser window:

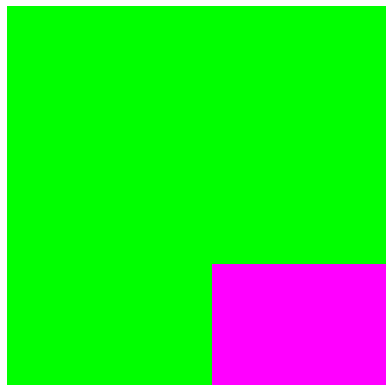
```
.box-set {  
  height: 400px;  
  background: darkgray;  
}  
.box {  
  position: fixed;  
  height: 150px;  
  width: 150px;  
  background: #2db34a;  
  border: 2px solid black;  
}  
.box-1 {  
  top: 6%;  
  left: 2%;  
}  
.box-2 {  
  top: 0;  
  right: -40px;  
}  
.box-3 {  
  bottom: -10px;  
  right: 20px;  
}  
.box-4 {  
  bottom: 0;  
}
```



Layering with Z-Index

Box Model and CSS Positioning

- The z-index property allows you to layer elements on top of each other.



```
position: absolute;  
z-index:1;
```



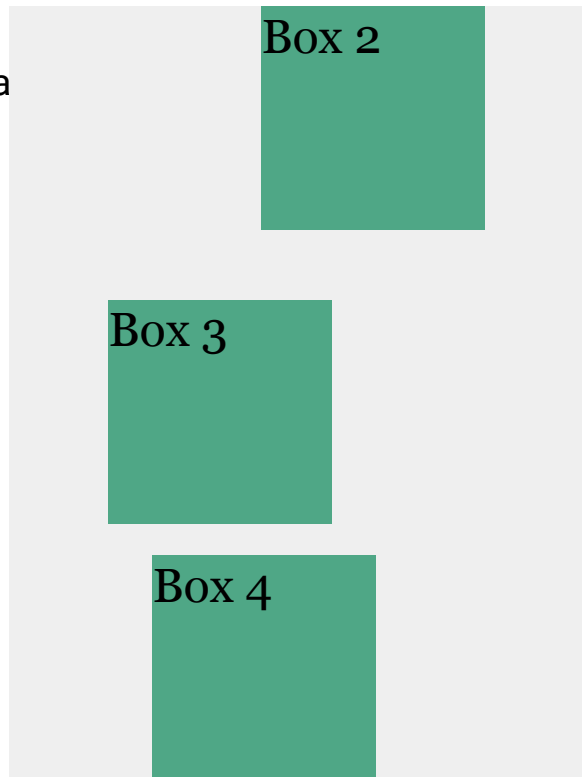
```
position: absolute;  
z-index:2;
```


Hiding things

Box Model and CSS Positioning

- Display: None allows you to hide elements from view.
- This will become useful in later sections, we'll hide and reveal specific HTML elements of our choosing.

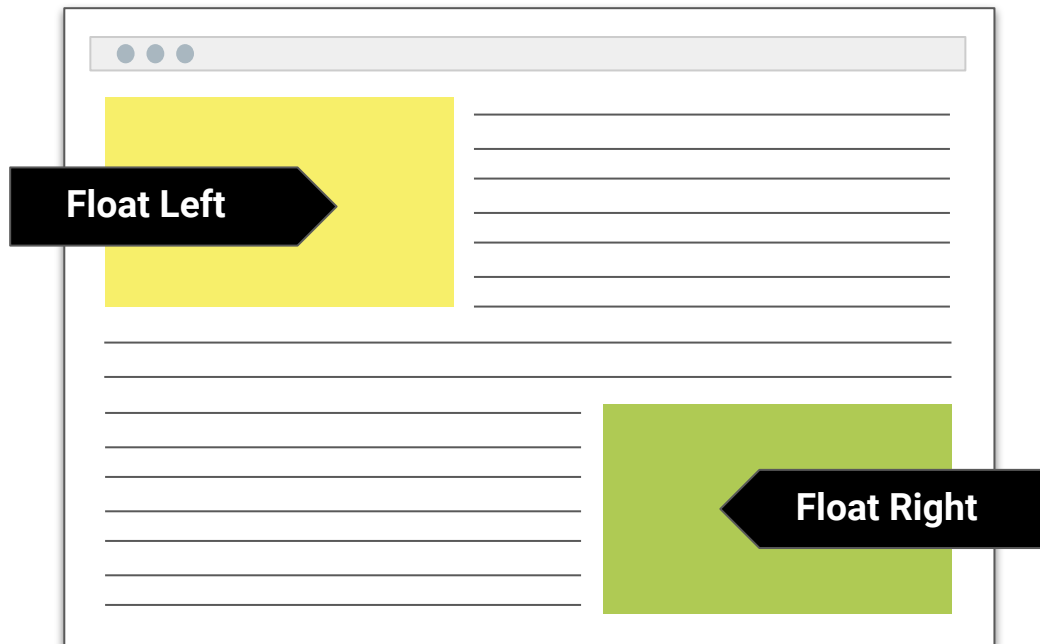
```
.box-1 {  
  display: none;  
}
```



The concept of Flow

Box Model and CSS Positioning

- By default, every HTML element displayed in the browser is governed by a concept called **flow**.
- This means that HTML elements force their adjacent elements to **flow around them**.



Flow analogy to MS Word

- ## wrap-text options



Block Elements

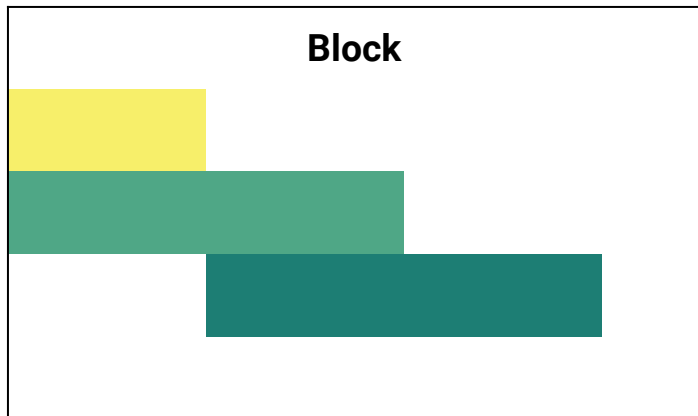
Box Model and CSS Positioning



By default, web clients render many HTML elements as **block elements**. Paragraphs, headers, divs, and more receive this treatment.



A block element will take up an entire line of space—unless you intervene with CSS properties.



Block Elements vs. In-Line Elements

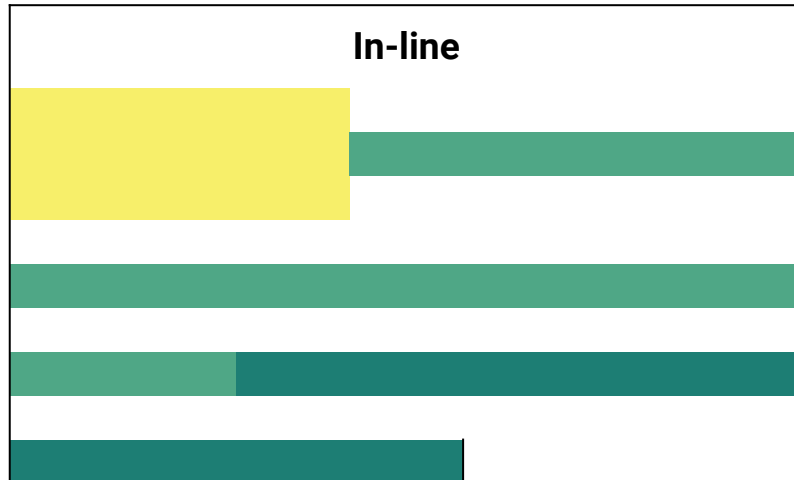
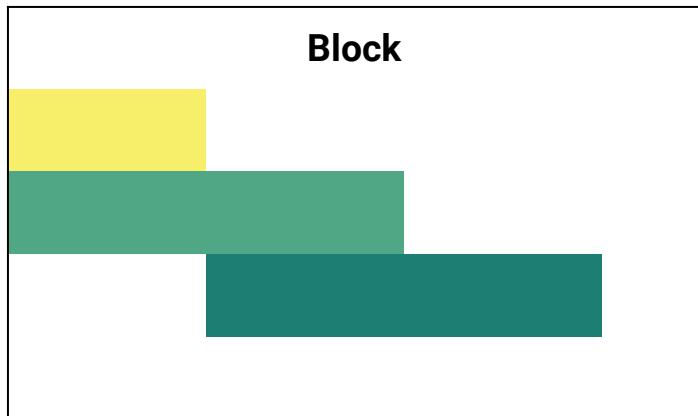
Box Model and CSS Positioning



Now, contrast block elements with **in-line elements**.



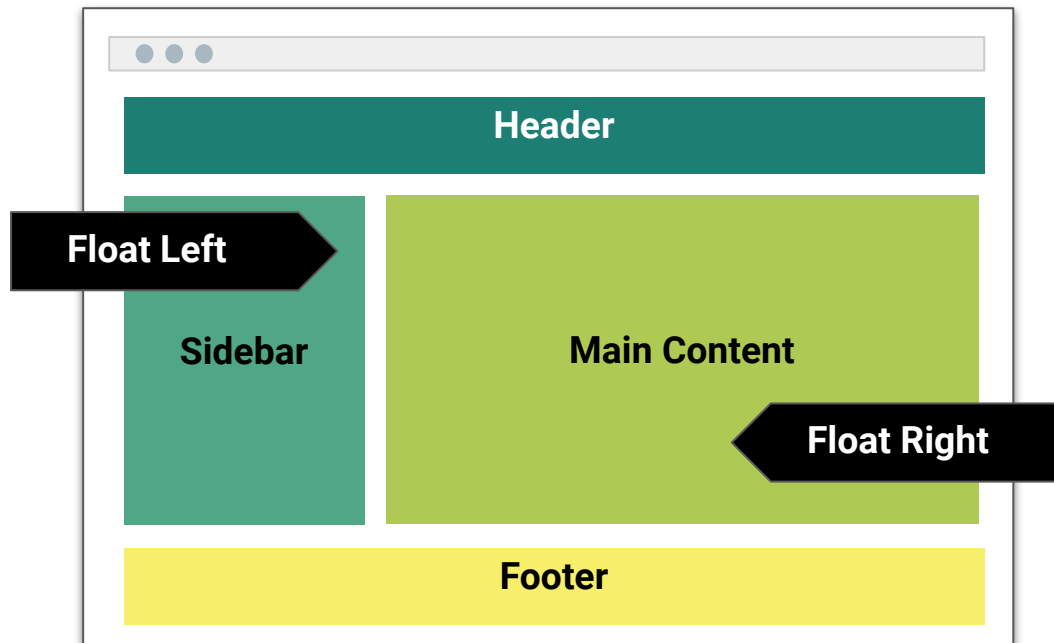
By using **float CSS properties**, we can command our website to display multiple HTML elements adjacently.



Floats

Box Model and CSS Positioning

- To transform these block elements into in-line elements, we use a CSS property called **float**. Floats are necessary when building web layouts.



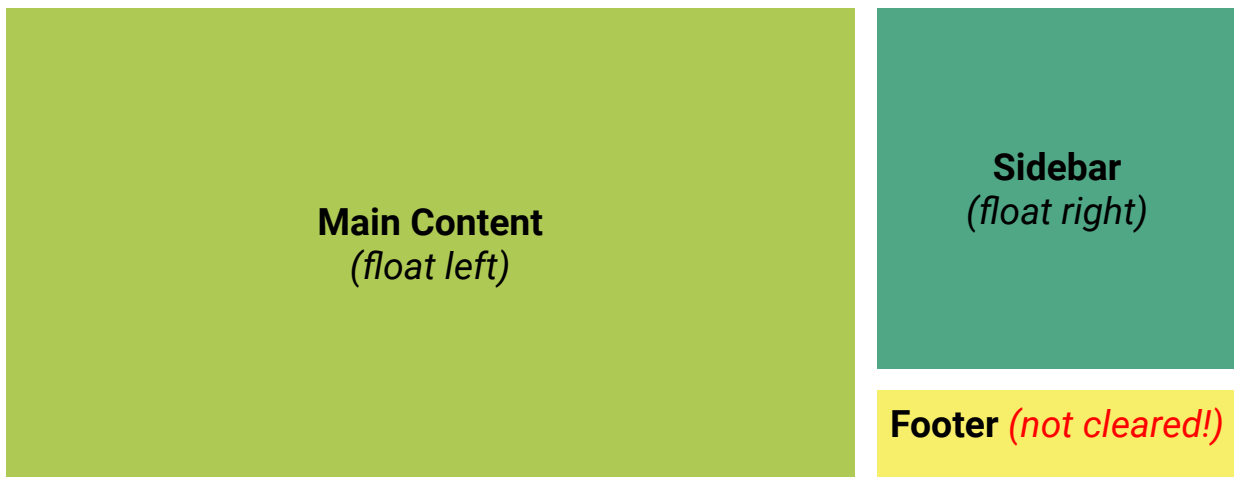
CSS

```
#sidebar {  
    float: left;  
}  
#main-content {  
    float: right;  
}
```

Clearing the Float

Box Model and CSS Positioning

- However, floats often get in the way of layouts. Sometimes we don't want to give each element the “in-line” treatment.



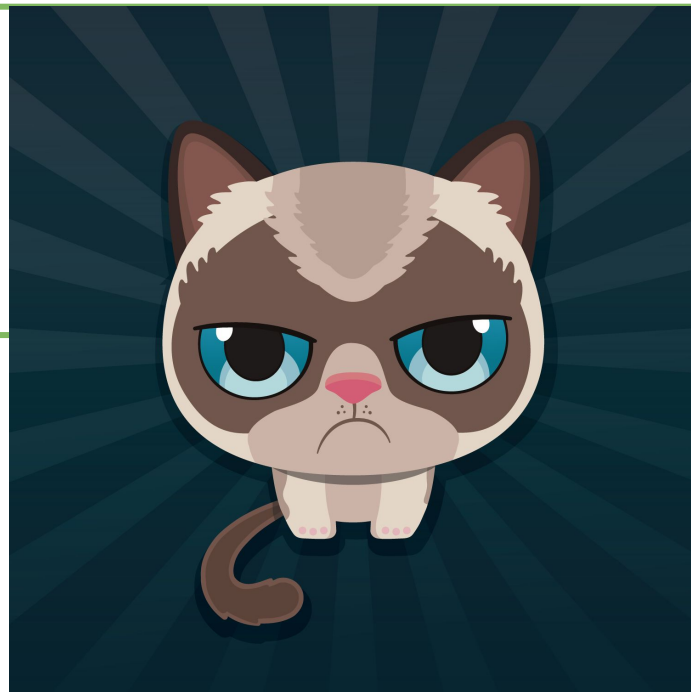
Clearfix Hack

Box Model and CSS Positioning

- Sometimes when elements don't match up in size, we get situations like this:

```
<div>
```

Uh-oh! The image is taller than the element containing it, and it's floated, so it's overflowing outside of its container!



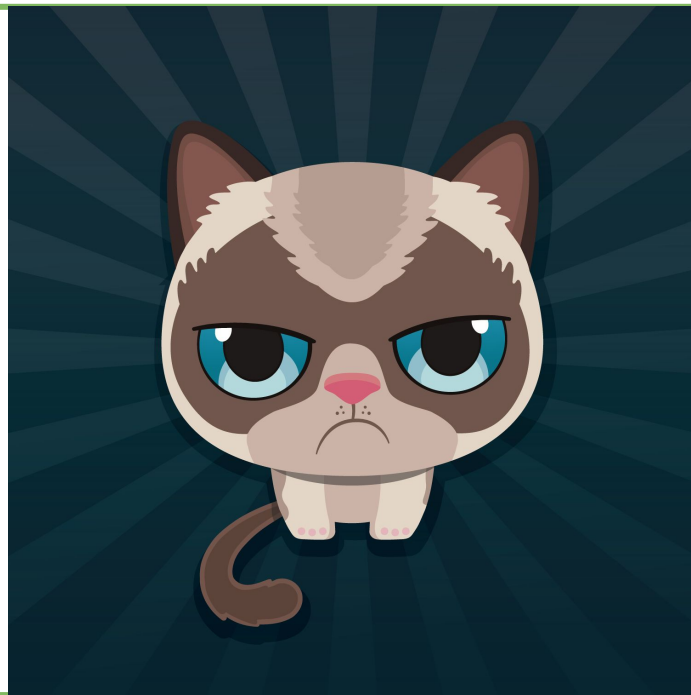
Clearfix Hack

Box Model and CSS Positioning

- We can get around this by using the **clearfix hack**.

```
<div class="clearfix">
```

Much better!



Clearfix Hack

Box Model and CSS Positioning



`::after` is what we call a pseudo-element. We use it to style specific parts of an element.



This will add an HTML element, hidden from view, after the content of the `.clearfix` element. This clears the float.

```
.clearfix::after {  
  content: "";  
  display: block;  
  clear: both;  
}
```



Activity: Aimed Positioning

In this activity, you will be given an HTML file you will style using CSS. In particular, they will be posting certain elements as described in the instructions.

Suggested Time:
10 Minutes



Instructions: Activity: Aimed Positioning

- Open the [Unsolved Starter](#).
- Using CSS, position the five headers in the starter code in their described locations.
 - Try to accomplish this using the box model
 - Try to accomplish this using absolute positioning
- Whenever you complete a step, save a unique version of your stylesheet and then create a new stylesheet for the next step.
- **Hints:**
 - You can move elements around the page using pixels OR percentages. Try out both to see how they work.
 - Not every task is capable of being accomplished without changing the order of HTML elements. Even then, some positions may still be impossible.
- **Bonus:**
 - Try to move the HTML elements provided using different kinds of positioning (static, relative, and fixed).





Time's Up! Let's Review.



Activity: Aimed Positioning

In this activity, you will be given a Bio Page HTML skeleton and it will style is with CSS si the HTML resembles the image provided in the unsolved folder.

Suggested Time:
20 Minutes



Instructions: Activity: Aimed Positioning

- Open the [Unsolved](#) folder and use the files inside as a starting point.
- Style the HTML file with the following:
 - Add a class called "container" on the div tag.
 - Add an id called "main-bio" for the first section tag.
 - Add an id called "contact-info" for the second section tag.
 - Add an id called "bio-image" for the bio image.
 - Style specs:
 - i. `body`
 1. The background color is `#efeee7`.
 2. The font used `"Georgia"`, `Times New Roman`, `Times`, `serif`;
 3. The font color is `#333333`.
 4. Be sure to zero out the body margins and padding so the page is flush to the top of the page.
 - ii. `header`
 1. The background-color is `#333333`.

Instructions:

Activity: Aimed Positioning

- `h1`
 - The font color is #eee.
 - The font size is 28px.
 - Look at the example on the screen, and eyeball the padding and/or margins and positioning of the text.

Instructions:

Activity: Aimed Positioning

- `h2`
 - The font size is 24px
- Make the container have a width of 1024 pixels, and center it. You do this using `margin: 0 auto;`.
- Make `#main-bio`, `#contact-info`, `#bio-image` all `float: left`.
- Make the `#bio-image` have a width of 200 pixels.
- Be sure to include `alt` text in all images
- `#main-bio` should have a width of 70%.
- Add margins to the image so there is distance between it and the bio text.
- `#content-info` should have a width of 30%.
 - Adjust the line height so it is 1.5 times the size of the font.
- Make the link color `#d21034`.
- `#figure` should be given relative positioning
- `#bio-name` should use absolute positioning and be placed at the bottom-center of your `#bio-image`

Instructions: Activity: Aimed Positioning

- Make sure to replace the content inside of the HTML document with your own name, github link, etc.
- Stage, commit, and push this new file to GitHub Pages.
- Bonus:
 - Using fixed positioning, attempt to create a footer on your page that will stay positioned at the bottom of the screen even when scrolling.
 - Underneath the main content of your page, add in a "portfolio" section. Try to make it so that the projects are set up in a kind of grid with two projects per row. Each of the projects should be named with a header, include a relevant image, and have a short paragraph describing the project.



Time's Up! Let's Review.

*The
End*