Difference between Library and Framework:

Library:

- A **library** is a collection of pre-written code, functions, or routines that developers can use to perform common tasks without having to write the code from scratch. Libraries are designed to be called upon by the developer as needed, allowing for more efficient and organized coding. Libraries provide specific functionalities that can be integrated into a program to extend its capabilities.

Framework:

- A **framework** is a comprehensive software structure that provides a foundation for building applications. It includes predefined rules, guidelines, and components that dictate how an application should be organized and developed. Frameworks often dictate the flow of control within an application, calling the developer's code at specific points. They provide a complete environment for developing software, making it easier to implement complex features while ensuring consistency and best practices.

Differences:

- Think of a library as a set of kitchen tools (like knives, spatulas, and blenders). You are the chef, and you decide which tool to use and when. You control the cooking process completely.

  Now, think of a framework as a cooking robot. The robot already has a plan in mind for the entire cooking process. It tells you when to chop, when to cook, and even when to serve. You follow its instructions, and the robot controls the flow.

Bootstrap 5:

Why to use Bootstrap:

- When creating a website, there are several common features that are typically required across web pages, including:
    - **Layout**: The skeleton structure of web pages.
    - **Typography**: Font styling and management.
    - **Navigation Menus**: Structuring site navigation for users.
    - **Form Layout**: Organizing and styling forms for user input.
    - **Buttons**: Applying consistent styles to buttons or links.
    - **Pagination**: Managing content across multiple pages.
    - **Icon Fonts**: Incorporating scalable icons.
    - **Responsiveness**: Handling different device sizes through media queries, which can make the CSS code longer and more complex.

  To streamline these common tasks, having a ready-made CSS framework allows developers to build websites quickly and easily. This type of pre-designed CSS is known as a **CSS framework** and provides all the essential features needed to construct nearly any website.

  By utilizing a CSS framework like Bootstrap, developers can access a collection of pre-designed components that simplify the web development process. Bootstrap not only saves time but also ensures that websites are responsive, meaning they look great on devices of all sizes.

  With a large community behind it, Bootstrap offers extensive resources, tutorials, and third-party plugins, making it easier for beginners to find support as they learn. Moreover, Bootstrap is customizable, allowing developers to modify its styles to fit their unique design needs.

  Learning Bootstrap will enhance your existing skills in HTML and CSS and empower you to create modern, professional websites with ease.

Introduction:

- Bootstrap 5 is a CSS based front-end framework that helps developers create responsive and visually appealing websites quickly and efficiently.
- It is an open-source framework developed by **Twitter.**
- It offers a wide range of HTML, CSS, and JavaScript tools to create responsive and mobile-first websites.

- Some large companies that use Bootstrap are: Twitter, Spotify, LinkedIn and many more.

Advantage of Using Bootstrap:

Using Bootstrap offers several benefits, including:

- **Saves Time:** Rapidly develop websites by using pre-built components.
- **Responsive Features:** Built-in responsiveness ensures websites look good on all devices.
- **Consistent Design:** Maintains a uniform style throughout the site, enhancing user experience.
- **Easy to Use:** Intuitive structure and documentation make it beginner-friendly.
- **Cross-Browser Compatibility:** Works well across various web browsers.
- **Open Source:** Free to use, modify, and distribute.

Version History of Bootstrap:

Bootstrap 1 (August 2011)

- Release: Initial version by Twitter.
- Features: Introduced a responsive grid system and basic styling for UI elements.

Bootstrap 2 (January 2012)

- Updates: Improved grid system for better layout control.
- Features: Added support for responsive layouts and new UI components like buttons and navigation bars.

Bootstrap 3 (August 2013)

- Redesign: Major update with a mobile-first approach.
- Features: Introduced a flat design, new components like cards, and enhanced grid system.

Bootstrap 4 (January 2018)

- Transition: Shifted to Flexbox for layout flexibility.

- Features: Introduced utility classes, improved customization with Sass, and replaced deprecated components.

Bootstrap 5 (May 2021)

- Latest Version: Focus on modern web development.
- Features: Dropped jQuery, introduced a utility API, improved form controls, and added new interactive components.
- Switch to Vanilla JavaScript
- Dropped support of Internet Explorer 10 and 11:
- Responsive Fonts: Bootstrap has introduced responsive font sizes which will automatically resize the typography element according to the viewport size

Student Task:

- Visit Bootstrap's official website and explore the documentation on components and utilities.

Download and Use Bootstrap:

- There are 2 ways to start using Bootstrap 5 on our website:

    1. **Physically Download the Bootstrap related zip file from this link:** https://getbootstrap.com/docs/5.0/getting-started/download/

    - Here we get a zip file, now unzip this file and get the bootstrap.min.css file from the css folder and get the bootstrap.min.js file from the js folder. And include these files inside the basic template of the HTML file.
        - Tip: arrange both the files inside a separate folder, (create a new folder inside the workspace called bootstrap and place both files inside this folder).

      Example: **index.html**

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>

    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">

</head>
<body>

    <h1>Welcome to Chitkara</h1>

    <button class="btn btn-primary">Click Here</button>

    <script src="bootstrap/bootstrap.min.js"></script>
</body>
</html>
```

Note: To work with the Bootstrap properly (responsive) the <meta> tag with viewport is mandatory.
Bootstrap 5 uses JavaScript for different components like models, tooltips, sliders etc. However if we use just CSS part of the bootstrap we don't need to include the js file.

2. **Include Bootstrap 5 from the CDN:**

    - Get the Bootstrap CDN (Content Delivery Network) link from the official site: https://getbootstrap.com/docs/5.3/getting-started/download/
      And paste it inside the <head> section of the the html file:

Example: **index.html**

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
 <meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+AL
EwIH" crossorigin="anonymous">

</head>

<body>

        <h1>Welcome to Chitkara</h1>

        <button class="btn btn-primary">Click Here</button>

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.min.js"
integrity="sha384-0pUGZvbkm6XF6gxjEnlmuGrJXVbNuzT9qBBavbLwCsOGabYfZo0T0to5eqrup
tLy" crossorigin="anonymous"></script>

</body>

</html>
```

Bootstrap Breakpoints:

- **What are Breakpoints?**
  Breakpoints are specific screen widths where the layout of the website changes to optimize for different device sizes.

Bootstrap includes six default breakpoints:

Breakpoint	Class infix	Dimensions
X-Small	*None*	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

Note: When you apply a breakpoint to an element, the style will be applied **at that breakpoint and larger**. For screens **smaller** than the specified breakpoint, the style won't apply unless you use another rule for smaller screen sizes.

Containers in Bootstrap:

- Containers are a fundamental building block of Bootstrap that contain, pad, and align your content within a given device or viewport. It is the most basic element in bootstrap.
- Bootstrap requires the container element to wrap the site contents, and manage it.
- Bootstrap comes with three different containers:
    1. .container, which sets a max-width at each responsive breakpoint
    2. .container-fluid, which is width: 100% at all breakpoints
    3. .container-{breakpoint}, which is width: 100% until the specified breakpoint

The table below illustrates how each container's `max-width` compares to the original `.container` and `.container-fluid` across each breakpoint.

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
`.container`	100%	540px	720px	960px	1140px	1320px
`.container-sm`	100%	540px	720px	960px	1140px	1320px
`.container-md`	100%	100%	720px	960px	1140px	1320px
`.container-lg`	100%	100%	100%	960px	1140px	1320px
`.container-xl`	100%	100%	100%	100%	1140px	1320px
`.container-xxl`	100%	100%	100%	100%	100%	1320px
`.container-fluid`	100%	100%	100%	100%	100%	100%

Default Container:

- The default `.container` class is a **responsive, fixed-width container** whose **max-width adjusts** at different breakpoints.
- It automatically applies `margin: auto` to the left and right sides, ensuring that the container is centered on the webpage.

Syntax::

```
<div class="container">
  <!-- Content here -->
</div>
```

Example:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">

    <style>
        .cl1 {
            border: 10px solid green;
        }
    </style>


</head>

<body>

    <div class="container cl1">

        <h1>Welcome to Chitkara</h1>


    </div>

</body>

</html>
```

Output:

Welcome to Chitkara

Note: If we want to apply our own styles, it is recommended to keep it below the bootstrap css link.

Student Task: apply the class following classes to the and by minimizing the screen see the width of the container inside the developer tool.

```
<div class="container-sm">100% wide until small breakpoint</div>
<div class="container-md">100% wide until medium breakpoint</div>
<div class="container-lg">100% wide until large breakpoint</div>
<div class="container-xl">100% wide until extra large breakpoint</div>
<div class="container-xxl">100% wide until extra extra large breakpoint</div>

<div class="container-fluid">100% wide for all the breakpoints</div>
```

Bootstrap Grid System:

- Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content. It's built with flexbox and is fully responsive.
- Bootstrap provides the following main 3 classes to design any layouts:
    1. container
    2. row: Which contains the columns.
    3. col: Columns that scale according to the viewport size.

Example: Container with rows:

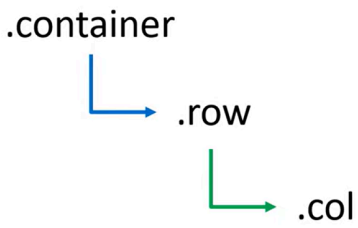
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
```

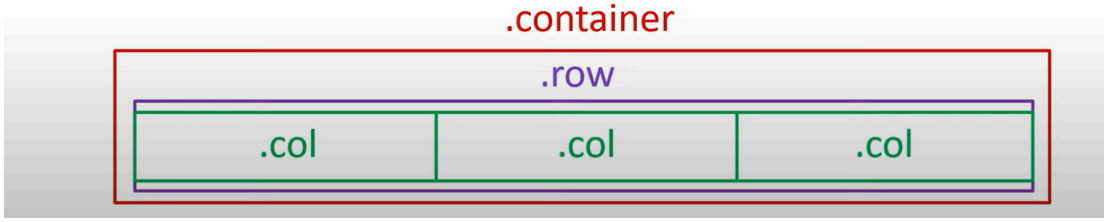
```
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">

    <style>

       body{
            background-color: gray;
        }
        .cl1 {
            border: 5px solid green;
            margin-top: 5px;
        }

        .row {
            border: 2px solid red;
        }

        .row:nth-child(1) {
            background-color: antiquewhite;
        }

        .row:nth-child(2) {
            background-color: aquamarine;
        }

        .row:nth-child(3) {
            background-color: violet;
        }

        .row:nth-child(4) {
            background-color: yellowgreen;
        }
    </style>


</head>

<body>
```

```
    <div class="container cl1">

        <div class="row">Row1</div>
        <div class="row">Row2</div>
        <div class="row">Row3</div>
        <div class="row">Row4</div>

    </div>

</body>

</html>
```

Output:

Here for the class row, automatically display: flex, and flex-wrap: wrap properties are applied. We can check on developer tool.





Creating equal width columns: By using col class.

Example:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">

    <style>

        body{
            background-color: gray;
        }
        .cl1 {
            border: 5px solid green;
            margin-top: 5px;
        }

        .row {
            border: 2px solid red;
        }

        .row:nth-child(1) {
            background-color: antiquewhite;
        }

        .row:nth-child(2) {
            background-color: aquamarine;
        }

        .row:nth-child(3) {
            background-color: violet;
        }
```

```
        .row:nth-child(4) {
            background-color: yellowgreen;
        }

        .col{
            border: 2px solid;
        }


    </style>


</head>

<body>

    <div class="container cl1">

        <div class="row">Row1</div>
        <div class="row">
            <div class="col">Column1</div>
            <div class="col">Column2</div>
        </div>
        <div class="row">Row3</div>
        <div class="row">Row4</div>

    </div>

</body>

</html>
```

Output:

We can create any number of columns inside a row, all the columns will get the equal width.
Here when the container width will become too small to fit column contents then it will automatically start wrapping.

Student Task:

- Create 5 columns inside the row4 in the above example.

Creating responsive columns with breakpoint using col class:

- We can apply breakpoints to the **col** class also for different screen sizes.

For example, **col-md:** here till the medium screen size, Inside the parent row, all the columns will take the equal amount of space, but below the medium screen, all the columns will take 100% width inside the parent row and will be wrapped line by line.

Example:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">

    <style>
        body {
            background-color: gray;
        }
```

```
        .cl1 {
            border: 5px solid green;
            margin-top: 5px;
        }

        .row {
            border: 2px solid red;
        }

        .row:nth-child(1) {
            background-color: antiquewhite;
        }

        .row:nth-child(2) {
            background-color: aquamarine;
        }

        .row:nth-child(3) {
            background-color: violet;
        }

        .row:nth-child(4) {
            background-color: yellowgreen;
        }

        .col-md {
            border: 2px solid;
        }
    </style>


</head>

<body>



    <div class="container cl1">
```

```
        <div class="row">Row1</div>
        <div class="row">
            <div class="col-md">Column1</div>
            <div class="col-md">Column2</div>
            <div class="col-md">Column3</div>
            <div class="col-md">Column4</div>
            <div class="col-md">Column5</div>

        </div>
        <div class="row">Row3</div>
        <div class="row">Row4</div>

    </div>

</body>

</html>
```

row-cols-* class:

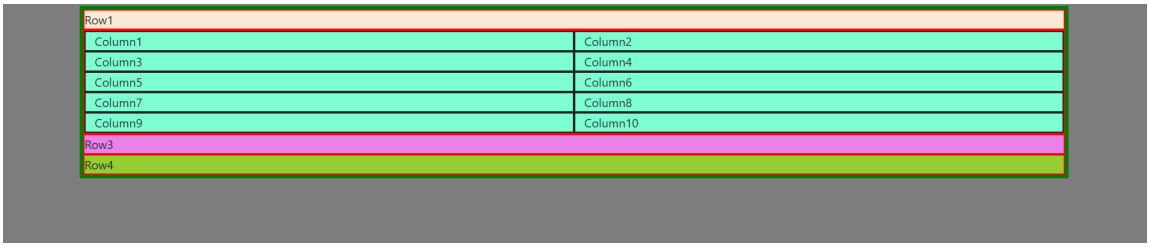
- In Bootstrap, the **row-cols** classes offer a convenient way to define the number of columns **within a single row**, facilitating the creation of responsive grid layouts without specifying explicit column widths.
- The **row-cols** classes determine how many columns will be displayed in a single row. For example, **row-cols-3** will arrange the content into three equally sized columns per row. If there are more than 3 columns, the extra columns will automatically wrap to the next row within the same parent row.
- Bootstrap provides a range of `row-cols` classes to accommodate different layouts:
    - `row-cols-1`: Single column per row.
    - `row-cols-2`: Two columns per row.
    - `row-cols-3`: Three columns per row.
    - `row-cols-4`: Four columns per row.
    - `row-cols-5`: Five columns per row.
    - `row-cols-6`: Six columns per row.
    - `row-cols-auto`: Columns will automatically adjust to fit the content.

Example:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">

    <style>

        body{
            background-color: gray;
        }
        .cl1 {
            border: 5px solid green;
            margin-top: 5px;
        }

        .row {
            border: 2px solid red;
        }

        .row:nth-child(1) {
            background-color: antiquewhite;
        }

        .row:nth-child(2) {
            background-color: aquamarine;
        }

        .row:nth-child(3) {
            background-color: violet;
        }

        .row:nth-child(4) {
            background-color: yellowgreen;
```

```
        }

        .col{
            border: 2px solid;
        }


    </style>


</head>

<body>



    <div class="container cl1">

        <div class="row">Row1</div>
        <div class="row row-cols-2">
            <div class="col">Column1</div>
            <div class="col">Column2</div>
            <div class="col">Column3</div>
            <div class="col">Column4</div>
            <div class="col">Column5</div>
            <div class="col">Column6</div>
            <div class="col">Column7</div>
            <div class="col">Column8</div>
            <div class="col">Column9</div>
            <div class="col">Column10</div>

        </div>
        <div class="row">Row3</div>
        <div class="row">Row4</div>

    </div>

</body>

</html>
```

Output:



Student Task: Add Lorem50 to column 1 and column 2 and observe the responsiveness of the webpage.

Creating responsive columns with breakpoint using row-cols class:

- We can apply breakpoints to the **row-cols** class also for different screen sizes.

For example, **row-cols-lg-2:** here till the large screen size, In side the parent row, 2 columns will be there and remaining columns will be wrapped to the next line in the same row, but below the large screen size, it will come to the default equal width column inside the parent row.

Example:

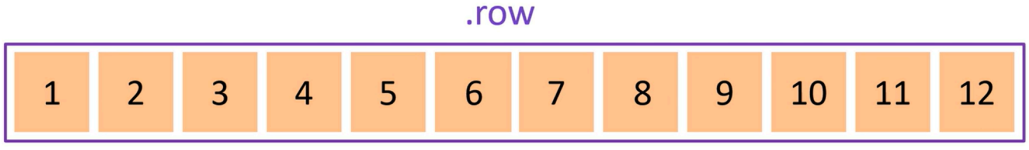
```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">
```

```
    <style>
        body {
            background-color: gray;
        }

        .cl1 {
            border: 5px solid green;
            margin-top: 5px;
        }

        .row {
            border: 2px solid red;
        }

        .row:nth-child(1) {
            background-color: antiquewhite;
        }

        .row:nth-child(2) {
            background-color: aquamarine;
        }

        .row:nth-child(3) {
            background-color: violet;
        }

        .row:nth-child(4) {
            background-color: yellowgreen;
        }

        .col {
            border: 2px solid;
        }
    </style>


</head>

<body>
```

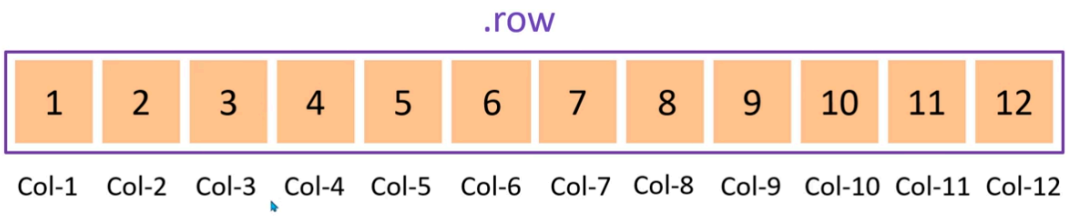
```
    <div class="container cl1">

        <div class="row">Row1</div>
        <div class="row row-cols-lg-2">
            <div class="col">Column1</div>
            <div class="col">Column2</div>
            <div class="col">Column3</div>
            <div class="col">Column4</div>
            <div class="col">Column5</div>

        </div>
        <div class="row">Row3</div>
        <div class="row">Row4</div>

    </div>

</body>

</html>
```

Here we can see the effect by minimizing or maximizing the screen width.

Note: Bootstrap 5 provides a responsive grid system based on Flexbox that allows for up to 12 columns across a page. This grid system is designed to be flexible and responsive, automatically adjusting for different screen sizes.



You can control the width of each column using Bootstrap's predefined classes, such as `col-*`, where `*` represents the number of columns you want to allocate out of the available 12.



You can merge columns to span multiple grids within the 12-column layout. For example, to create two columns where the first column occupies 8 out of 12 grids and the second column occupies 4 out of 12 grids, you can use the `col-8` and `col-4` classes, respectively.

Example:

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">

    <style>

        body{
            background-color: gray;
        }
        .cl1 {
            border: 5px solid green;
            margin-top: 5px;
        }
```

```
        .row {
            border: 2px solid red;
        }


        .row:nth-child(1) {
            background-color: antiquewhite;
        }


        .row:nth-child(2) {
            background-color: aquamarine;
        }


        .row:nth-child(3) {
            background-color: violet;
        }

        .row:nth-child(4) {
            background-color: yellowgreen;
        }

        .row > div{
            border: 2px solid;
        }

    </style>

</head>
<body>

    <div class="container cl1">

        <div class="row">Row1</div>
        <div class="row">
            <div class="col-8">Column1</div>
            <div class="col-4">Column2</div>
        </div>
        <div class="row">Row3</div>
```

```
        <div class="row">Row4</div>

    </div>
</body>

</html>
```

Output:

Behavior of Exceeding 12 Columns:

If the total number of columns in a row exceeds 12, Bootstrap will automatically wrap the extra columns to the next line within the same row. For instance, if you have `col-8`, `col-4`, and another `col-2`, the `col-2` will move to the next line.

Example:

```
<div class="row">
     <div class="col-8">Column1</div>
     <div class="col-4">Column2</div>
     <div class="col-2">Column2 (wrapped to next line)</div>
 </div>
```

In this example, the first two columns (`col-8` and `col-4`) sum up to 12, so the third column (`col-2`) is pushed to a new line within the same row.

Student Task: Create the following layout:

Solution:

```
<!DOCTYPE html>
<html lang="en">

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

integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6h
W+ALEwIH" crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle
.min.js"

integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN
7N6jIeHz"
        crossorigin="anonymous"></script>
    <style>
        .container {
            border: 10px solid red;
        }

        .row {
            border: 2px solid blue;
        }
```

```
        .row>div {
            border: 2px solid green;
        }

        .col-12>ul {
            display: flex;
            justify-content: flex-end;
            gap: 10px;
            list-style: none;
        }
    </style>
</head>

<body>


    <div class="container">

        <div class="row">

            <div class="col-12">
                <h1 class="text-center">Welcome to Chitkara</h1>
            </div>

        </div>


        <div class="row">

            <div class="col-12">

                <ul>
                    <li><a href="#">Home</a></li>
                    <li><a href="#">About Us</a></li>
                    <li><a href="#">Contact Us</a></li>
                    <li><a href="#">Products</a></li>
                </ul>

            </div>
        </div>
```

```
        <div class="row">

            <div class="col-3">

                <ul>
                    <li><a href="#">Home</a></li>
                    <li><a href="#">About Us</a></li>
                    <li><a href="#">Contact Us</a></li>
                    <li><a href="#">Products</a></li>
                </ul>
            </div>

            <div class="col-9">

                <h2>Main Part</h2>
                <p>Lorem ipsum dolor sit amet consectetur adipisicing
elit. Explicabo provident at placeat, ipsa eaque
                    non unde beatae assumenda quos. Ipsum mollitia quos
necessitatibus quis accusamus, quia repellat
                    cupiditate veritatis tenetur porro molestiae nobis
molestias, expedita unde cumque voluptatem sint
                    dolor optio? Iusto necessitatibus laudantium doloribus
voluptatibus asperiores perspiciatis,
                    similique harum!</p>

            </div>

        </div>

        <div class="row">
            <div class="col text-center">
                <p> @Copyright Info</p>
            </div>
        </div>

    </div>

</body>
```

```
</html>
```

Activity1:

Let's decorate the above layout as follows: By removing the borders and using the semantic tags with background color.

Solution:

```
<!DOCTYPE html>
<html lang="en">


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

integrity="sha384-QWTKZyjpPEjISv5WaRU9OFeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6h
W+ALEwIH" crossorigin="anonymous">
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle
.min.js"

integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDzOxhy9GkcIdslK1eN
7N6jIeHz"
        crossorigin="anonymous"></script>
```

```
    <style>
        body {
            background-color: gray;
        }

        .col-12>ul {
            display: flex;
            justify-content: flex-end;
            gap: 10px;
            list-style: none;
        }

        header {
            background-color: orange;
        }

        nav {
            background-color: antiquewhite;
        }

        aside {
            background-color: aquamarine;
        }

        main {
            background-color: azure;
        }

        footer {
            background-color: yellowgreen;
        }
    </style>
</head>


<body>
```

```
<div class="container">

    <div class="row">

        <header class="col-12">
            <h1 class="text-center">Welcome to Chitkara</h1>
        </header>

    </div>



    <div class="row">

        <nav class="col-12">

            <ul>
                <li><a href="#">Home</a></li>
                <li><a href="#">About Us</a></li>
                <li><a href="#">Contact Us</a></li>
                <li><a href="#">Products</a></li>
            </ul>

        </nav>
    </div>

    <div class="row">

        <aside class="col-3">
```

```
                <ul>
                    <li><a href="#">Home</a></li>
                    <li><a href="#">About Us</a></li>
                    <li><a href="#">Contact Us</a></li>
                    <li><a href="#">Products</a></li>
                </ul>
            </aside>


            <main class="col-9">


                <h2>Main Part</h2>
                <p>Lorem ipsum dolor sit amet consectetur adipisicing
elit. Explicabo provident at placeat, ipsa eaque
                    non unde beatae assumenda quos. Ipsum mollitia quos
necessitatibus quis accusamus, quia repellat
                    cupiditate veritatis tenetur porro molestiae nobis
molestias, expedita unde cumque voluptatem sint
                    dolor optio? Iusto necessitatibus laudantium doloribus
voluptatibus asperiores perspiciatis,
                    similique harum!</p>


            </main>


        </div>


        <div class="row">
            <footer class="col text-center">
                <p> @Copyright Info</p>
            </footer>
        </div>


    </div>
```

```
</body>


</html>
```

Activity2:

Let's apply the media query on the above layout using the breakpoints, so that once the screen size goes below the medium screen then the main part should come below the side bar.

To achieve this requirement, we need to just apply the classes inside the **<aside>** and **<main>** tag as col-md-3 and col-md-9 respectively instead of col-3 and col-9.

Note: here col-md-3 means till the medium screen this column will take 3 grid width, and below the medium screen it will take full width.

Activity3:

Let's modify the above layout for the following rules:

1. **For the Large and Extra large screen:** <aside> should take the width of 3 grid and <main> should take width of 9 grid.
2. **For the medium screen:** <aside> and <main> should take the width of 6 grid each
3. **For the small screen:** <main> should come below the <aside> element.

Solution: Apply the breakpoints for each screen in the proportion of 12 grids.

```
<aside class="col-lg-3 col-md-6">

<main class="col-lg-9 col-md-6">
```

Bootstrap 5 Offset class:

The `offset` classes in Bootstrap are used to create space or "offset" columns from the left side of the grid. This is particularly useful when you want to adjust the alignment or positioning of your columns without adding empty columns manually.

Syntax:
The offset classes follow the format:

- **`offset-{breakpoint}-{size}`**
- Where `breakpoint` is the screen size (like `sm`, `md`, `lg`, etc.), and `size` is the number of columns (1–11) by which you want to offset the column.

Example:

- `col-md-6 offset-md-3` creates a 6-column wide column that's centered by offsetting it 3 columns from the left, on medium screens and above.

```
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">


    <style>


        body{
            background-color: gray;
        }


        .container{
            background-color: orange;
```

```
            border: 10px solid red;
        }

        .row{
            background-color: antiquewhite;
        }

        .cl{
            background-color: aquamarine;
            border: 2px solid;
        }


    </style>

</head>


<body>

    <div class="container">

        <div class="row">

            <div class="col-3 offset-3 cl ">
                Lorem ipsum dolor sit amet.
            </div>

            <div class="col-3 offset-3 cl ">
                Lorem ipsum dolor sit amet.
            </div>

        </div>
    </div>

</body>

</html>
```

Here if we use offset-md-3 then below the medium screen this offset class stops working.

Reordering the columns:

Here we can use the following classes:

- order-first
- order-last
- order- * (here we can give the value from 0 to 12, higher value will become high order and smaller value will become the small value)

We can also use the breakpoints in between the order classes also.
Example:

- order-md-first
- order-md-1

Note: For ordering negative value will not work we can intermixed with order-first and order-last class with order-* class.

Example:

```
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">


    <style>


        body{
            background-color: gray;
        }
```

```
        .container{
            background-color: orange;
            border: 10px solid red;
        }

        .row{
            background-color: antiquewhite;
        }

        .cl{
            background-color: aquamarine;
            border: 2px solid;
        }


    </style>

</head>


<body>

    <div class="container">
        <div class="row">
            <div class="col-3 cl order-last">
                Column1
            </div>

            <div class="col-3 cl order-1">
                Column2
            </div>

            <div class="col-3 cl order-0">
                Column3
            </div>

            <div class="col-3 cl order-first">
                Column4
```

```
            </div>
        </div>
    </div>
</body>


</html>
```

Here if we use the breakpoint with order class, then below that breakpoint the element will come to its default order.

Example:

```
 <div class="container">
        <div class="row">
            <div class="col-3 cl order-md-last">
                Column1
            </div>


            <div class="col-3 cl order-md-1">
                Column2
            </div>


            <div class="col-3 cl order-md-0">
                Column3
            </div>


            <div class="col-3 cl order-md-first">
                Column4
            </div>
        </div>
```

Bootstrap 5 Margin and Padding classes:

In Bootstrap, spacing utilities provide a simple way to control the margin, padding of elements. These utilities are shorthand for setting the `margin`, `padding` properties using predefined classes.

1. Margin (`m`)

The margin utility classes set the space **outside** an element. Bootstrap provides a variety of margin classes to target specific sides, as well as for different breakpoints (responsive).

Syntax:

- `m`: Applies margin to all sides.
- `mt`: Applies margin to the top.
- `mb`: Applies margin to the bottom.
- `ms`: (Start) Applies margin to the left.
- `me`: (End) Applies margin to the right.
- `mx`: Applies margin to the left and right (horizontal).
- `my`: Applies margin to the top and bottom (vertical).

Values:

- `0`: Removes margin (`m-0`, `mt-0`, etc.).
- 1, 2, 3, 4, 5: Applies margin in increments based on Bootstrap's spacing scale (from smaller to larger).
- `auto`: Automatically adjusts the margin (useful for centering elements).

Example:

```
<div class="m-3">Margin on all sides</div>
<div class="mt-2">Top margin</div>
<div class="mx-auto">Horizontally centered with auto margin</div>
```

2. Padding (`p`)

The padding utility classes set the space **inside** an element (i.e., between the element content and its border).

Syntax:

- `p`: Applies padding to all sides.
- `pt`: Applies padding to the top.
- `pb`: Applies padding to the bottom.
- `ps`: (Start) Applies padding to the left.
- `pe`: (End) Applies padding to the right.
- `px`: Applies padding to the left and right (horizontal).
- `py`: Applies padding to the top and bottom (vertical).

Values:

- `0`: Removes padding (`p-0`, `pt-0`, etc.).
- `1`, `2`, `3`, `4`, `5`: Applies padding in increments based on the spacing scale.
- You can combine different padding classes for different sides.

Example:

```
<div class="p-4">Padding on all sides</div>
<div class="py-3">Padding on top and bottom</div>
<div class="px-2">Padding on left and right</div>
```

Responsive Spacing

Bootstrap allows for responsive spacing utilities by appending breakpoint prefixes to the class names.

Example:

```
  <div class="m-md-3 p-lg-5">Margin for medium and padding for large screens</div>
  <div class="mt-2 mt-lg-4">margin-top on small screens margin-top on large screens</div>
```

Bootstrap Spacing Scale:

The **spacing scale** in Bootstrap refers to a predefined set of values used for margins, padding, and gaps. It provides a consistent and responsive way to manage spacing across elements. Bootstrap uses a **0 to 5 scale**, where each number corresponds to a different amount of spacing, making it easier to apply specific spacing sizes without writing custom CSS.

Bootstrap Spacing Scale:

- **0**: `0px` (No space)
- **1**: `0.25rem` (`4px`)
- **2**: `0.5rem` (`8px`)
- **3**: `1rem` (`16px`)
- **4**: `1.5rem` (`24px`)
- **5**: `3rem` (`48px`)
- **auto**: Automatically adjusts the margin to fill available space (mainly for centering elements).

Example Classes Using the Scale:

1. **Margins** (`m`, `mt`, `mb`, etc.):
    - `m-0`: No margin (`0px`)
    - `m-1`: Small margin (`0.25rem` or `4px`)
    - `m-2`: Medium margin (`0.5rem` or `8px`)
    - `m-3`: Standard margin (`1rem` or `16px`)
    - `m-4`: Larger margin (`1.5rem` or `24px`)
    - `m-5`: Extra-large margin (`3rem` or `48px`)
    - `m-auto`: Automatically adjusts the margin
2. **Padding** (`p`, `pt`, `pb`, etc.):
    - `p-0`: No padding
    - `p-1`: Small padding (`0.25rem` or `4px`)
    - `p-2`: Medium padding (`0.5rem` or `8px`)
    - `p-3`: Standard padding (`1rem` or `16px`)
    - `p-4`: Larger padding (`1.5rem` or `24px`)
    - `p-5`: Extra-large padding (`3rem` or `48px`)

Example: Margin Auto:

```html
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">


    <style>
        body {
            background-color: gray;
        }


        .container {
            background-color: orange;
            border: 10px solid red;
        }

        .row {
            background-color: antiquewhite;
        }

        .cl {
            background-color: aquamarine;
            border: 2px solid;
        }
    </style>

</head>


<body>



    <div class="container">
```

```
        <div class="row">
            <div class="col-3 cl m-auto">Column</div>
        </div>
        <div class="row">
            <div class="col-3 cl m-lg-auto">Column</div>
        </div>
        <div class="row">
            <div class="col-3 cl ms-auto">Column</div>
        </div>

        <div class="row">
            <div class="col-3 cl me-auto">Column</div>
        </div>
    </div>

</body>


</html>
```

Example 2: with responsive breakpoint:

```
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">


    <style>
        body {
            background-color: gray;
```

```
        }


        .container {
            background-color: orange;
            border: 10px solid red;
        }

        .row {
            background-color: antiquewhite;
        }

        .cl {
            background-color: aquamarine;
            border: 2px solid;
        }
    </style>

</head>


<body>



    <div class="container">

        <div class="row">
            <div class="col-sm-2 cl me-md-auto">Lorem20</div>
            <div class="col-sm-3 cl">Lorem20</div>
        </div>

    </div>

</body>


</html>
```

Here we can apply my-5 or my-md-5 class also to give margin from top and bottom inside the first column.

Example: Using Padding

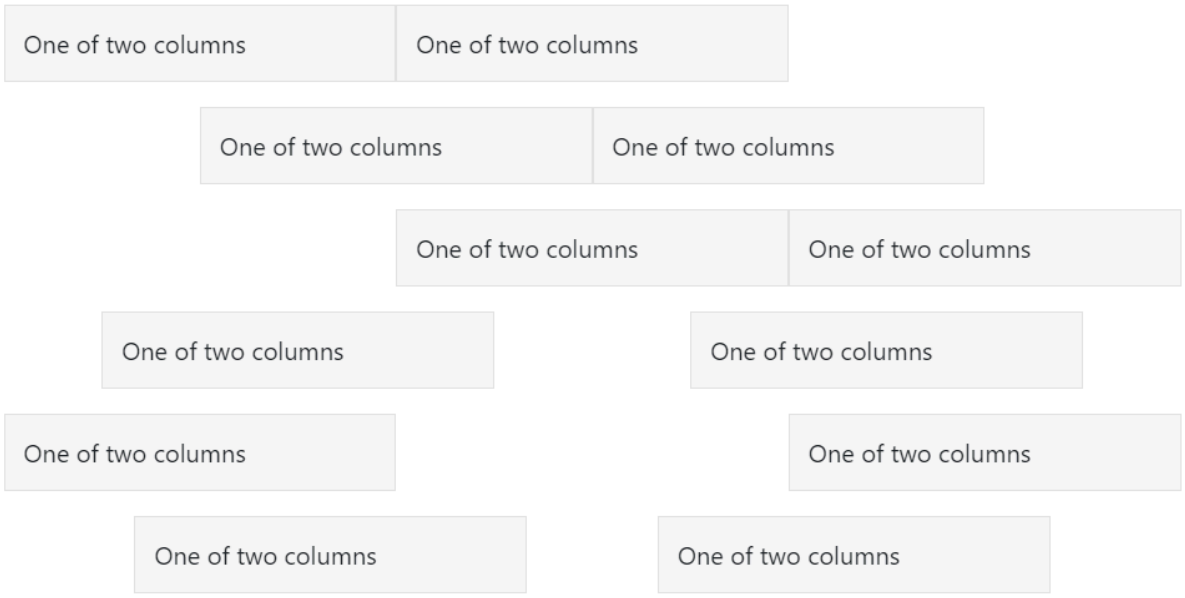
```
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">


    <style>
        body {
            background-color: gray;
        }


        .container {
            background-color: orange;
            border: 10px solid red;
        }

        .row {
            background-color: antiquewhite;
        }

        .cl {
            background-color: aquamarine;
            border: 2px solid;
        }
    </style>

</head>
```

```
<body>



    <div class="container">
        <div class="row">
            <div class="col-6 cl p-5">
                Lorem ipsum dolor sit amet consectetur adipisicing elit.
Perferendis quod placeat dicta nobis doloribus tempora eius fugit
nesciunt. Corporis, voluptates.
            </div>
            <div class="col-6 cl p-lg-5">Lorem ipsum dolor sit amet
consectetur, adipisicing elit. Libero ducimus nam unde dolorum mollitia
veniam repellat accusamus doloribus temporibus iusto.</div>
        </div>

    </div>

</body>


</html>
```

Bootstrap 5 Alignments:

- Since Bootstrap **row** class is implemented by using flexbox concept.
- We can use flexbox alignment utilities to vertically and horizontally align columns.

Horizontal alignment:



Example:

```
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">


    <style>
        body {
            background-color: gray;
        }


        .container {
            background-color: orange;
            border: 10px solid red;
```

```
        }

        .row {
            background-color: antiquewhite;
        }

        .cl {
            background-color: aquamarine;
            border: 2px solid;
        }
    </style>

</head>


<body>

    <div class="container">
        <div class="row justify-content-start">
          <div class="col-3 cl">
            One of two columns
          </div>
          <div class="col-3 cl">
            One of two columns
          </div>
        </div>
        <div class="row justify-content-center">
          <div class="col-3 cl">
            One of two columns
          </div>
          <div class="col-3 cl">
            One of two columns
          </div>
        </div>
        <div class="row justify-content-end">
          <div class="col-3 cl">
            One of two columns
          </div>
          <div class="col-3 cl">
            One of two columns
```

```
        </div>
      </div>
      <div class="row justify-content-around">
        <div class="col-3 cl">
          One of two columns
        </div>
        <div class="col-3 cl">
          One of two columns
        </div>
      </div>
      <div class="row justify-content-between">
        <div class="col-3 cl">
          One of two columns
        </div>
        <div class="col-3 cl">
          One of two columns
        </div>
      </div>
      <div class="row justify-content-evenly">
        <div class="col-3 cl">
          One of two columns
        </div>
        <div class="col-3 cl">
          One of two columns
        </div>
      </div>
    </div>
</body>


</html>
```

Here we can make use of breakpoint related classes also.

`justify-content-lg-start`

Vertical Alignment:

- Here we need to give the height for the rows

One of three columns	One of three columns	One of three columns
One of three columns	One of three columns	One of three columns
One of three columns	One of three columns	One of three columns

Example:

```
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">
```

```
<style>
    body {
        background-color: gray;
    }


    .container {
        background-color: orange;
        border: 10px solid red;
    }

    .row {
        background-color: antiquewhite;
        height: 20vh;
    }

    .row:nth-child(1){
        background-color: cyan;
    }

    .row:nth-child(2){
        background-color: yellow;
    }

    .row:nth-child(3){
        background-color: yellowgreen;
    }


    .cl {
        background-color: aquamarine;
        border: 2px solid;
```

```
    }
  </style>

</head>


<body>

  <div class="container">
    <div class="row rw align-items-start">
      <div class="col cl">
        One of three columns
      </div>
      <div class="col cl">
        One of three columns
      </div>
      <div class="col cl">
        One of three columns
      </div>
    </div>
    <div class="row rw align-items-center">
      <div class="col cl">
        One of three columns
      </div>
      <div class="col cl">
        One of three columns
      </div>
      <div class="col cl">
        One of three columns
      </div>
    </div>
    <div class="row rw align-items-end">
      <div class="col cl">
```

```
        One of three columns
      </div>
      <div class="col cl">
        One of three columns
      </div>
      <div class="col cl">
        One of three columns
      </div>
    </div>
  </div>
</body>


</html>
```

Here we can make use of breakpoint related classes also.

align-items-lg-start

Flex-direction:

We can also make use of the flex-direction related classes also for the row.

- flex-row
- flex-row-reverse
- flex-column
- flex-column-reverse

There classes can be used along with breakpoints also.

Example:

- flex-md-column

Example: till the large screen size the flex direction will be row and below the large screen the flex direction will be column.

```
<!DOCTYPE html>
<html lang="en">


<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="bootstrap/bootstrap.min.css">


    <style>
        body {
            background-color: gray;
        }


        .container {
            background-color: orange;
            border: 10px solid red;
        }


        .row:nth-child(1){
            background-color: cyan;
        }

        .row:nth-child(2){
            background-color: yellow;
        }

        .row:nth-child(3){
```

```
            background-color: yellowgreen;
        }


        .cl {
            background-color: aquamarine;
            border: 2px solid;
        }
    </style>

</head>


<body>

    <div class="container">

        <div class="row flex-lg-row flex-column ">

            <div class="col cl">Column1</div>
            <div class="col cl">Column2</div>
            <div class="col cl">Column3</div>

        </div>


    </div>
</body>


</html>
```

Bootstrap 5 Display utilities:

- Change the value of the `display` property with Bootstrap responsive display utility classes.

Syntax:

- `.d-{value}` for `xs`
- `.d-{breakpoint}-{value}` for `sm`, `md`, `lg`, `xl`, and `xxl`.

Where some of the *value* is:

- `none`
- `inline`
- `inline-block`
- `block`
- `grid`
- `flex`

Example:

```
<!DOCTYPE html>
<html lang="en">


<head>
   <meta charset="UTF-8">
   <meta name="viewport" content="width=device-width, initial-scale=1.0">
   <title>Document</title>
   <link rel="stylesheet" href="bootstrap/bootstrap.min.css">


   <style>


    div:nth-child(1){
      background-color: red;
    }

    div:nth-child(2){
```

```
      background-color: aquamarine;
    }

    div:nth-child(3){
      background-color: yellow
    }

    div:nth-child(4){
      background-color: yellowgreen;
    }


  </style>

</head>


<body>

  <div class="d-inline d-lg-block p-2">d-inline</div>
  <div class="d-inline d-lg-block p-2">d-inline</div>

  <div class="d-none d-lg-block p-2">d-inline</div>
  <div class="d-none d-lg-block p-2">d-inline</div>


</body>


</html>
```

Bootstrap 5 Color and Background color classes:

Color classes:

.text-primary

.text-secondary

.text-success

.text-danger

.text-warning

.text-info

.text-light

.text-dark

.text-body

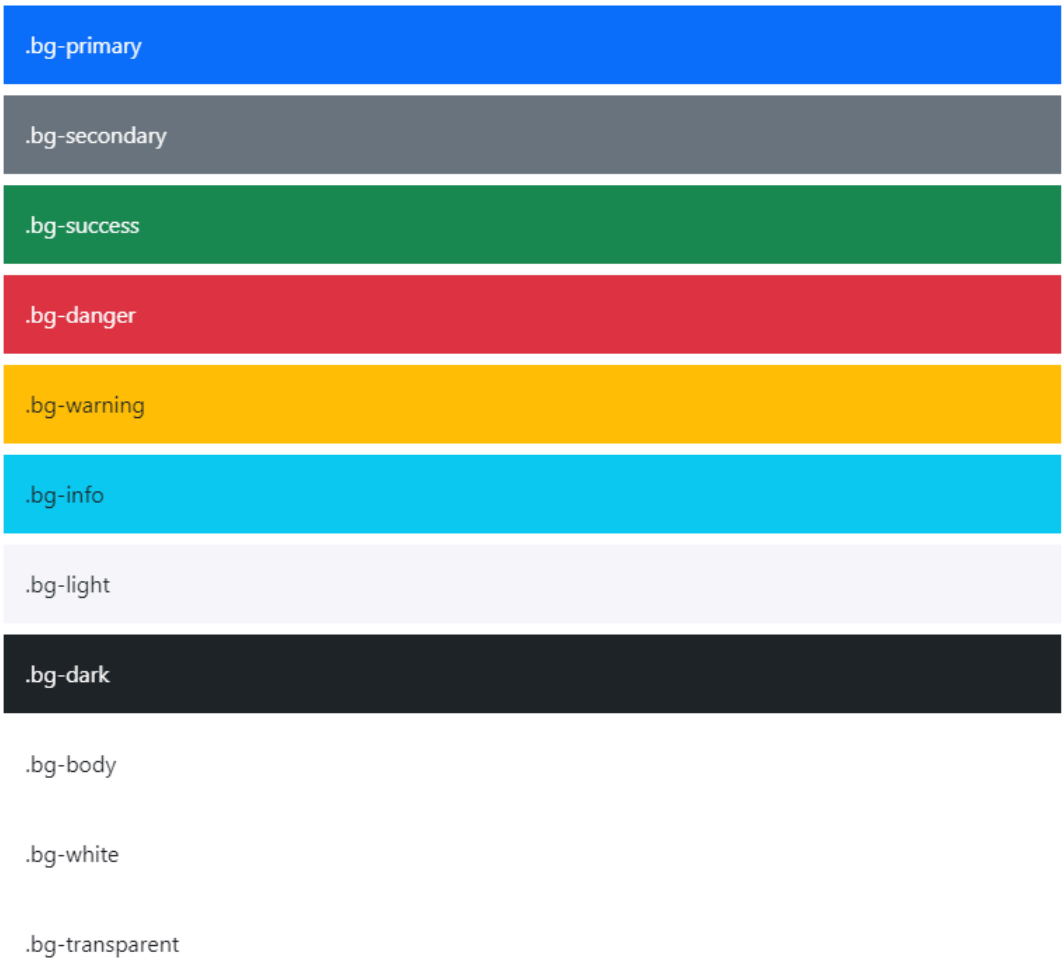
.text-muted

.text-white

.text-black-50

.text-white-50

Background color classes:



Example:

```
<div class="p-3 mb-2 bg-primary text-white">.bg-primary</div>
<div class="p-3 mb-2 bg-secondary text-white">.bg-secondary</div>
<div class="p-3 mb-2 bg-success text-white">.bg-success</div>
<div class="p-3 mb-2 bg-danger text-white">.bg-danger</div>
<div class="p-3 mb-2 bg-warning text-dark">.bg-warning</div>
<div class="p-3 mb-2 bg-info text-dark">.bg-info</div>
<div class="p-3 mb-2 bg-light text-dark">.bg-lightttttt</div>
<div class="p-3 mb-2 bg-dark text-white">.bg-dark</div>
<div class="p-3 mb-2 bg-body text-dark">.bg-body</div>
<div class="p-3 mb-2 bg-white text-dark">.bg-white</div>
<div class="p-3 mb-2 bg-transparent text-dark">.bg-transparent</div>
```

Bootstrap 5 Text properties:

1. **Text-alignment:**

   - text-start
   - text-center
   - text-end

   Responsive classes:

   - text-md-start
   - text-lg-center
   - text--sm-end

   Example:

```
<p class="text-start">Start aligned text on all viewport sizes.</p>
<p class="text-center">Center aligned text on all viewport sizes.</p>
<p class="text-end">End aligned text on all viewport sizes.</p>

<p class="text-sm-start">Start aligned text on viewports sized SM (small) or wider.</p>
<p class="text-md-start">Start aligned text on viewports sized MD (medium) or wider.</p>
<p class="text-lg-start">Start aligned text on viewports sized LG (large) or wider.</p>
<p class="text-xl-start">Start aligned text on viewports sized XL (extra-large) or wider.</p>
```

2. **Text-transform:** here we don't have the responsive classes.

```
<p class="text-lowercase">Lowercase text.</p>
<p class="text-uppercase">Uppercased text.</p>
<p class="text-capitalize">CapiTaliZed text.</p>
```

3. **Text-decoration:**

```
<p class="text-decoration-underline">This text has a line underneath it.</p>
<p class="text-decoration-line-through">This text has a line going through it.</p>
<a href="#" class="text-decoration-none">This link has its text decoration removed</a>
```

Bootstrap 5 typography:

1. **h1 to h6 classes:**

- For all the HTML heading elements like from <h1> to <h6> in Bootstrap we have **.h1** to **.h6** classes are also available.

  Example:

```
<p class="h1">h1. Bootstrap heading</p>
<p class="h2">h2. Bootstrap heading</p>
<p class="h3">h3. Bootstrap heading</p>
<p class="h4">h4. Bootstrap heading</p>
<p class="h5">h5. Bootstrap heading</p>
<p class="h6">h6. Bootstrap heading</p>
```

2. **Display classes:**

- Traditional heading elements are designed to work best in the meat of your page content. When you need a heading to stand out, consider using a display heading—a larger, slightly more opinionated heading style.

  Example:

```
<h1 class="display-1">Display 1</h1>
<h1 class="display-2">Display 2</h1>
<h1 class="display-3">Display 3</h1>
<h1 class="display-4">Display 4</h1>
<h1 class="display-5">Display 5</h1>
<h1 class="display-6">Display 6</h1>
```

  Display-font-sizes:

  1: 5rem: 80px
  2: 4.5rem: 72px
  3: 4rem: 64px
  4: 3.5rem: 56px

5: 3rem: 48px
6: 2.5rem: 40px

3. **Lead:**

- It stands out from regular paragraphs.

  Example:

  Lorem ipsum, dolor sit amet.

4. **Unstyled the List:**

- list -unstyled: Remove the default list-style and left margin on list items

  Example:

  <ul class="list-unstyled">

Home
ContactUs
AboutUs
Logout

- list-inline and list-inline-item: Remove a list's bullets and apply some light margin with a combination of these two classes.

  Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Viewport Height Example</title>
```

```
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.
css" rel="stylesheet">

    <style>

       .list-inline-item a{
            text-decoration: none;
        }

    </style>


</head>
<body>

       <ul class="list-inline">
            <li class="list-inline-item"><a href="#">Home</a></li>
            <li class="list-inline-item"><a href="#">Contact Us</a></li>
            <li class="list-inline-item"><a href="#">About Us</a></li>
            <li class="list-inline-item"><a href="#">Logout</a></li>
        </ul>


</body>
</html>
```

Bootstrap 5 Button classes:

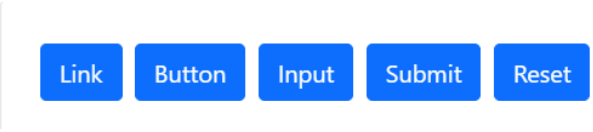
- Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control.


```
<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>

<button type="button" class="btn btn-link">Link</button>
```

Button Tags:

- The .btn classes are designed to be used with the <button> element. However, you can also use these classes on <a> or <input> elements (though some browsers may apply a slightly different rendering).




```
<a class="btn btn-primary" href="#" role="button">Link</a>
<button class="btn btn-primary" type="submit">Button</button>
<input class="btn btn-primary" type="button" value="Input">
<input class="btn btn-primary" type="submit" value="Submit">
<input class="btn btn-primary" type="reset" value="Reset">
```

Outline Buttons:

- In need of a button, but not the hefty background colors they bring? Replace the default modifier classes with the .btn-outline-* ones to remove all background images and colors on any button.


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

Button sizes:

- Fancy larger or smaller buttons? Add .btn-lg or .btn-sm for additional sizes.

```
<button type="button" class="btn btn-primary btn-lg">Large button</button>
<button type="button" class="btn btn-primary btn-sm">Small button</button>
```

Disabling the Button:

- Use the disabled attribute

```
<button type="button" class="btn btn-lg btn-primary" disabled>Primary button</button>
```

Disabling the <a> tag button:

- Use the disabled class.

```
<a href="#" class="btn btn-primary btn-lg disabled" >Primary link</a>
```

Block Button:

- Create responsive stacks of full-width, “block buttons” with a mix of our display and gap utilities.

```
    <div class="container">

        <div class="d-grid gap-2">
            <button class="btn btn-primary"
type="button">Button</button>
            <button class="btn btn-primary"
type="button">Button</button>
        </div>


        <div class="d-grid gap-2 d-md-block mt-2">
            <button class="btn btn-primary"
type="button">Button</button>
            <button class="btn btn-primary"
type="button">Button</button>
        </div>


    </div>
```

Bootstrap 5 Table classes:

- Use the **table** class

```
<table class="table">

    <thead>
        <tr>
            <th>Roll</th>
            <th>Name</th>
            <th>Marks</th>
        </tr>
    </thead>

    <tbody>
        <tr>
```

```
            <td>10</td>
            <td>Ram</td>
            <td>500</td>
        </tr>

        <tr>
            <td>10</td>
            <td>Ram</td>
            <td>500</td>
        </tr>

        <tr>
            <td>10</td>
            <td>Ram</td>
            <td>500</td>
        </tr>
        <tr>
            <td>10</td>
            <td>Ram</td>
            <td>500</td>
        </tr>
    </tbody>

</table>
```

For the Dark table use the class table-dark color, along with the table class:

```
<table class="table table-dark">
```

Student Task:

- **Create a container and out the above table inside 2 columns of equal width with table-dark and a normal table.**

Solution: Given Template:

```
<div class="container">
    <div class="row">
        <div class="col">
        </div>
        <div class="col">
        </div>
    </div>
</div>
```

For More details about tables refer to the Bootstrap docs.

Bootstrap 5 Form class:

- Refer to the Bootstrap docs

Bootstrap 5 Components:

- Refer to the Bootstrap docs.