Lab 3: Bootstrap - Study Buddy

In this lab, you will learn basic interactions with the Bootstrap framework. Starter code provided includes an undecorated HTML file, awaiting your renovation. We will walk you through the basics of Bootstrap, including importing fonts & icons, using buttons, positioning elements, and last but not least advanced CSS tricks such as media query.

Table of Contents

Helpful tools:

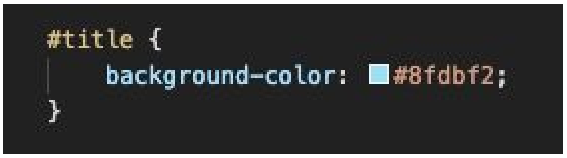
- How to create a navbar: https://getbootstrap.com/docs/4.5/components/navbar/

Lab Video 1 (Section 1a)

1. Download the starter files and unzip them. After this, you should have a folder called studyBuddyFiles-LAB starter.
    a. **At this point, rename index.html to LastnameLab3.html (You'll use your actual last name). If you fail to rename your file, then you'll very likely run into trouble later on when trying to look at your account website.**
2. In your HTML file, you'll need to copy the navbar from Lecture 14 and put it underneath the section that's commented as "Nav Bar".
3. Open your HTML file in a browser (e.g., Google Chrome), and ensure that the navbar turns into a hamburger menu when shrinking the width of the browser window.
    a. Note that you could also use the Atom Live Server package for this step instead of a browser.
4. As an experiment, comment out the last 3 <script> elements, and see if the navbar hamburger menu works when you click on it. You should notice that without these 3 lines of code, nothing will happen when you click on the hamburger menu. This is because Bootstrap uses JavaScript to add functionality to many of their components. Go ahead and uncomment those 3 <script> elements, since we'll want the functionality that the JavaScript provides for us.
    a. Read about comments here if you don't know what they are: https://www.w3schools.com/html/html_comments.asp

Lab Video 2 (Section 1b)

1. Go through your HTML file and notice all of the sections in it. You'll see there's sections for the title, features, testimonials, pricing, and call to action. In order to more easily differentiate the title section (The section we're going to work on soon), we'll apply a background color to it. To do this, go to styles.css and target the title section using the id selector. The id selector uses the # symbol in front of an id in order to target the element with that id. You can read more about the id selector here: https://www.w3schools.com/css/css_selectors.asp#:~:text=The%20CSS%20id%20Selector,the%20id%20of%20the%20element. Once you've done this, give it a background color of #8fdbf2. By the end of this, you should have this in your styles.css:

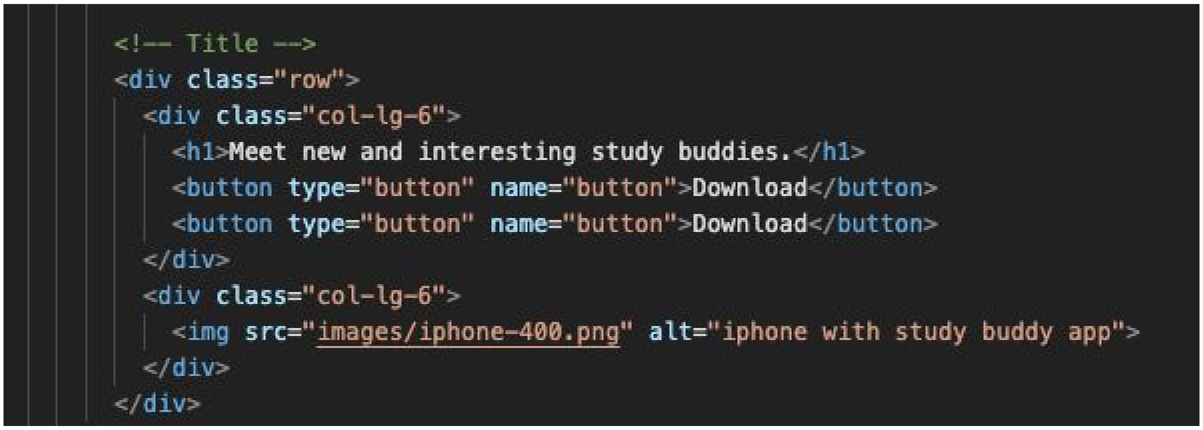


```
#title {
    background-color: #8fdbf2;
}
```

2. You might've noticed that the background color hasn't changed. This is because we haven't linked our stylesheet yet. In order to do this, put
<link rel="stylesheet" href="css/styles.css">
right below the other <link> element for the Bootstrap CSS. This order is important, because we want the Bootstrap styles to be applied first, and then our custom styles to be applied afterwards. Switching the order could cause the Bootstrap styles to overwrite

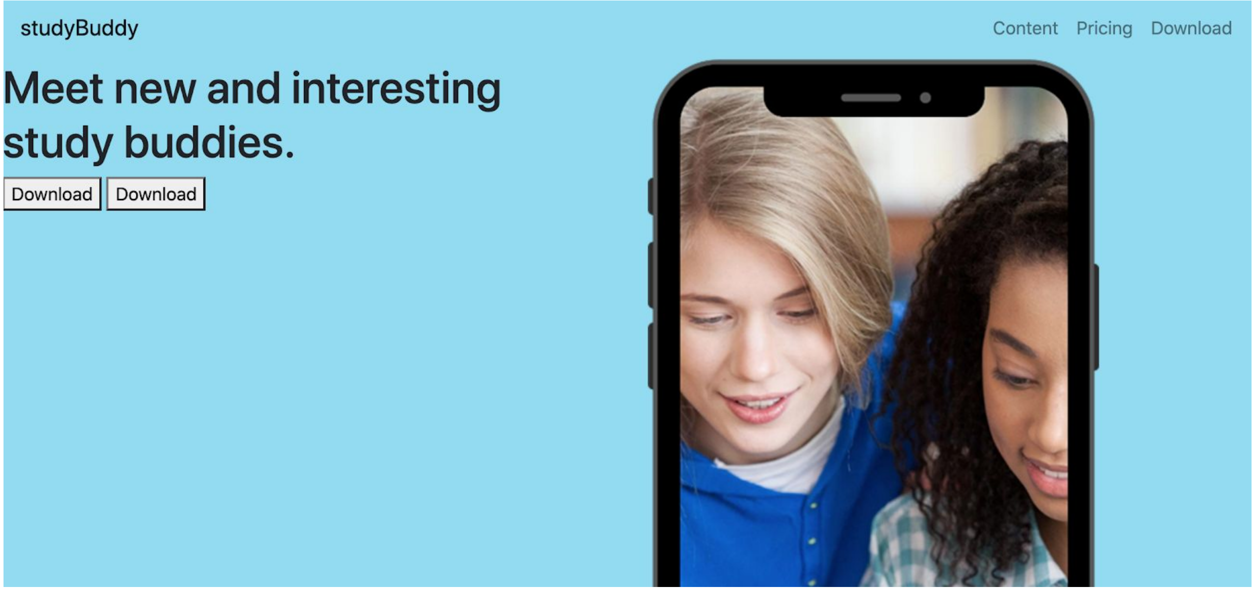
our custom styles. After this, the background color should appear in the title section of the webpage.

3. In the class of your nav element, get rid of the “bg-dark” class and change the “navbar-dark” class to “navbar-light”. This should make your navbar the same color as the background of the title section.
4. Now we want to separate the title and download buttons from the image of the phone in the title section. We’ll have the title and download buttons take up the first 50% of the screen, and the image of the phone will take up the last 50% of the screen. We want this to be the case only if the screen is a laptop size or larger. To do this, we’ll make use of Bootstrap’s grid system. Go ahead and see if you can figure out how to implement this.
5. After you’ve given it a try, check below to see that your solution looks something like this. We first enclosed the entire section in a div with a class of “row” to specify that this div will be holding columns. After, we put the title and download buttons in a div with a class of “col-lg-6”. Since a row in Bootstrap is made up of 12 columns, then we know that using a div with 6 columns will give us what we wanted (The first 50% of the screen to have the title and download buttons). Similarly, we’ll put the phone img in a div with a class of “col-lg-6”, and this should take up the last 50% of the screen. Finally, by using “lg” in “col-lg-6”, we’ve said that this styling will only apply on screens that are large and above (i.e., extra large). Anything below large (i.e., medium, small, and extra small), will not have this styling applied to it.



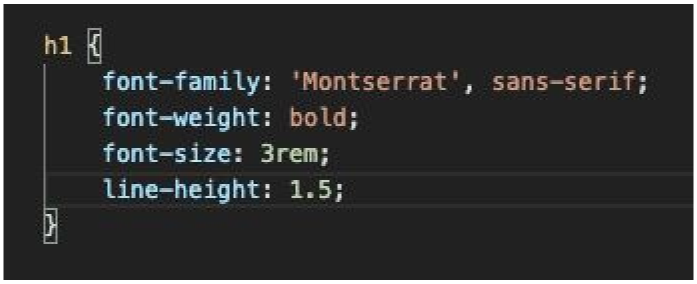
```html
<!-- Title -->
<div class="row">
  <div class="col-lg-6">
    <h1>Meet new and interesting study buddies.</h1>
    <button type="button" name="button">Download</button>
    <button type="button" name="button">Download</button>
  </div>
  <div class="col-lg-6">
    <img src="images/iphone-400.png" alt="iphone with study buddy app">
  </div>
</div>
```

6. At this point, check that your screen matches the image below to ensure that you’re on the right track.



Meet new and interesting study buddies.

7. From here, we’ll get some new fonts using Google Fonts.
8. Go to Google Fonts (search for it on Google)
9. Search for Ubuntu and then click on it to display all of the styles
10. Click “Select this style” for the Light 300 style
11. Go to the top of the page and click “Browse fonts”
12. Search for Montserrat and then click on it to display all of the styles
13. Click “Select this style” for the Regular 400 style
14. On the right hand side of the page, there should be a “Selected Families” section with an “Embed option underneath it. Copy the <link> element and paste it in the <head>. By doing this, you should be able to access both fonts now.
15. In styles.css, add the css shown below for <h1>



```css
h1 {
    font-family: 'Montserrat', sans-serif;
    font-weight: bold;
    font-size: 3rem;
    line-height: 1.5;
}
```

Lab Video 3 (Section 2a)

Containers:

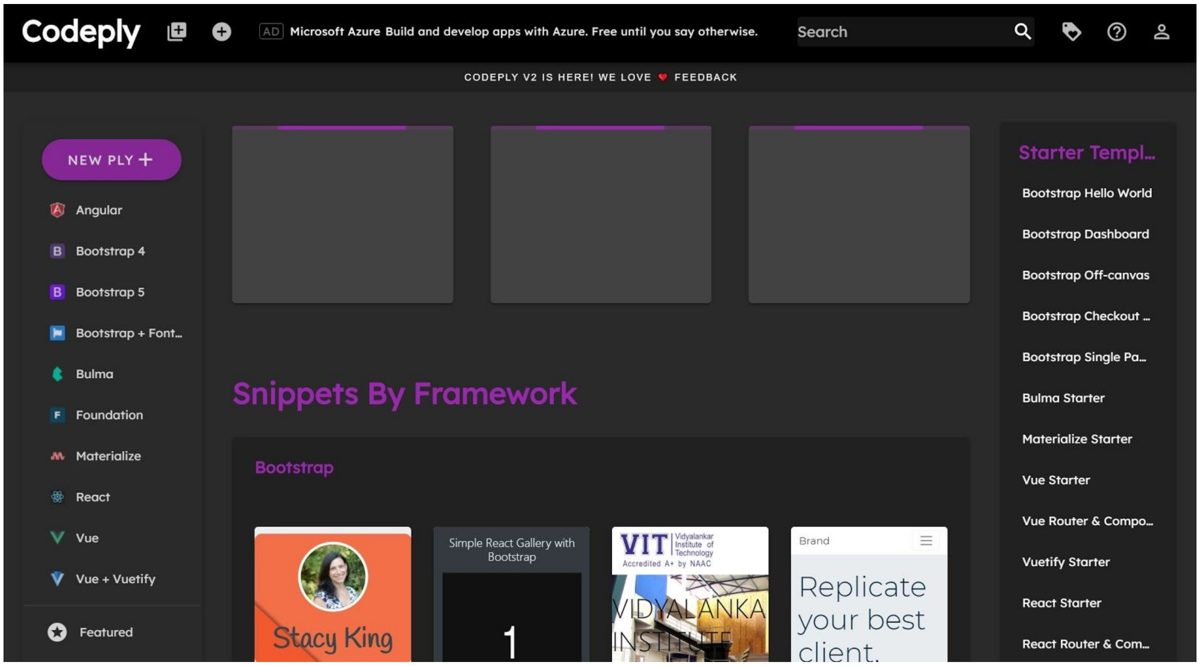
They are basic building blocks for many components in Bootstrap and It is the most basic layout element when using a grid system. It contains the **row** elements and the **row** elements are the **container** of columns (known as grid system).

Container is mainly used to set the content's margins dealing **with** the responsive behaviors of your layout. The **container** class is used to create boxed content.

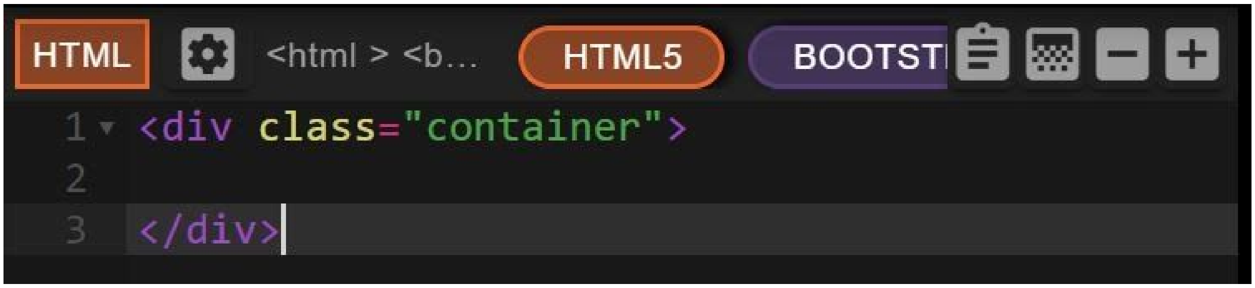
There are two types of containers that you can use based on the choice of width:

- fixed-width container (meaning its max-width changes at each breakpoint) or
- fluid-width (meaning it’s 100% wide all the time).

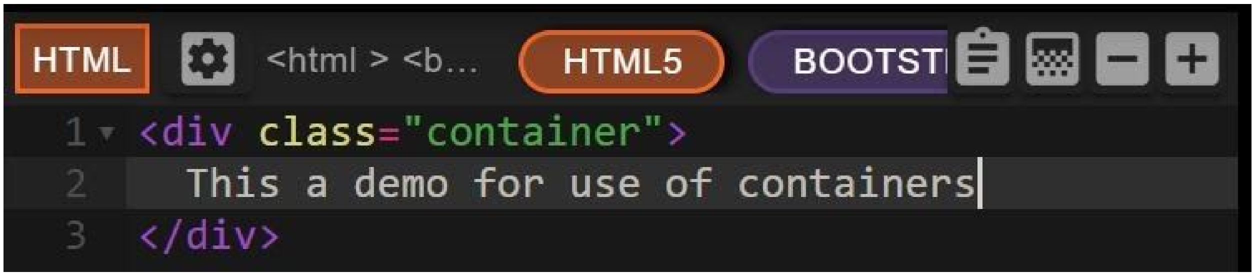
Let us see how container class is implemented with an example.
Note: You can use codeply from COGS 3 homepage to initially test your code before using it on your text editor (atom). The snippets attached below are from codeply.



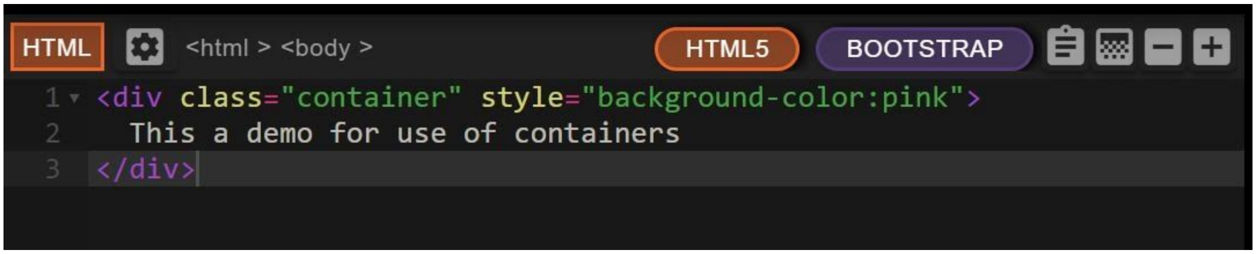
Here is the codeply website. You can start implementing containers by clicking on **Bootstrap 4** on the left side of the screen.



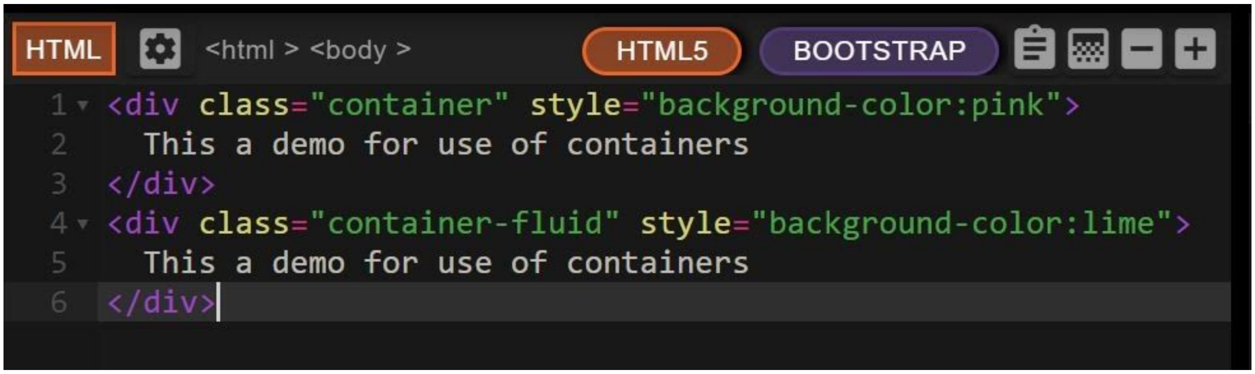
```html
<div class="container">

</div>
```



```html
<div class="container">
  This a demo for use of containers
</div>
```

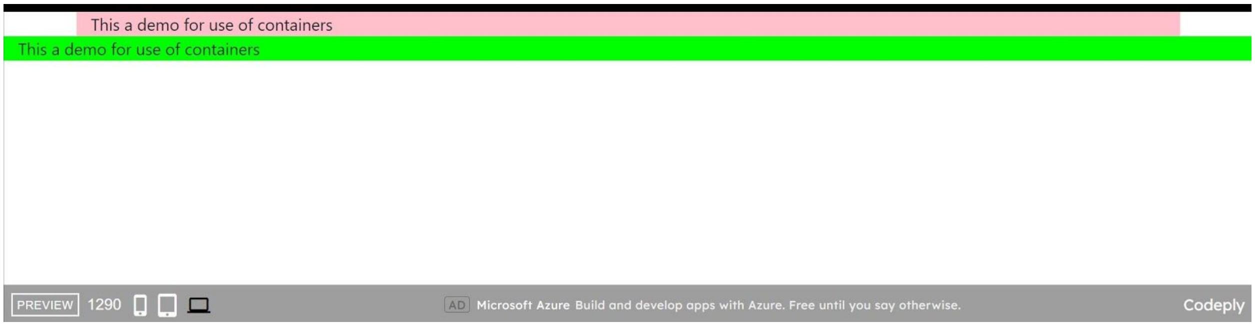


```html
<div class="container" style="background-color:pink">
  This a demo for use of containers
</div>
```



```html
<div class="container" style="background-color:pink">
  This a demo for use of containers
</div>
<div class="container-fluid" style="background-color:lime">
  This a demo for use of containers
</div>
```

This snippet shows use of the two types of containers and below is the preview of how each container affects your layout. You can see that the container class having a pink background has some padding on the left and right whereas the container-fluid class spans the entire width of the layout.

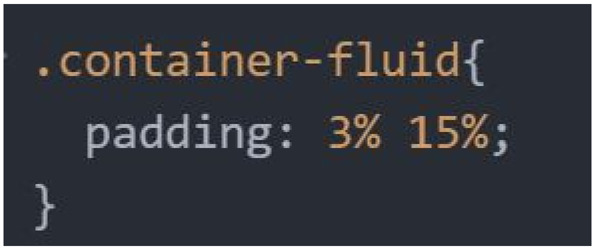


```
This a demo for use of containers
This a demo for use of containers
```

PREVIEW
1290
AD Microsoft Azure Build and develop apps with Azure. Free until you say otherwise.
Codeply

When using the container class for your webpage design, all you have to do is wrap the <div class= “container”>.....</div> or <div class=”container-fluid”>......</div> around the grid elements of your layout (in our case- around the navigation bar and the image)
After adding this class to your layout, you can see that the grid elements collapse and rearrange according to the page size.

Stylesheet for Container class:
Although Bootstrap has its default style for Container class, you can edit its style depending on how you want our website to look.

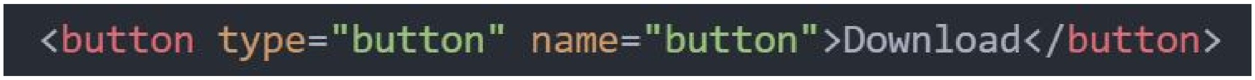


```
.container-fluid{
  padding: 3% 15%;
}
```

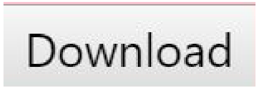
Styling container-fluid class this way adds padding on all four sides (3% padding on the top and bottom; 15% on left and right). This will affect the placement of the grids and ensures that the contents on the page are not too stretched out.

Lab Video 4 (Section 2b)

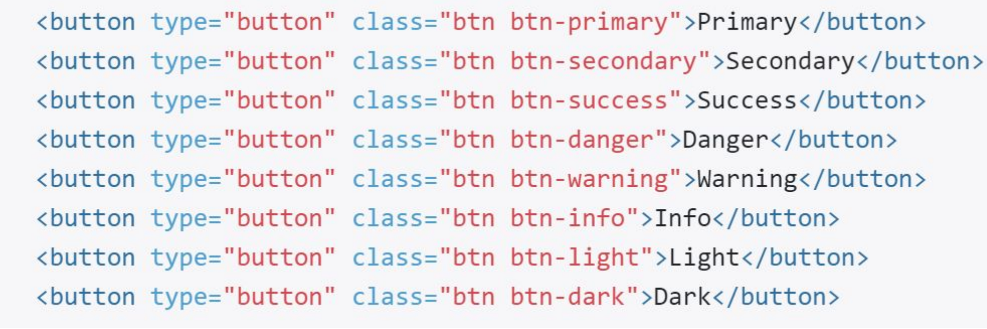
Buttons on Bootstrap:
Bootstrap offers multiple button designs. It is designed using the **<button>** element along with the **.btn** class.
The basic html code for implementing a button looks like this:



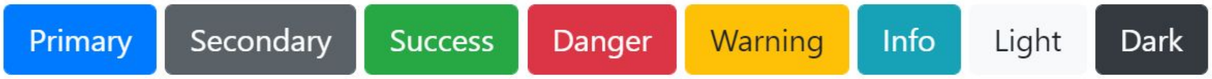
```
<button type="button" name="button">Download</button>
```

The button implemented : 

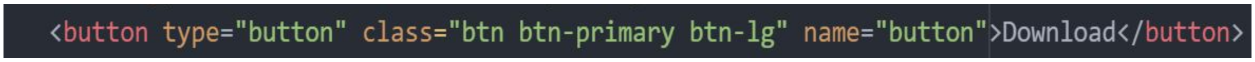
This looks very simple and could be made better with use of **.btn** class. They can be designed with multiple predefined colors (Visit Bootstrap documentation to see all options)



```
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
```



You can also make the button large by adding .btn-lg class

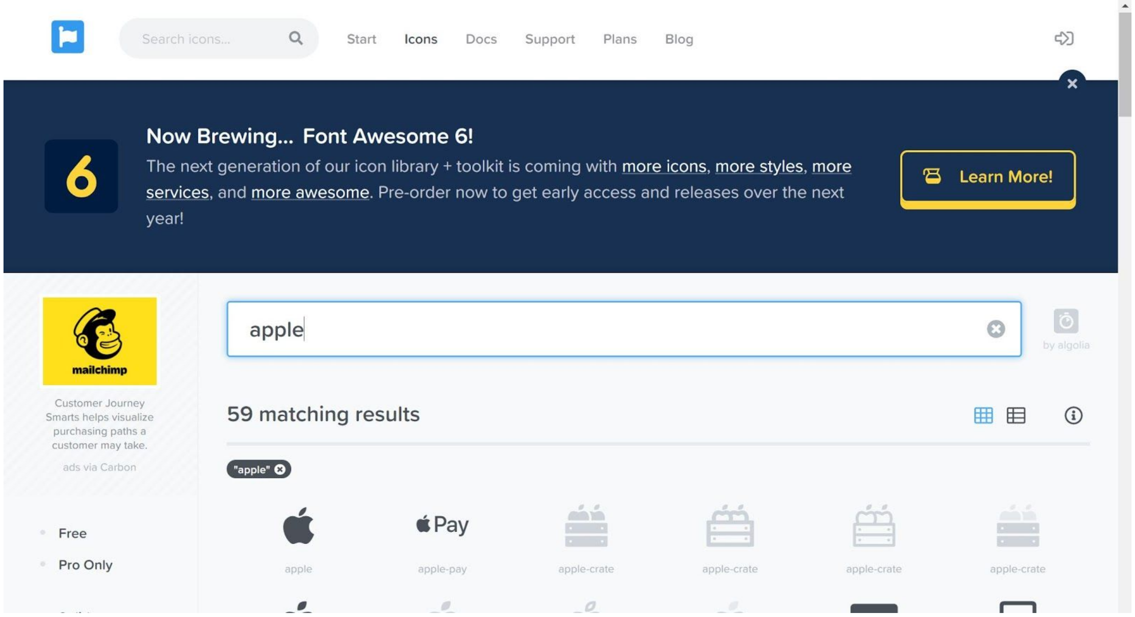


```
<button type="button" class="btn btn-primary btn-lg" name="button">Download</button>
```

FontAwesome Icons:

Font Awesome website is a library for special logos that can be added to your website. It is an open-source website so you can access all the logos for free!

You can add the apple and google play logos to the 'Download' button using FontAwesome. Go to the icons section on the Font Awesome website and search for the apple logo.



Now Brewing... Font Awesome 6!

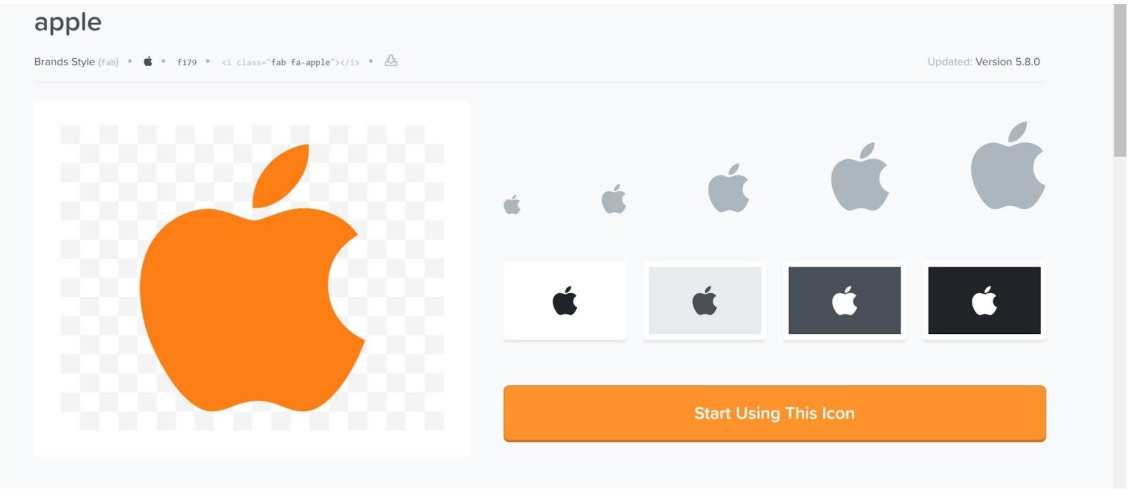
The next generation of our icon library + toolkit is coming with more icons, more styles, more services, and more awesome. Pre-order now to get early access and releases over the next year!

Customer Journey Smarts helps visualize purchasing paths a customer may take.

ads via Carbon

apple

59 matching results

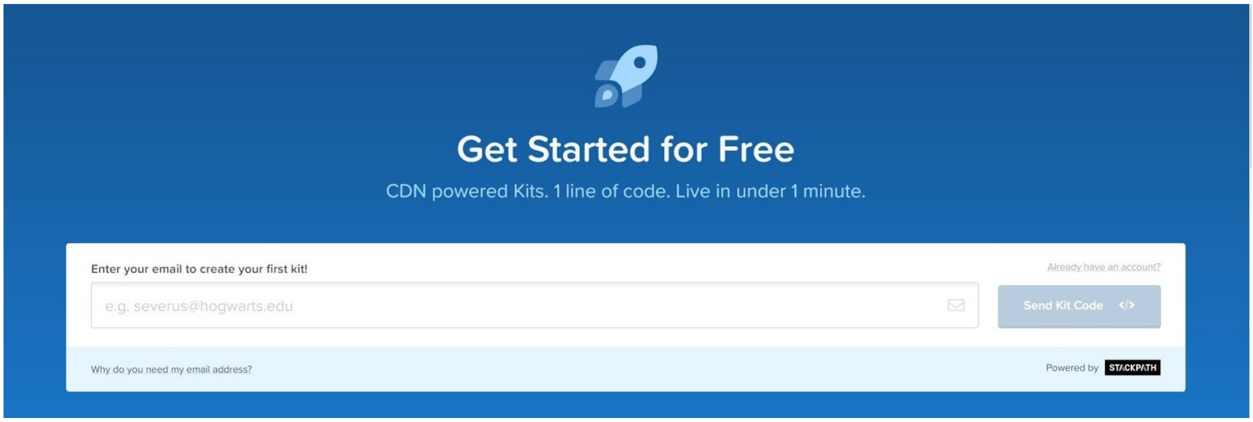


apple

Brands Style (fab) • • f179 • `<i class="fab fa-apple"></i>` •

Updated: Version 5.8.0

You can choose from multiple icon sizes. Once you've chosen your icon, it will take you to the registration page. You would have to register with your email to access these icons.



Get Started for Free

CDN powered Kits. 1 line of code. Live in under 1 minute.

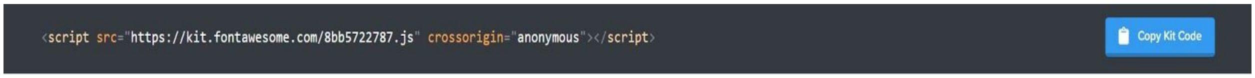
Enter your email to create your first kit!

Already have an account?

e.g. severus@hogwarts.edu

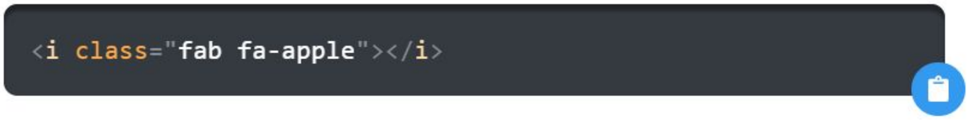
Why do you need my email address?

You will receive an email containing the html script for Font Awesome which should be added to the <head> section of your html document (similar to linking your css stylesheet to your html document). The script will look like this:



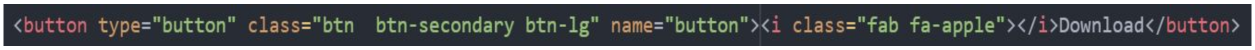
```
<script src="https://kit.fontawesome.com/8bb5722787.js" crossorigin="anonymous"></script>
```

After copying this code to your html document, you can copy the html code for including the apple logo to your download button:



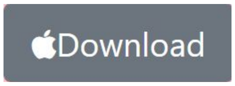
```
<i class="fab fa-apple"></i>
```

Make sure you copy this code to right before "Download"



```
<button type="button" class="btn  btn-secondary btn-lg" name="button"><i class="fab fa-apple"></i>Download</button>
```

After implementing this, your download button will have the apple logo in it.



Lab Video 5 (Sections 2b & 2c)

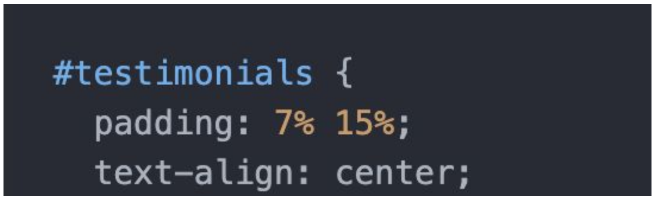
1. In the css sheet, add a new section titled "Navigation Bar"
2. In the html file, locate your class selector of "navbar-brand"
3. Using the selector name above, go back to your css sheet and change font family (under Navigation Bar section) to a desired font (ie. Ubuntu)
4. In your css sheet--underneath the same selector referenced in the above steps--increase your font size (ie. around 2.5rem)
5. If desired, adjust the font weight to be bolded
6. Using the navbar (*not* navbar-brand) class in your css file, attempt to add bottom padding underneath the "studyBuddy" logo in the navigation bar (ie. around 4.5rem)

7. Using nav-item class in your css file, add padding to the top/bottom & left/right (ie. 0 18px)
8. Now, in html file, target class titled “navbar-link” to style these attributes
9. Go back to the css sheet to change the font family, font size and colors for these items. Adjust preferences accordingly
10. Feel free to use Inspect Tool on your studyBuddy website in order to understand where unwanted padding should be removed
11. Following step 10 should allow you to see an additional ~16px that is padded around the perimeters of the studyBuddy element
12. Go into your css sheet and change this padding to 0. Do this by going to your .navbar section and changing “padding-bottom” to “padding”; add 0 to top/left and leave 4.5rem for the bottom
13. In your html, find the Google & Apple download buttons. Attach a class selector titled “download-button” to each appropriate tag
14. Go into your css and create a new section called “Download Buttons”
15. With download-button class, add margins (ie. 5% 3% 5% 0)
16. Go into your css and create another new section called “Title Image”
17. Go into html file to add a custom class selector in the iPhone image tag
18. Using custom selector from step 17, reference that in your css sheet to adjust width
19. Manually google how to rotate an image in css; apply these learned skills into your own css sheet by rotating the iPhone image

Lab Video 6 (Sections 2d & 2e)

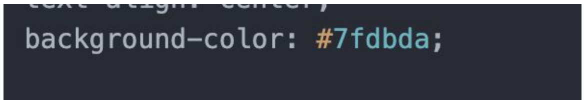
1. In your html file, locate your “Features” section
2. Make sure you implement an outer div tag with class = “row”. Refresh your webpage to ensure there are three columns in this container and that it is responsive.
3. Add padding around this “Features” section. Do this to all four sides by referencing the features section id (use # in css sheet). Keep aesthetic appeal in mind.
4. To center the text in each feature box, go to the css sheet. Target the feature-box class to align this text as “center”
5. Now, add padding to each of these feature boxes using the same class above (ie. 5%).
6. Adjust the font-family and weight of your h3 header to make these features bolder.
7. Next, select the paragraph element to change its color.
8. Find icons using fontawesome.com (heart, target and checkmark)
9. Copy the html of each icon; paste into your html feature box below the div tag.
10. Make these icons bigger by adding the size class (from fontawesome.com) (ie. 3x)
11. Style these icons by creating a unique class (ie. “icon”) to target in your css sheet.
12. In your css features section, create an icon class to change their colors. Keep complementary colors in mind.
13. Add bottom margins to this icon class in css (ie. 1rem)
14. In css still, use icon:hover pseudo class to change the color as you hover over these icons

Lab Video 7 (Sections 2e & 2f)

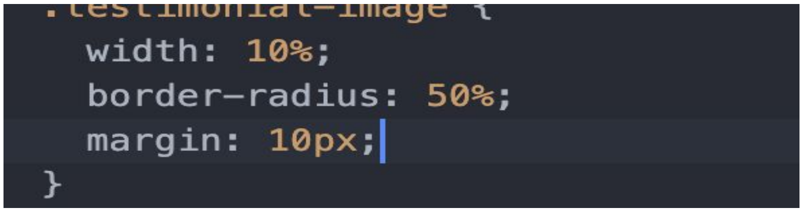
1. Now it’s time to style testimonial section, we are going to learn about carousel
2. In your html file, you are going to be adding to the section called “testimonial”
3. In your CSS sheet, add a new section titled “Testimonial”
4. First, You are going to change the padding of all the testimonials.
    - 7% on the bottom and top, 15% left and right
5. 

```
#testimonials {
  padding: 7% 15%;
  text-align: center;
```

6. And then, aline the text as “center”
7. We have to add background color. Add background color

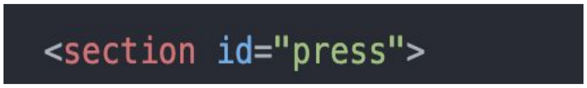


```
text-align: center;
background-color: #7fdbda;
```

8. If you want to change the text color , you need to style “h2” in your css file
9. Add css for h2 to have font Montserrat, for-size of 1.5, font-weight of bold.
10. In your html section, we need to add class “testimonial-image” in image code
11. Go into your css and create a new section called “testimonial-image”
12. And we need to change width and make the image round (using border-radius)
13. Go into your css file to make image round, you need to add border-radius:50%width:10% and margin 10px
14. 

```
.testimonial-image {
  width: 10%;
  border-radius: 50%;
  margin: 10px;
}
```

15. Now we have to press section
16. In your HTML file, you are going to be adding to the section called “Press”
17. First , create a new class named”press”



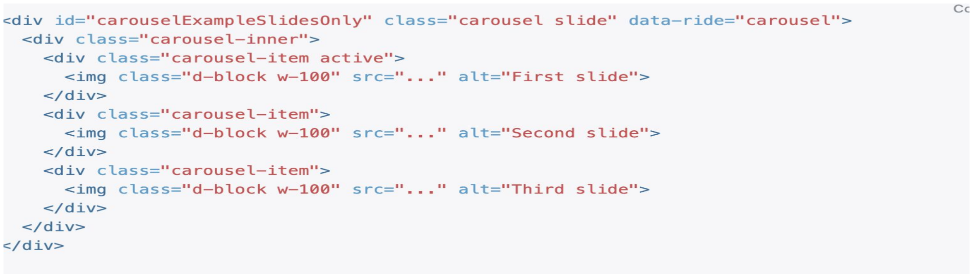
```
<section id="press">
```

18. Add all images associated with press (four images) using image tag
19. Go into your css , create press section (use #)
20. Add background color and aline the text “center”
21. Finally, to target the press logo we need to target it and change width and margin.

Lab Video 8 (Section 2f)

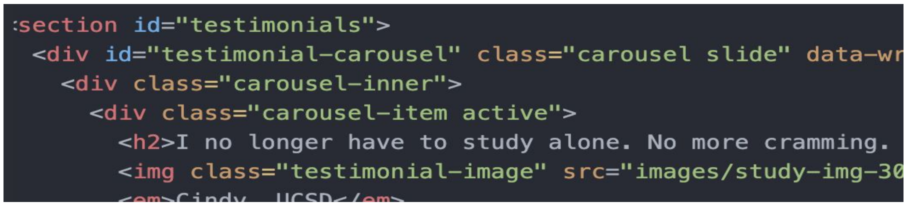
1. We need to style bootstrap carousel
2. Go into this website”https://getbootstrap.com/docs/4.0/components/carousel/” to copy and paste some code for carousel.

3. Copy the code under the slide section on https://getbootstrap.com/docs/4.0/components/carousel/.

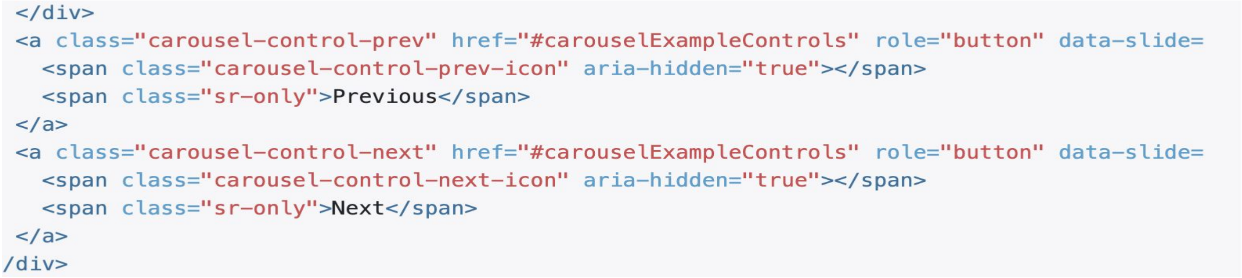
4. 

```
<div id="carouselExampleSlidesOnly" class="carousel slide" data-ride="carousel">
  <div class="carousel-inner">
    <div class="carousel-item active">
      <img class="d-block w-100" src="..." alt="First slide">
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="..." alt="Second slide">
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="..." alt="Third slide">
    </div>
  </div>
</div>
```

5. In the beginning, create a new “div” class named “testimonial -carousel” under testimonial section id. And then add testimonial- image!

6. 

```
:section id="testimonials">
  <div id="testimonial-carousel" class="carousel slide" data-wr
    <div class="carousel-inner">
      <div class="carousel-item active">
        <h2>I no longer have to study alone. No more cramming.
        <img class="testimonial-image" src="images/study-img-30
        <em>Cindy, UCSD</em>
```

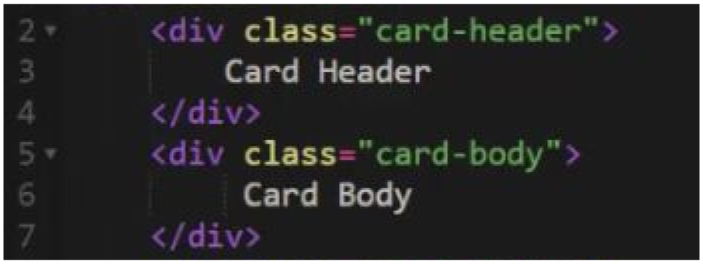
7. And inside “testimonial-carousel” div, you are going to create new two div class named “carousel-inner” and “carousel-item active”
8. Now, we need to add carousel with control
9. Copy the carousel code with control (See image below)
10. 

```
  </div>
  <a class="carousel-control-prev" href="#carouselExampleControls" role="button" data-slide=
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleControls" role="button" data-slide=
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
/div>
```

11. And paste into your html and and change “href = “carouselExampleControls” to href="#testimonial-carousel"
12. Remove background color (red, greed etc…) to fit studybuddy website color.
13. Go into your css sheet and change carousel-item padding.
    - 7% on the bottom/ top , 15% left and right

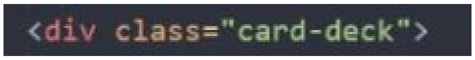
Lab Video 9 (Section 2g)

1. In your HTML file, you are going to be adding to the section called “Pricing”
2. Now, to begin feel free to go to the bootstrap or bootsnip to look at some examples
3. To start, create a new div class named “card” under the section id.
4. And inside that div, we will be adding a class of card header and card body. Now, save and refresh your page. To better visualize your card, you can add text under each class like so:



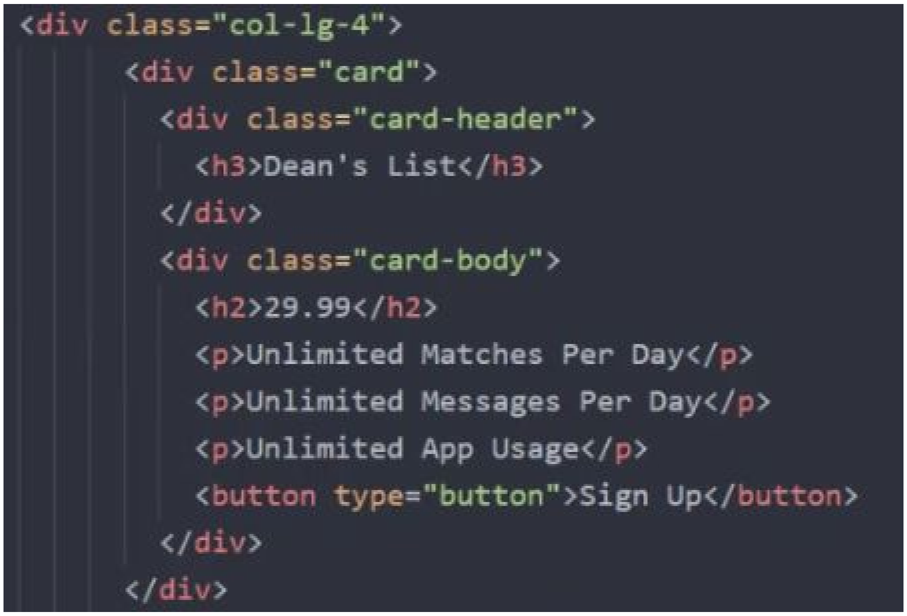
```
    <div class="card-header">
        Card Header
    </div>
    <div class="card-body">
         Card Body
    </div>
```

5. Copy your first card and paste it two more times as we will need a card deck of three different pricing options.
6. Now, we will use the headers and information provided for each of the pricing tiers. Move the <h3> headers such as “Honors”, “Dean’s List”, and “Magna Cum Laude Honors” under the card header class.
7. And move the <h2> and <p> tags of each pricing tier under the card body. You can now hit save and refresh your webpage to see your progress.
8. As you can see, the pricing cards are currently taking up the entire page length. To change that to make them stack side by side, you will need to use the div class “card deck” which can be found when you inspect the card deck examples on bootstrap or bootstrap. Make sure that the div class surrounds all three pricing cards that you have created. Now, beautify, save and refresh your page to see the updated website.



```
<div class="card-deck">
```

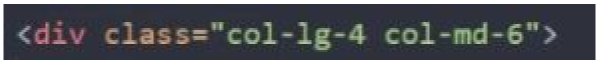
9. Now that you have your card deck with all three pricing tiers, we want to make sure that the deck is at 100% for mobile and ipad users.
10. We need to start by creating a div class “row” that encapsulates our card deck.
11. Now we will wrap theache card deck with a div class called “col-lg 4” Because by default, when we are using the largest sizes (desktops), we want each card deck to take up the 4 units of the grid. We want this behavior to fit with all three cards, so we will add it to each pricing card. So now each of the pricing tiers should have a similar card like this:



```
<div class="col-lg-4">
      <div class="card">
        <div class="card-header">
          <h3>Dean's List</h3>
        </div>
        <div class="card-body">
          <h2>29.99</h2>
          <p>Unlimited Matches Per Day</p>
          <p>Unlimited Messages Per Day</p>
          <p>Unlimited App Usage</p>
          <button type="button">Sign Up</button>
        </div>
      </div>
```

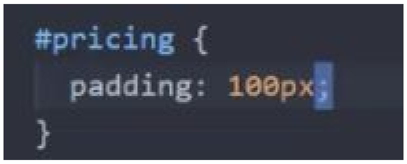
12. Now beautify your code, save, and refresh your webpage to see the changes.
13. We now want to move on to mediums which will apply to Ipad and tablet users.

14. Part of the specifications is that we want to have 2 of the cards take up half of the width. And we want the third one to take up the entire width.
15. To do this, add “col-md 6” after the “col-lg-4” specifications that we had before.



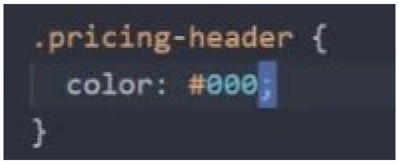
```
<div class="col-lg-4 col-md-6">
```

16. Now, we will need to stylize our buttons to match or web theme. Make sure that the first button has the outline and take up the full width. While the others will have the dark theme.
    a. For the outlined button use “btn btn-lg btn-block btn-outline-dark”
    b. Make sure to use the class “btn btn-lg btn-block btn-dark” for the dark theme
17. When you save and refresh your page, you will see that all we have to do now is stylize the css sheet to customize the cards.
18. To start, we want to change the padding of all the cards that we have. Scroll down to the bottom of your CSS sheet and add the pricing section, most important the pricing selector
    a. We want the padding of the cards to have about 100px on all four sides.



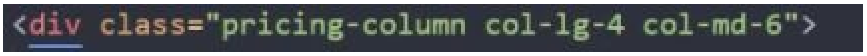
```
#pricing {
  padding: 100px;
}
```

19. Now, we want to stylize the pricing header. We will need to go back to the HTML file and add the class name of “pricing-header” describing the section.
20. To target the pricing header we need to specifically target it and make the color black so that it is consistent with the color scheme



```
.pricing-header {
  color: #000;
}
```

21. We will now add some padding to the columns between each of the cards to make it more visually pleasing
    a. To do this, we will add “pricing-column” to each of the column divs.



```
<div class="pricing-column col-lg-4 col-md-6">
```

22. Now go to the CSS and add 3% to the top and bottom of the pricing columns. And 2% for left and right.
23. Finally, we will center our text for each of the cards.

Lab Video 10 (Section 2h)

Z-Index:

At default, images are positioned at zero. However, the stacking order of elements can move farther from you starting at negative one and toward you starting at positive one. In this section, you will learn how to hide elements behind the features.

1. Set a background for the features that is not white so that they are not transparent. In order for the visual changes of an image's stacking to be seen, they must be opaque.
2. Next, you will need to go to the CSS file and change the z-index of the image to negative one. And its position to absolute. Needs to have a positioning statement or it won't work.

Lab Video 11 (Section 2i)

Objectives: Implement @media query for page responsiveness and mobile compatibility

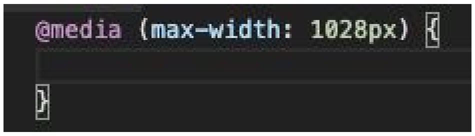
CSS enclosed in @media query are applied once the query result matches, i.e.:

```
@media (max-width: 900px) {
   h1 {
       color: red;
   }
}
```

The 'color: red' statement for h1 tag would only be applied once the user's interface width is less than 900px.

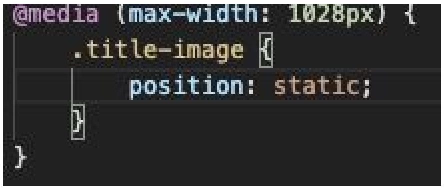
We would leverage this functionality to create separate versions of CSS rules for different users, namely, desktop and mobile devices.

1. Create a media query using the @media keyword, targeting a max-width of 1028px.



```
@media (max-width: 1028px) {

}
```

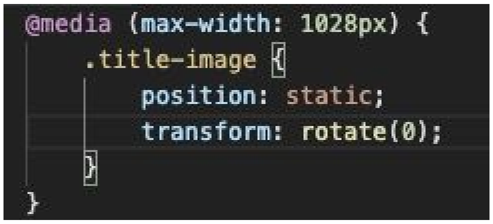
2. Append 'position: static' to 'title-image' class under @media query.



```
@media (max-width: 1028px) {
    .title-image {
        position: static;
    }
}
```

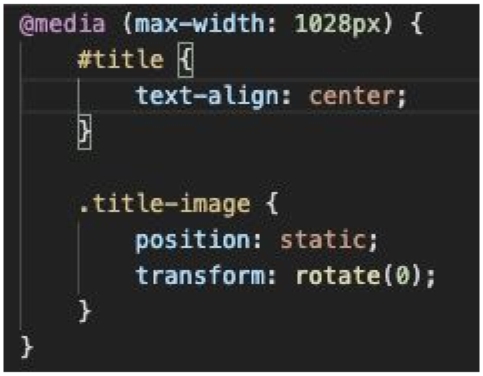
This rule resets the 'position' attribute of 'title-image' class once user resizes their window to a width less than 1028px, or accessing from a portrait-oriented mobile device

3. Append 'transform: rotate(0)' to 'title-image' class under @media query to reset the rotation effect.



```
@media (max-width: 1028px) {
    .title-image {
        position: static;
        transform: rotate(0);
    }
}
```

4. Lastly, append ‘text-align: center’ to ‘#title’ under @media query. This centers our title on mobile devices.

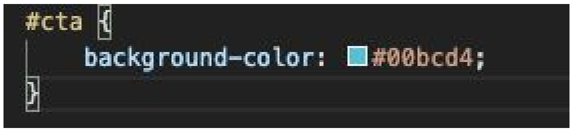


```
@media (max-width: 1028px) {
    #title {
        text-align: center;
    }

    .title-image {
        position: static;
        transform: rotate(0);
    }
}
```

Lab Video 12 (Section 2j)

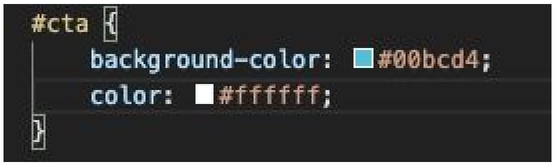
Objectives: Finish the Call to Action (CTA) section and finalizes our website

1. Append ‘background-color: #color of your choice’ to ‘#cta’. Note that ‘#color of your choice’ is the hex code / rgb(a) value / name you’d prefer. In this tutorial I’ll use ‘#00bcd4’ as an example.



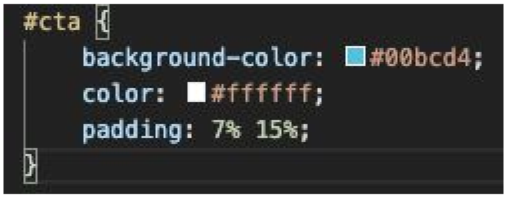
```
#cta {
    background-color: #00bcd4;
}
```

2. Append ‘color: #color of your choice’ to ‘#cta’.



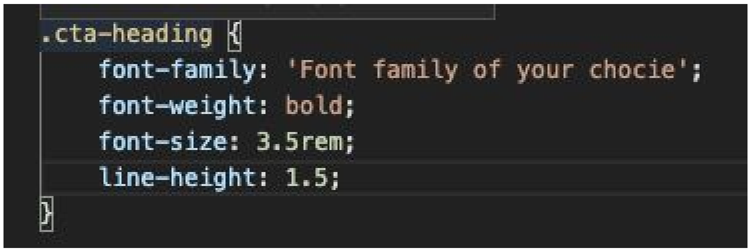
```
#cta {
    background-color: #00bcd4;
    color: #ffffff;
}
```

3. Append ‘padding: 7% 15%’ to ‘#cta’. This gives a nice perimeter to your ‘#cta’ section.



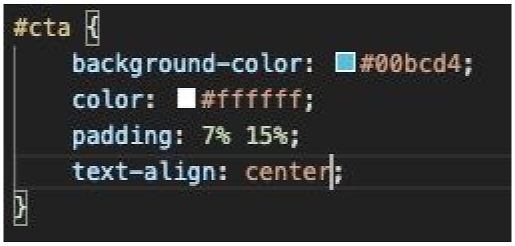
```
#cta {
    background-color: #00bcd4;
    color: #ffffff;
    padding: 7% 15%;
}
```

4. In your HTML file, add a ‘class=”cta-heading”’ attribute to the h3 tag below your CTA section. Now, set ‘font-family’, ‘font-weight’, ‘font-size’, and ‘line-height’ to the ‘.cta-heading’ class in your CSS file.



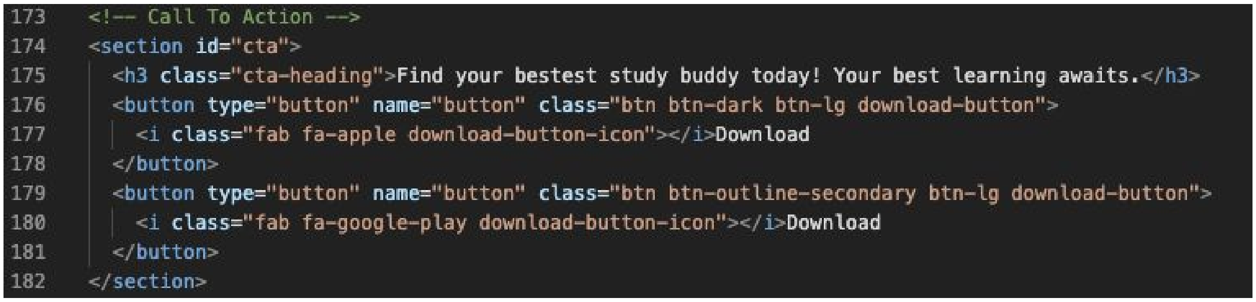
```
.cta-heading {
    font-family: 'Font family of your chocie';
    font-weight: bold;
    font-size: 3.5rem;
    line-height: 1.5;
}
```

5. Center your texts by adding a ‘text-align: center’ rule to your ‘#cta’.



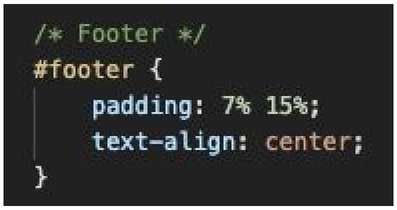
```
#cta {
    background-color: #00bcd4;
    color: #ffffff;
    padding: 7% 15%;
    text-align: center;
}
```

6. Stylize your buttons with font-awesome icons using the same method you’ve used previously. The result should look something like this:



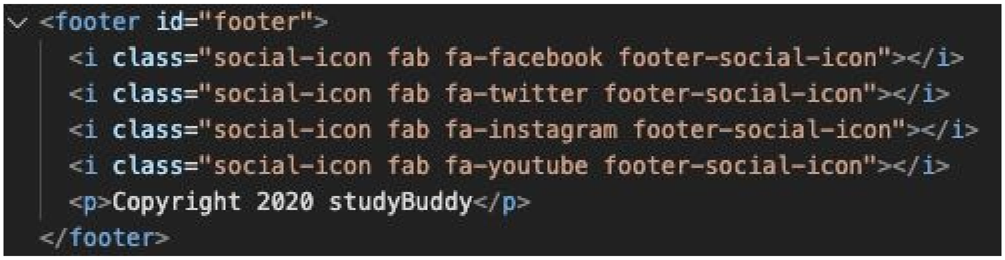
```
<!-- Call To Action -->
<section id="cta">
  <h3 class="cta-heading">Find your bestest study buddy today! Your best learning awaits.</h3>
  <button type="button" name="button" class="btn btn-dark btn-lg download-button">
    <i class="fab fa-apple download-button-icon"></i>Download
  </button>
  <button type="button" name="button" class="btn btn-outline-secondary btn-lg download-button">
    <i class="fab fa-google-play download-button-icon"></i>Download
  </button>
</section>
```

7. Lastly, we need to stylize the footer in CSS. Append ‘padding: 7% 15%’, ‘text-align: center’ to ‘#footer’. Remember to place your ‘#footer’ rule above @media query - @media query always goes by the last.



```
/* Footer */
#footer {
    padding: 7% 15%;
    text-align: center;
}
```

8. Add some social icons to your footer section using the same technique previously introduced. The result should look something like this:



```
<footer id="footer">
  <i class="social-icon fab fa-facebook footer-social-icon"></i>
  <i class="social-icon fab fa-twitter footer-social-icon"></i>
  <i class="social-icon fab fa-instagram footer-social-icon"></i>
  <i class="social-icon fab fa-youtube footer-social-icon"></i>
  <p>Copyright 2020 studyBuddy</p>
</footer>
```

Congratulations! You have completed your first Bootstrap website!