

# Lab 7 > Bootstrap Grid System

<https://getbootstrap.com/docs/4.1/layout/utilities-for-layout/>

## LAB 7: BOOTSTRAP GRID SYSTEM

This lab will help you become familiar with Bootstrap, a CSS & Javascript Framework. Go to [getbootstrap.com](https://getbootstrap.com) to find documentation to help you use Bootstrap. Bootstrap is based on a 12-column grid system, and uses Flexbox properties for layout. It's a responsive framework, which means the Bootstrap CSS stylesheet already has media queries built in. You will use the Bootstrap framework by writing your own HTML tags, and adding class names to reference Bootstrap's stylesheet. You can think of Bootstrap as a massive CSS library of class names.

**This lab is worth 10 points.**

## GETTING STARTED

Understanding Bootstrap's grid system is a good starting point. Start to read over the Bootstrap documentation for layout. As you build a layout with Bootstrap you can look at the documentation for more information.

<https://getbootstrap.com/docs/4.1/layout/grid/>

### STEPS:

1. Open Brackets, and choose **Open Folder** and browse to the Lab 7 root folder.
2. Select the Start HTML file, and preview the page. It looks like a blank page with Hello World on it. Notice the HTML page includes a link in the head section pointing to Bootstrap's CSS file, and it includes several Javascript frameworks at the bottom of the page.

*Note:* This **Starter Template** was copied from the GetBootstrap site. Look at the Bootstrap documentation, under Introduction, and notice they provide the code for you to start your own page using Bootstrap!

<https://getbootstrap.com/docs/4.3/getting-started/introduction/>

3. Change the text in <title> and <h1> tags to fit your project.

```
13 <title>Demo 1 - Bootstrap grid system</title>
14 </head>
15 <body>
16 <h1>Trying out Bootstrap!</h1>
```

4. Below the <h1> heading, you will add <div> tags and try out Bootstrap's grid system by adding class names (container, row, col) that reference Bootstrap's CSS classes.

--Add a div using the class name "container".

--Place another div inside the container div using the class name "row".

--Place 3 div tags inside your row, using the class name "col".

--Each col div should have an <h3> heading and one <p> paragraph.  
See the code sample below:

The row div is a Flex container, and it allows the col divs inside to scale and be arranged in one row by default. If you add any additional col divs to the row they will continue to scale automatically to fit the row.

```
13     <title>Demo 1 - Bootstrap grid system</title>
14 </head>
15 <body>
16     <h1>Trying out Bootstrap!</h1>
17     <div class="container">
18         <div class="row dianeProducts">
19             <div class="col">
20                 <h3>Product 1</h3>
21                 <p>This will blow your mind.</p>
22             </div>
23             <div class="col">
24                 <h3>Product 2</h3>
25                 <p>You just have to have this.</p>
26             </div>
27             <div class="col">
28                 <h3>Product 3</h3>
29                 <p>And now you need this product, too.</p>
30             </div>
31         </div><!-- closes the row -->
32     </div><!-- closes the container -->
33 </body>
```

*Note:* You can use the class name "container-fluid" or "container". Try "container-fluid" and preview the page to see the difference. Use .container-fluid for a full width container, spanning the entire width of the viewport. (Container allows some space on the sides at larger screen sizes.)

## CUSTOMIZE BOOTSTRAP WITH YOUR OWN CSS STYLESHEET

You can add your own stylesheet to customize the appearance of your page. Bootstrap is meant to be very open-ended, allowing you to build a page quickly, and then add your own custom styles. (You can also use SASS to customize Bootstrap. This lab does not cover SASS.) Bootstrap itself is very plain, white backgrounds, black text, simple sans-serif font. It is just waiting for you to add your own design. This allows every site built with Bootstrap to be unique in appearance.

1. Add the `<link>` tag to the `<head>` section pointing to your own CSS stylesheet name.

Make sure your link comes after the link to the Bootstrap stylesheet (to make sure your styles override the Bootstrap stylesheet).

```
7
8  <!-- Bootstrap CSS -->
9  <link rel="stylesheet" href="https://stackpath.bootstrapcdn.
    com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-
    MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO"
    crossorigin="anonymous">
10 <!-- my own custom style sheet -->
11 <link rel="stylesheet" href="dianestyles1.css">
12
13 <title>Demo 1 - Bootstrap grid system</title>
14 </head>
```

2. In Brackets, choose **New** and save the new document in your lab root folder, and name it with your CSS filename and the CSS extension.

The example here uses the name "dianestyles1.css". Make up your own CSS file name.

3. Add your own class name to the row div.

```
<div class="row dianeproducts">
```

Notice you can have multiple class names on an HTML element, with a space to separate the class names.

You can create your own class names to help you create CSS rules for your own custom styling.

4. Create a CSS rule to style the row div and the divs inside the row.

Add CSS rules for tags including body and H1, to refine text formatting.

Experiment with CSS properties to see that you can override Bootstrap's styling to add in colors, fonts, text formatting, background colors and images.

Here is our CSS style sheet:

```
1 body {  
2   background-color: #FF9869;  
3 }  
4 .dianeProducts div {  
5   border: white 1px solid;  
6   background-color: #FF2B4F;  
7 }  
8 .dianeProducts {  
9   box-shadow: 0px 0px 14px rgba(0,0,0,.3);  
10 }  
11 h1 {  
12   font-size: 5em;  
13   font-weight: normal;  
14   color: red;  
15   text-align: center;  
16   margin-bottom: 30px;  
17 }
```

Our page looks like this after adding styling:

# Trying out Bootstrap!

## Product 1

This will blow your mind.

## Product 2

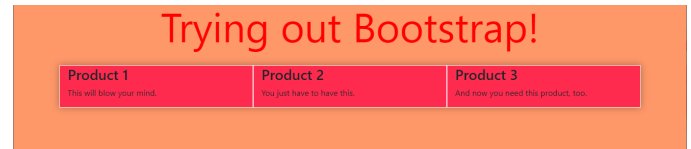
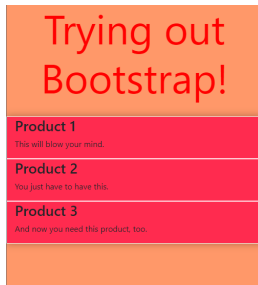
You just have to have this.

## Product 3

And now you need this product, too.

Bootstrap has media queries built into its stylesheet. You can add additional class names to indicate how your divs are arranged on various screen sizes.

The col class name is scaling, with all three divs fitting in one row...but we would like the layout to be one column on small screens, 2-column on tablets, and 3-column on larger screens.



Bootstrap is based on a 12-column grid. You can add a screen size and the number of columns in your class name for columns.

5. Modify the column class names so that the layout will be one column on small screens, 2-column on tablets, and 3-column on larger screens:

```
16      <h1>Trying out Bootstrap!</h1>
17      <div class="container">
18          <div class="row dianeProducts">
19              <div class="col-12 col-sm-6 col-md-4">
20                  <h3>Product 1</h3>
21                  <p>This will blow your mind.</p>
22              </div>
23              <div class="col-12 col-sm-6 col-md-4">
24                  <h3>Product 2</h3>
25                  <p>You just have to have this.</p>
26              </div>
27              <div class="col-12 col-sm-6 col-md-4">
28                  <h3>Product 3</h3>
29                  <p>And now you need this product, too.</p>
30              </div>
31          </div>
32      </div><!-- closes the row -->
33  </div><!-- closes the container -->
```

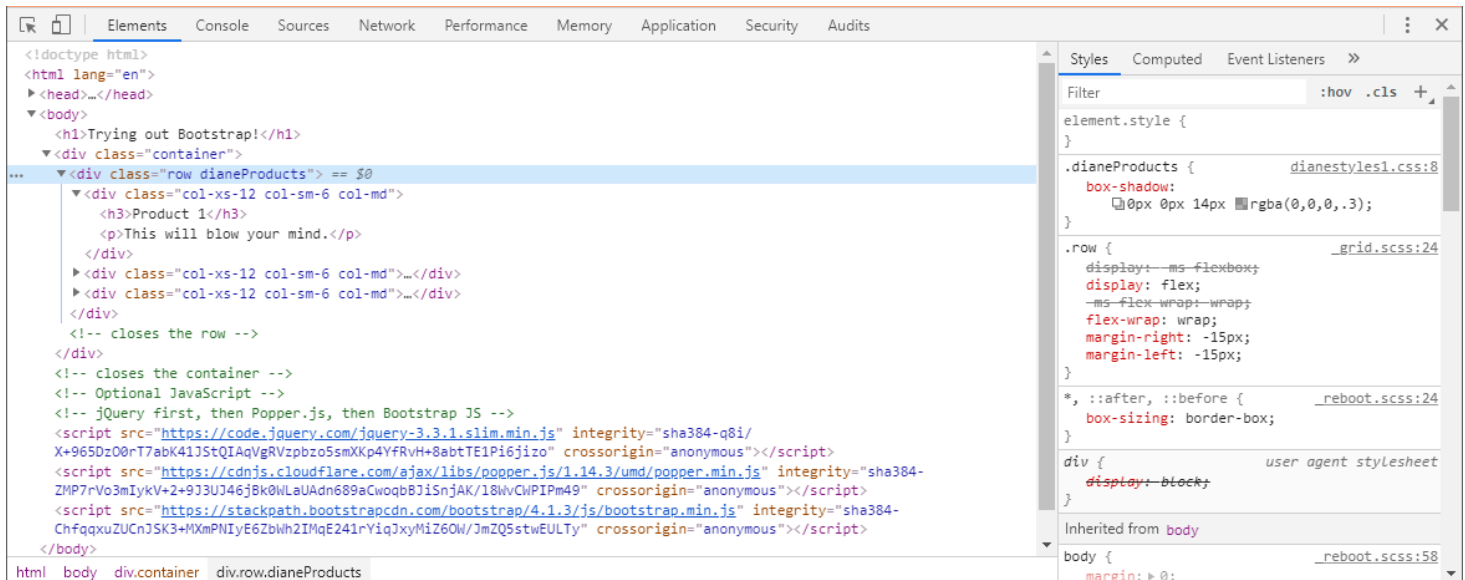
*Note:* The col class name is targeting the smallest screen size (Extra Small) and affects all screen sizes until you add more class names. Bootstrap is coded Mobile-first, so it is a good idea to specify the number of columns by adding the class name for the smallest screen size first. Then add additional class names to specify the grid layout on larger screen sizes as needed.

If you only specified the smallest size, it is inherited by all screen sizes. For example, it's ok to specify a header is col-12 to make it take up the entire 12 column grid on all screen sizes.

Bootstrap uses the following breakpoints in its media queries:

|                            | Extra small<br><576px | Small<br>≥576px | Medium<br>≥768px | Large<br>≥992px | Extra large<br>≥1200px |
|----------------------------|-----------------------|-----------------|------------------|-----------------|------------------------|
| <b>Max container width</b> | None (auto)           | 540px           | 720px            | 960px           | 1140px                 |
| <b>Class prefix</b>        | .col-                 | .col-sm-        | .col-md-         | .col-lg-        | .col-xl-               |
| <b># of columns</b>        | 12                    |                 |                  |                 |                        |

6. Try using **Chrome Inspect** to learn a lot about the styling that each Bootstrap class name is adding. For example, by selecting the row div, you can see **display: flex** in the CSS style sheet. You can see Bootstrap is using Flexbox layout, and expecting the row to help you position the contents inside the row using Flex properties.



7. Experiment by adding more divs to your row. We now have 6 divs in one row. If more than 12 columns are placed within a single row, each group of extra columns will wrap onto a new line. A row container can be used as a section in your website. Notice that if the objects don't fit they will flow into a second row within the row div.



## COLUMN BREAKS

8. Add a new container and row, with 4 divs inside.

```
45
46 <div class="container">
47   <div class="row dianeProducts">
48     <div class="col">Column containing content that will describe my
49       products in detail.</div>
50     <div class="col">Column containing content that will describe my
51       products in detail.</div>
52     <div class="w-100"></div>
53     <div class="col">Column containing content that will describe my
54       products in detail.</div>
55     <div class="col">Column containing content that will describe my
56       products in detail.</div>
57   </div>
58 </div>
```

You can add the class name **w-100** to an empty div to force a break between columns. The example above is using the class name **col** to make the divs scale and fit automatically on all screen sizes.

## RESPONSIVE IMAGES

<https://getbootstrap.com/docs/4.1/content/images/>

9. Add images to the divs:

The images are not scaling at first. But you just need to add the class name **img-fluid** to any image to make it responsive.

```
45
46 <div class="container">
47   <div class="row">
48     <div class="col">
49       
50       Column containing content that will describe my products in detail.
51     </div>
52     <div class="col">
53       
54       Column containing content that will describe my products in detail.
55     </div>
56     <div class="w-100"></div>
57     <div class="col">
58       
59       Column containing content that will describe my products in detail.
60     </div>
61     <div class="col">
62       
63       Column containing content that will describe my products in detail.
64     </div>
65   </div>
66 </div>
67
```

## MARGIN CLASSES - UTILITIES

10. Add space between the container divs by using spacing utility classes:

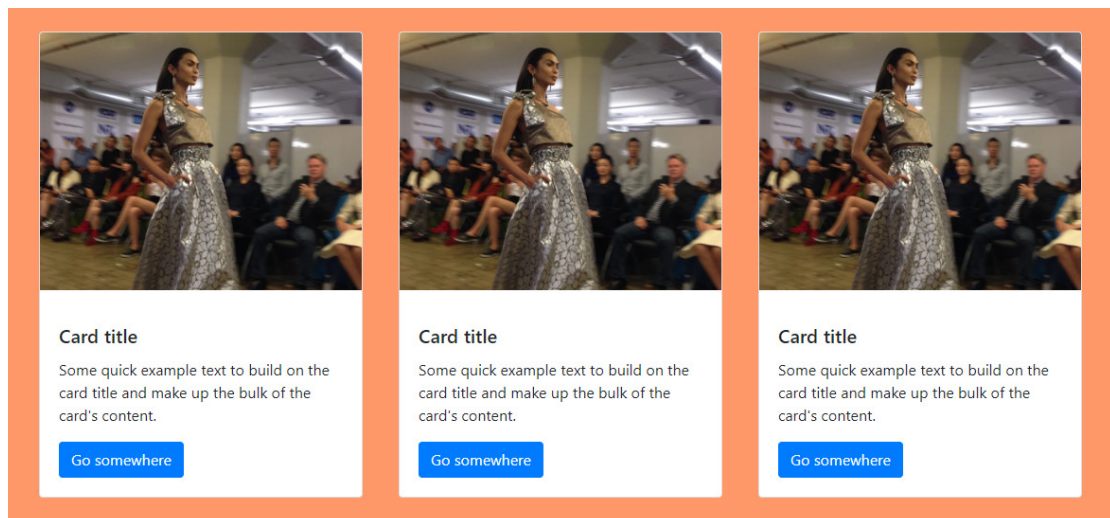
```
<div class="container mt-5">
```

Adding the class name mt-5 is adding a top margin on the container div. 5 is setting the distance; try various values in whole numbers.

Read the documentation to see the patterns you can use in your class names to add margins or padding to HTML elements.

<https://getbootstrap.com/docs/4.1/utilities/spacing/>

## COMPONENTS - CARDS



11. Try using the Bootstrap documentation to add a component to your page. Find the HTML example of a **card** and copy and paste it into a new container and row div.

<https://getbootstrap.com/docs/4.1/components/card/>

12. Paste in 3 cards, and then change the src of the image tags to use photos inside your lab Images folder.

The images are already scaling to fit the card size. The code example that you pasted from the Bootstrap site includes a width on the card of 18rem.

```
<div class="card" style="width: 18rem;">
```

You can change or remove that width to fit your layout.

13. Change the width to 30%;

14. Add the mx-auto class name to the card divs to center them.



Here's the HTML of the cards we added:

```
66 </div>
67 <div class="container mt-5">
68   <div class="row mb-5">
69     <div class="card mx-auto" style="width:30%;">
70       
71       <div class="card-body">
72         <h5 class="card-title mt-3">Card title</h5>
73         <p class="card-text">Some quick example text to build on the card title and
          make up the bulk of the card's content.</p>
74         <a href="#" class="btn btn-primary">Go somewhere</a>
75       </div></div>
76       <div class="card mx-auto" style="width:30%;">
77       
78       <div class="card-body">
79         <h5 class="card-title mt-3">Card title</h5>
80         <p class="card-text">Some quick example text to build on the card title and
          make up the bulk of the card's content.</p>
81         <a href="#" class="btn btn-primary">Go somewhere</a>
82       </div></div>
83       <div class="card mx-auto" style="width:30%;">
84       
85       <div class="card-body">
86         <h5 class="card-title mt-3">Card title</h5>
87         <p class="card-text">Some quick example text to build on the card title and
          make up the bulk of the card's content.</p>
88         <a href="#" class="btn btn-primary">Go somewhere</a>
89       </div>
90 </div><!-- closes a card -->
91 </div><!-- closes the row -->
92 </div><!-- closes the container -->
```

*Extra Credit:* Try out a Carousel slide show! Try creating a NavBar (which can toggle into a hamburger icon on small screens)!

Follow the documentation from the Bootstrap website:

<https://getbootstrap.com/docs/4.1/components/carousel/>

<https://getbootstrap.com/docs/4.1/components/navbar/>

## WHEN YOU ARE FINISHED

Name your root folder:

**Last Name\_Lab7**

Save a copy of your root folder as a zip folder, and upload your zip folder to Canvas.

I hope you found this Lab useful!