



**TASK**

# **Bootstrap**

[Visit our website](#)

# Introduction

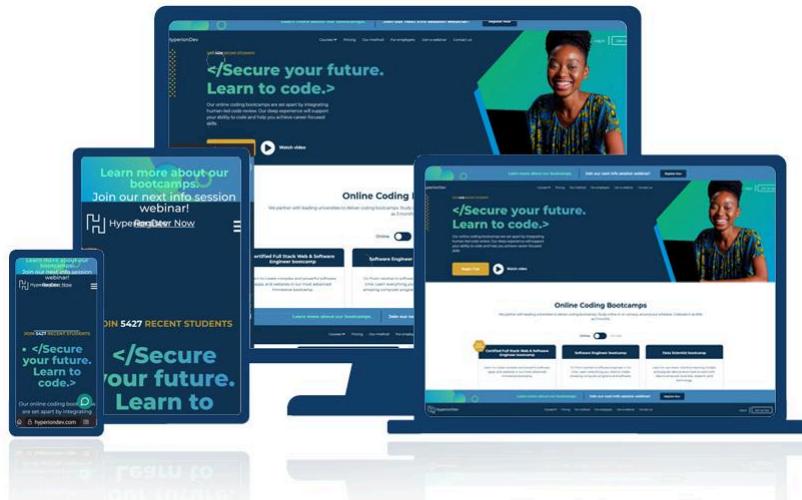
## WELCOME TO THE BOOTSTRAP TASK.

You've learnt the basics of CSS and its role in web development, but now it's time to level up with Bootstrap – a powerful framework for creating responsive, stylish websites. The best part? You do not need to be a pro to use it. Bootstrap simplifies CSS, offering pre-built components to save time and effort. It is a toolkit crafted by developers to make your life easier.

Let's dive in and see how Bootstrap can transform your web design!

## RESPONSIVE DESIGN

Take a look at the picture below. Notice how, in the world we live in, we have so many different screen sizes. Think about one of the most popular applications that people use nowadays, like Whatsapp.



Have you ever noticed that regardless of the device that you use Whatsapp on, the screen always remains the same? This is one of the critical features of responsive design.

At this point in time, you most likely have only created one website design, and that design works with your laptop screen, but now imagine you had to look at your website on your mobile device or on a 4k TV screen. As you can imagine, your website

would end up breaking on those devices; this is because you haven't implemented any form of **responsive design**.

By now, you might be wondering what responsive design is. It's a web developer's way of creating a website that adapts to the size of the user's screen. While this can take some time, Bootstrap simplifies the process and helps you implement it effectively.

## WHAT IS BOOTSTRAP?

We have also mentioned that there are many developers around the world who are creating systems for developers to use to help them save time and energy. Bootstrap is one of the tools that have been created by developers to help other developers save time and effort during the process of creating systems.

It is known as a **framework** which you will slowly become more familiar with as you progress through the course. Simply put, a framework is code written by someone else that you can use in your program to make your development progress quicker and make your code more manageable and cleaner to write.

In this task, we will be going through how to install bootstrap and how to use it to make your website responsive and well-structured.

## INSTALLING BOOTSTRAP

Let's go through the quick process of "installing" bootstrap (it's just like importing CSS and JavaScript files into your HTML file).

Just follow these steps:

1. Head over to [Bootstrap's official website](#) and scroll down to the section that says "**Include via CDN**".
2. Look for the CSS link and copy it. It will look something similar to the below link:

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH" crossorigin="anonymous">
```

3. Open any HTML file where you would like to apply Bootstrap styling and paste the CSS link inside the `<head>` section to ensure the styles are applied.  
For example:

```
<head>
  <meta charset="UTF-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <title>Bootstrap Example</title>
  <link
    href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
    rel="stylesheet"

    integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
    crossorigin="anonymous"
  />
</head>
```

4. Go back to the "**Include via CDN**" section and copy the JavaScript bundle link, which looks similar to this:

```
<script
  src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
  integrity="sha384-YvpcrYf0tY31HB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIds1K1eN7N6jIeHz"
  crossorigin="anonymous"></script>
```

5. Open the same HTML file and add the JavaScript bundle at the very end of the `<body>` section, just before the closing `</body>` tag. This helps to ensure the following:
  - Firstly, the page content and styles are loaded first for faster rendering.
  - Secondly, it ensures that interactive Bootstrap features like dropdowns, modals, and tooltips will function properly.

For example:

```
<body>
  <h1>Hello bootstrap</h1>

  <script

    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
    integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIds1K1eN7N6jIeHz"
    crossorigin="anonymous"
  ></script>
</body>
```

6. Finally, save your HTML file, and that's it. You should now be able to use Bootstrap's features to style your pages more easily and add interactive components with minimal effort.

## USING BOOTSTRAP

Just like any concept in programming, you will need to learn how to implement bootstrap. Before we can begin implementing it, we need to understand the backbone of Bootstrap and how it will calculate how to create a responsive website.

### The grid system

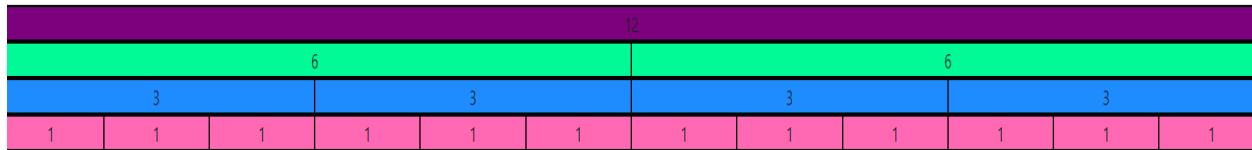
Whenever you look at a website, you are actually looking at a grid system. Take a look at the image below of our website.



As you can see, there are a lot of different colours on the screen. These are known as “columns” (usually invisible); this is what helps a web page identify where certain elements are. Bootstrap works with the grid system.

You may notice that there are 12 purple columns. This is the **maximum** number of columns bootstrap can work with. This means that once you start designing your web pages using Bootstrap, you should never forget that you only have twelve columns to design your web page on. work with the limit of 12 columns in the back of your head.

Take a look at the image below:



This is an example of a grid pyramid that was created using Bootstrap and standard HTML. As you can see, we are now able to place some aspects next to each other using the different columns Bootstrap provides for us.

Because we are now making use of grids, what you will find is that no matter how you resize the browser, the website will still stay on the page and automatically adjust itself to fit all the content in the way it was placed. This can be done in minutes with Bootstrap, compared to writing our own version of the grid system using normal CSS.

## Screen sizes

While Bootstrap is powerful, it becomes even more powerful once we start learning and implementing screen sizes. Bootstrap allows us to select what data will be displayed based on the screen size we provide.

Take a look at the screenshot below:

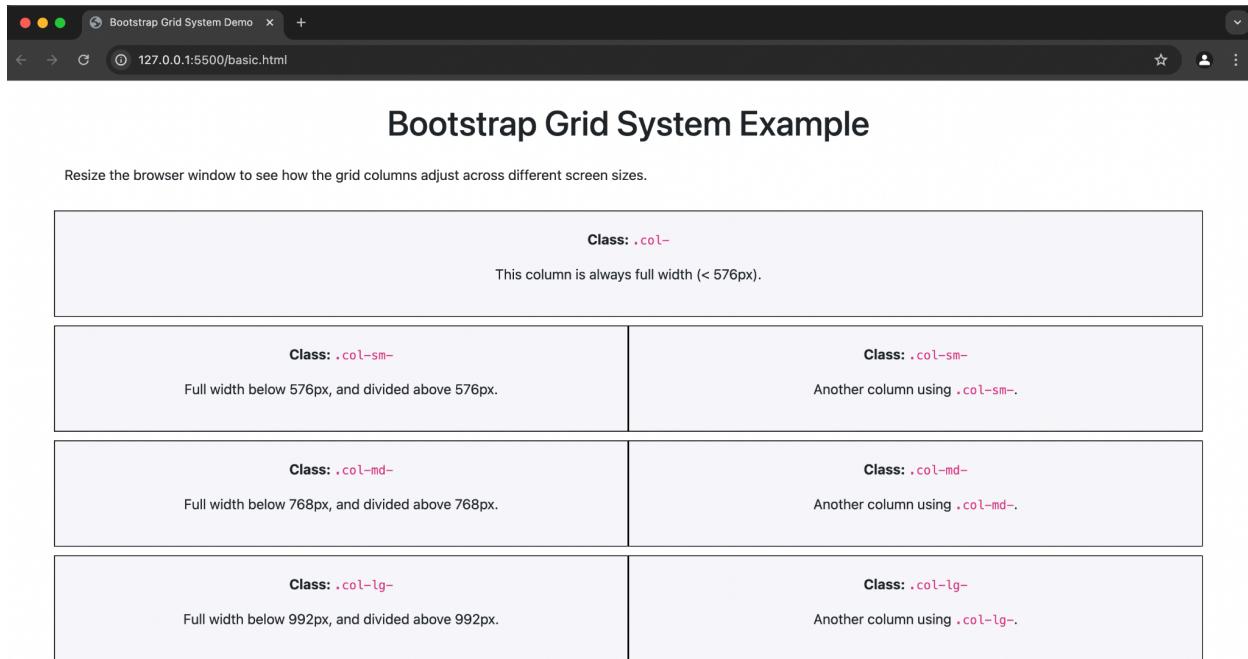
Class Prefix	Screen Size	Grid Behaviour	Container Width	Suitable For
.col-	Extra small (<576px)	Horizontal at all times	None (auto)	Portrait phones
.col-sm-	Small (>=576px)	Collapsed to start, horizontal above breakpoints	540px	Landscape phones
.col-md-	Medium (>=768px)	Collapsed to start, horizontal above breakpoints	720px	Tablets
.col-lg-	Large (>=992px)	Collapsed to start, horizontal above breakpoints	960px	Laptops
.col-xl-	Extra Large (>=1200px)	Collapsed to start, horizontal above breakpoints	1140px	Laptops and Desktops

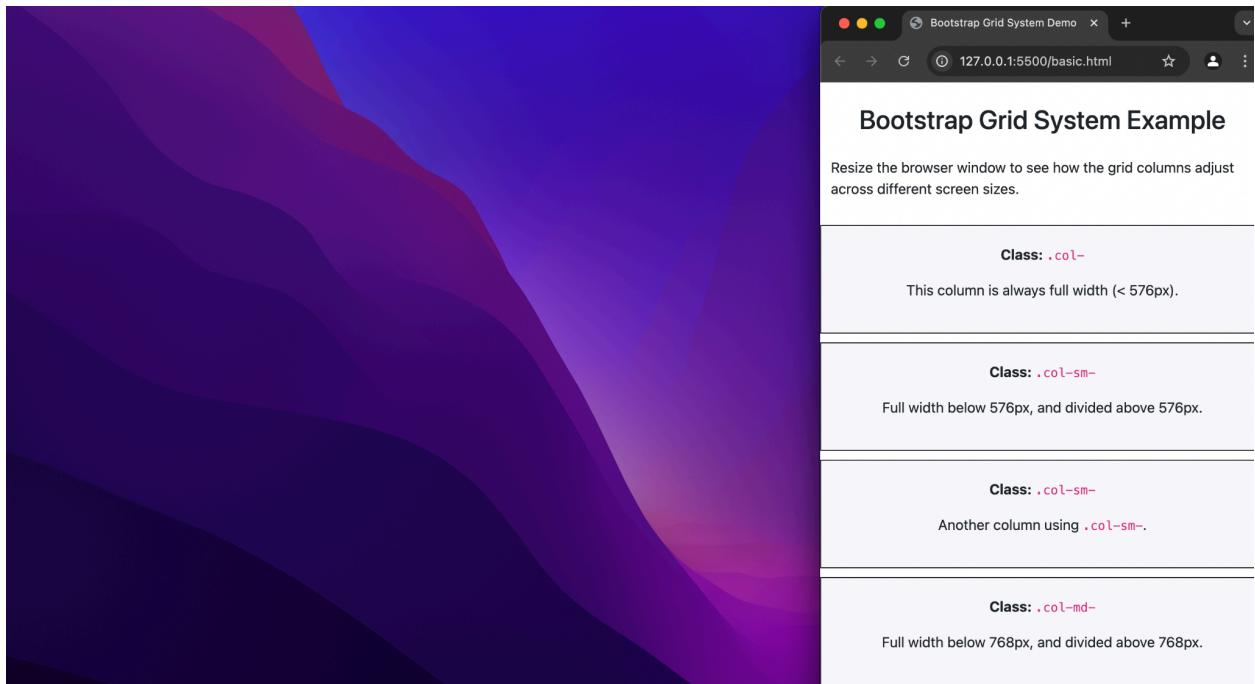
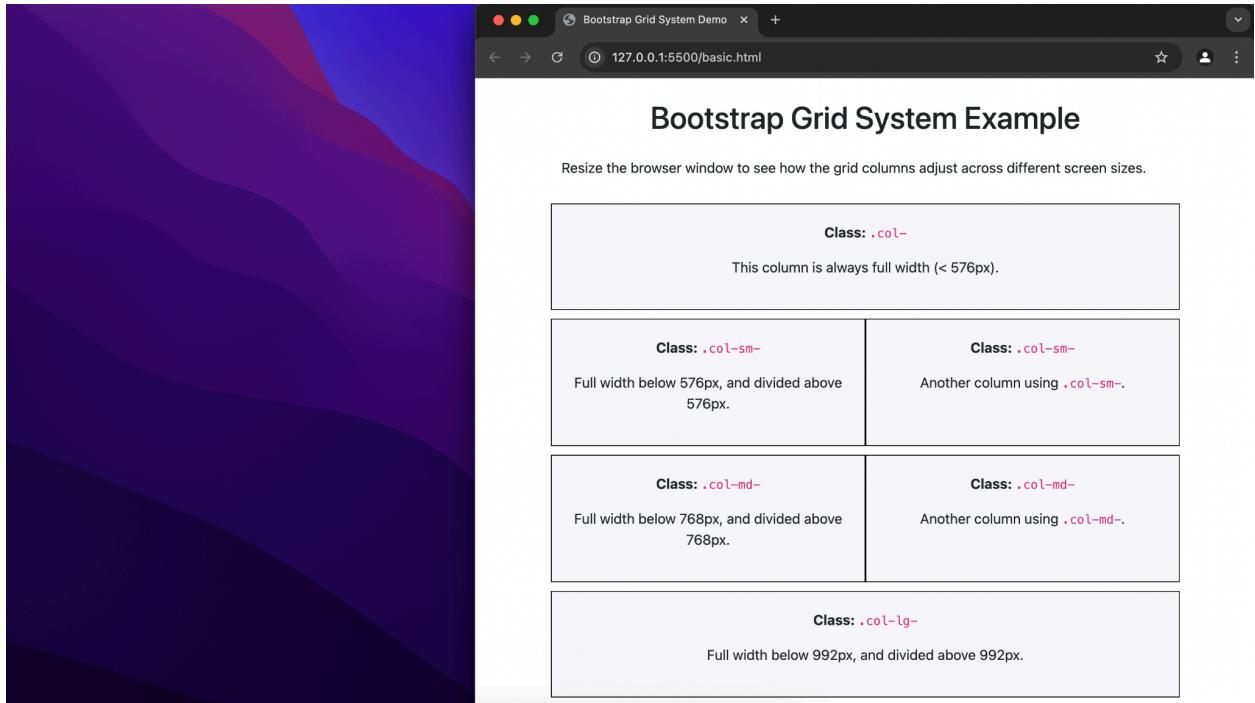
These are all the screen sizes that are used in Bootstrap. For the most part, when you

work with Bootstrap, you will most commonly use “**col-md**” (medium device) as you are most likely working on a tablet/laptop.

However, you will implement all the other screen sizes the exact same way you will be implementing the medium device tag.

Look at this image that demonstrates how these classes (.col-, .col-sm-, .col-md-, etc.) adjust their layout, and notice how the column responses change dynamically as the browser size changes.





Here's the code used to create the layout shown in the images above:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Bootstrap Grid System Demo</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet"/>
  </head>

  <body>
    <div class="container">
      <h1 class="text-center my-4">Bootstrap Grid System Example</h1>
      <p class="mb-4">
        Resize the browser window to see how the grid columns adjust across different screen sizes.
      </p>

      <div class="row">
        <div class="col col-example">
          <p><strong>Class:</strong> <code>.col-</code></p>
          <p>This column is always full width (< 576px).</p>
        </div>
      </div>

      <div class="row">
        <div class="col-sm col-example">
          <p><strong>Class:</strong> <code>.col-sm-</code></p>
          <p>Full width below 576px, and divided above 576px.</p>
        </div>
        <div class="col-sm col-example">
```

```

<p><strong>Class:</strong> <code>.col-sm-</code></p>
<p>Another column using <code>.col-sm-</code>.</p>
</div>
</div>

<div class="row">
  <div class="col-md col-example">
    <p><strong>Class:</strong> <code>.col-md-</code></p>
    <p>Full width below 768px, and divided above 768px.</p>
  </div>
  <div class="col-md col-example">
    <p><strong>Class:</strong> <code>.col-md-</code></p>
    <p>Another column using <code>.col-md-</code>.</p>
  </div>
</div>

<div class="row">
  <div class="col-lg col-example">
    <p><strong>Class:</strong> <code>.col-lg-</code></p>
    <p>Full width below 992px, and divided above 992px.</p>
  </div>
  <div class="col-lg col-example">
    <p><strong>Class:</strong> <code>.col-lg-</code></p>
    <p>Another column using <code>.col-lg-</code>.</p>
  </div>
</div>

<div class="row">
  <div class="col-xl col-example">
    <p><strong>Class:</strong> <code>.col-xl-</code></p>
    <p>Full width below 1200px, and divided above 1200px.</p>
  </div>
  <div class="col-xl col-example">
    <p><strong>Class:</strong> <code>.col-xl-</code></p>
    <p>Another column using <code>.col-xl-</code>.</p>
  </div>
</div>

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>
</body>
</html>

```



## Take note

While CSS is usually placed in a separate file, it's included in the above example code within the HTML for simplicity and to make this example easier to follow.

---

Let's break down the key concepts of Bootstrap's grid system to help you understand how it works. This explanation will set the stage for understanding the rows and columns shown in the image below.

### 1. **Col**

- The `col` keyword stands for "column." It's how Bootstrap identifies elements as part of the grid structure. In the diagram, the green blocks represent the columns, which are the main units that divide the row.

### 2. **sm/md/lg/xl**

- These define screen size breakpoints, allowing your layout to adapt to different devices. For instance, using `col-md-6` makes a column span half the row's width on medium screens and larger. The different block colours in the diagram demonstrate how these spans would look.

### 3. \*

- The asterisk represents the number of columns a block spans (out of 12):
  - i. 12 (**purple**): A full-width column spanning the entire row.
  - ii. 6 (**green**): A column spanning half the row.
  - iii. 3 (**blue**): A column taking up one-quarter of the row.
  - iv. 1 (**pink**): A narrow column taking up 1/12 of the row.

These colours visually demonstrate how column sizes and spans work within the 12-column grid."

Now that you understand the concepts of columns, breakpoints, and sizing, let's take a closer look at how rows work in Bootstrap's grid system, as shown in the diagram below:



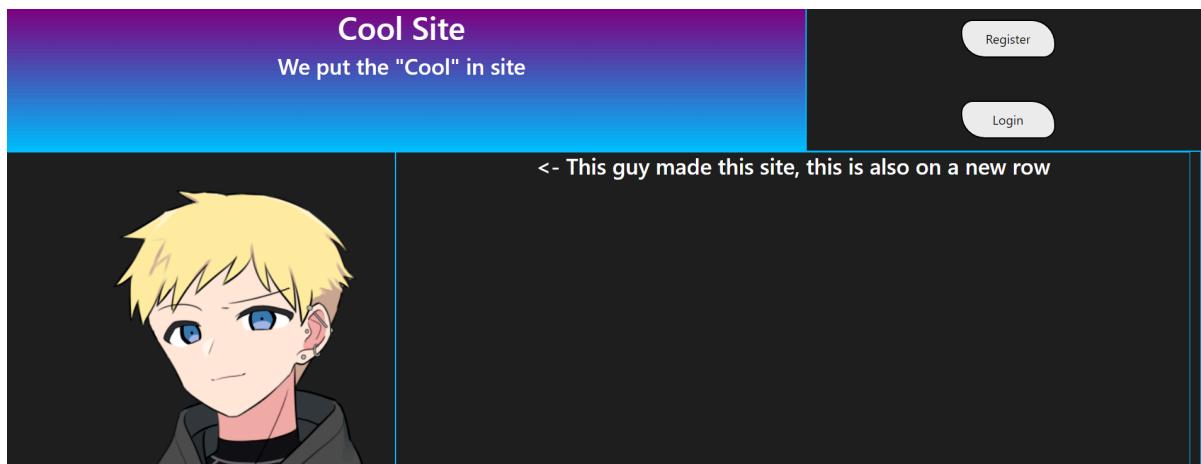
In the grid pyramid image that shows all the columns in different colours, you will notice that each number goes onto a different “line”. These lines are known as rows. Unlike columns, there can be an infinite number of rows (as a web page can reach an unlimited height).

## Putting it all together

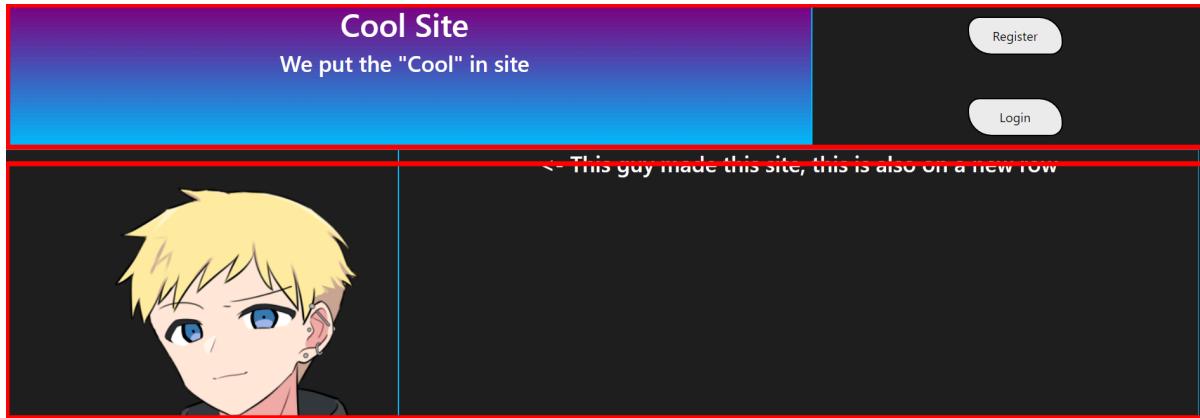
Now that we know that websites are built using the grid system, we can start making use of Bootstrap. Something that makes Bootstrap so powerful for new learners is that they only have to learn class names. It adds no additional HTML tags or new systems that make it a complex learning experience.

Here is a cheat sheet that contains all the style snippets you need. You can use this: [\*\*HTML and CSS Cheat Sheet\*\*](#).

Let's start by creating a basic webpage using Bootstrap. Below is a screenshot of the final product. While it may not be the most visually stunning design, it serves as a practical example to demonstrate how Bootstrap works:



Let's begin by identifying where the rows are in this project. The image illustrates a website layout divided into distinct rows, with each row outlined in red.



The code below creates two horizontal rows in a website layout using Bootstrap's `row` class. Each `<div class="row">` represents a horizontal section, which spans the full width of the container. The goal here is to divide the page into distinct rows without yet focusing on columns.

```

<body>
  <div class="row">
    <!-- Content for the first row can be added --&gt;
  &lt;/div&gt;

  &lt;div class="row"&gt;
    <!-- Content for the second row can be added --&gt;
  &lt;/div&gt;

  <!-- Insert imported JavaScript bundle before the closing &lt;/body&gt; --&gt;
&lt;/body&gt;
</pre>

```

Now that we have created our rows, let's look at the image below to see how many columns your website will be made up of:



You will notice that inside each section, there are either 8 or 4 different columns. We

decide how many we want. The code just below divides the website into columns with the "col-md-8" and "col-md-4" classes. This allows for a responsive layout with 8 and 4 column sections. The specific numbers in the code refer to the column widths - 1-8 and 9-12 respectively. The layout can be adjusted by modifying these column class values to fit the desired design. So, let's head over to our code and put our columns in place:

```
<body>
  <div class="row">
    <div class="col-md-8">
      
    </div>
    <div class="col-md-4">
      <!-- This is col 9-12 (login/register button) --&gt;
    &lt;/div&gt;
  &lt;/div&gt;
  &lt;div class="row"&gt;
    &lt;div class="col-md-4"&gt;
      <!-- This is col 1-4 (The image) --&gt;
    &lt;/div&gt;
    &lt;div class="col-md-8"&gt;
      <!-- This is col 5-12 (The text) --&gt;
    &lt;/div&gt;
  &lt;/div&gt;

  <!-- Insert imported JavaScript bundle before the closing &lt;/body&gt; --&gt;

&lt;/body&gt;</pre>
```

As you can see, we have added a **div** inside each of the **row** divs and assigned each of them a class name of **col-md-\***. The **\*** in this class represents the number of columns that the **div** will span within the 12-column grid system provided by Bootstrap. This number can be replaced with any value between 1 and 12, as long as the total number of columns within a single row does not exceed 12.

For example, in the code above, the first row contains two columns: one with **col-md-8** and another with **col-md-4**. These add up to 12 (8 + 4), so the layout stays within Bootstrap's grid system rules. Similarly, you can adjust the numbers to create different layouts, such as three equal columns (**col-md-4**, **col-md-4**, **col-md-4**) or one large and one small column (**col-md-9**, **col-md-3**).

The **md** in **col-md-\*** refers to the breakpoint for medium-sized devices. You are able to replace **md** with other breakpoints, like **sm**, **lg**, or **xl**, to define how columns should behave on different screen sizes.

After planning out the rows and columns for the layout, you can start adding standard HTML elements, such as images, buttons, text, or other content, inside these **div** tags. The example below demonstrates a basic layout that creates something similar to the images shown above. However, feel free to customise it with your own code, for instance, you could replace the example content with an image of your choice or adapt it to better match your design preferences.

Experimenting with your own content and making adjustments will not only help you understand how the Bootstrap grid system works but also allow you to create a layout tailored to your specific goals.

```
<body>
  <div class="row">
    <div class="col-md-8 background">
      <h1>Cool Site</h1>
      <h3>We put the "Cool" in site</h3>
    </div>
    <div class="col-md-4">
      <button>Register</button>
      <br>
      <br>
      <button>Login</button>
    </div>
  </div>
  <div class="row">
    <div class="col-md-4">
      
    </div>
    <div class="col-md-8">
      <h3><- This guy made this site, this is also on a new row</h3>
    </div>
  </div>

  <!-- Insert imported JavaScript bundle before the closing </body> -->
</body>
```

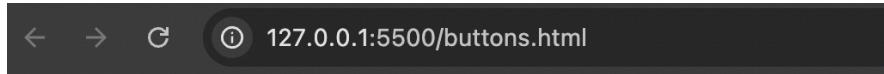
Once you have finished up all your code you have officially created your first ever responsive website. This is a huge step in your programming journey and you should be proud of what you have accomplished.

## BOOTSTRAP BUTTONS AND STYLING

Bootstrap provides a powerful, built-in way to style buttons, making it easy to create visually consistent and interactive web elements. Each button style can be applied by adding specific class names, allowing developers to convey different purposes and actions clearly to users.

Below is a table summarising common Bootstrap button classes and their descriptions:

Class	Description
.btn	Adds basic styling to any button
.btn-danger	Indicates a dangerous or potentially negative action
.btn-dark	Dark grey button
.btn-default	Indicates a default/standard button
.btn-info	Contextual button for informational alert messages
.btn-light	Light grey button
.btn-link	Makes a button look like a link (will still have button behavior)
.btn-primary	Provides extra visual weight and identifies the primary action in a set of buttons
.btn-success	Indicates a successful or positive action
.btn-secondary	Indicates a "less" important action
.btn-warning	Indicates caution should be taken with this action



Class	Button	Description
.btn-danger		Indicates danger
.btn-dark		Dark grey button
.btn-default		Default/standard button
.btn-info		For informational alerts
.btn-light		Light grey button
.btn-link		Link-like button
.btn-primary		Primary action button
.btn-success		Indicates success
.btn-secondary		Less important action
.btn-warning		Indicates caution

These class names allow for quick customisation and visual feedback, which can enhance the user experience and functionality of a web application.

For more detailed examples and advanced usage, visit the [\*\*Bootstrap Documentation\*\*](#).

---



## Practical task

You are going to create an online shopping page using Bootstrap. The idea behind this task is to create an online store layout with products displayed in a grid format. Please implement all of the features below by using Bootstrap in your program.

Follow these steps:

- Create a new file named **shopping.html**.
- Items
  - Add 10 products to the page.
  - Use the grid system to display the products in rows and columns.
  - To make the images responsive, you can use Bootstrap's **img-fluid** class, which will make images scale based on the screen size. Additionally, add custom CSS to change the image when the user hovers over it (e.g., scaling the image or changing the source).
  - Have your page display a price and name for each product.
  - Include a button under each product to "Buy" the product (this does not need to be functional yet).
- Footer
  - Create a horizontal form.
  - This form should get the customer's contact details (this does not need to be functional yet.)
  - Include a small company logo next to the form.

**Important:** Be sure to upload all files required for the task submission inside your task folder and then click "Request review" on your dashboard.

---



## Share your thoughts

Please take some time to complete this short feedback [form](#) to help us ensure we provide you with the best possible learning experience.

---