

Lab 9 Basic of Bootstrap and Grid

```
1.<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />
    <link rel="stylesheet" href="css/bootstrap.css" />
    <title>Modified Grid Layout</title>
    <style>
      .col {
        padding: 1rem;
        border: 2px solid #fff;
        color: #fff;
        text-align: center;
      }

      .bg-1 {
        background-color: #817d7df8;
      }
      .bg-2 {
        background-color: #868788;
      }
      .bg-3 {
        background-color: #939494;
      }
      .bg-4 {
        background-color: #87878a;
      }
      .bg-5 {
        background-color: #9e9997;
      }
      .bg-6 {
        background-color: #607d8b;
      }
    </style>
  </head>

  <body>
    <div class="container">
      <h2 class="mt-4">◆ First Row: 3 Equal Columns</h2>
```

```

<div class="row">
    <div class="col bg-1">Column 1</div>
    <div class="col bg-2">Column 2</div>
    <div class="col bg-3">Column 3</div>
</div>

<h2 class="mt-4">◆ Second Row: 2 Equal Columns</h2>
<div class="row">
    <div class="col bg-4" color="red">Column A</div>
    <div class="col bg-5">Column B</div>
</div>

<h2 class="mt-4">◆ Third Row: Full Width Alert</h2>
<div class="row">
    <div class="col bg-6">
        <div class="alert alert-warning">
            This is a full-width column with an alert box. Resize the
            browser to
            see responsive behavior!
        </div>
    </div>
</div>

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

```

O/P:

◆ First Row: 3 Equal Columns

Column 1	Column 2	Column 3
----------	----------	----------

◆ Second Row: 2 Equal Columns

Column A	Column B
----------	----------

◆ Third Row: Full Width Alert

This is a full-width column with an alert box. Resize the browser to see responsive behavior!

```

2.<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet" href="css/bootstrap.css">
    <title>Hello World!</title>
</head>
<body>
    <h1>Hello World!</h1>
    <div class="alert alert-primary">
        A simple primary alert—check it out!
    </div>
    <!-- <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/2.9.3/umd/popper.min.js"
integrity="sha512-XLo6bQe08irJObCc86rFEKQdcFYbGGIHVXcfMsxpbvF8ompmd1SNJjqVY5hmjQ01Ts0UmmSQGfqpt3fGjm6pGA==" crossorigin="anonymous"
referrerPolicy="no-referrer"></script> -->
    <script src="js/bootstrap.js"></script>
</body>
</html>

```

O/P:

Hello World!

A simple primary alert
—check it out!

3.

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta http-equiv="X-UA-Compatible" content="IE=edge" />
    <meta name="viewport" content="width=device-width,
initial-scale=1.0" />

```

```
<!-- Using the responsive meta viewport tag above simply means that  
the browser will render the width of the webpage with the size of the  
viewport where the webpage is currently being viewed. -->  
<link rel="stylesheet" href="css/bootstrap.css" />  
<title>Equal Width Simple Grid</title>  
<style>  
[class*="col"] {  
    padding: 1rem;  
    background-color: #e55d33;  
    border: 2px solid #fff;  
    color: #fff;  
}  
</style>  
</head>  
  
<body>  
    <div class="container">  
        <div class="row">  
            <div class="col">Hello</div>  
            <div class="col">Dev</div>  
        </div>  
        <div class="row">  
            <div class="col">Ops</div>  
        </div>  
        <div class="row">  
            <div class="col">  
                <div class="alert alert-primary">  
                    Here are two grid layouts that apply to every device and  
viewport,  
                    from xs to xl. Add any number of unit-less classes for each  
breakpoint you need and every column will be the same  
width.  
                </div>  
            </div>  
        </div>  
    </div>  
    <!-- <script  
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/2.9.3/umd/popper.  
min.js"  
  
integrity="sha512-XLo6bQe08irJObCc86rFEKQdcFYbGGIHVXcfMsxpbvF8ompmd1SNJ  
jqVY5hmjQ01Ts0UmmSQGfqpt3fGjm6pGA=="  
crossorigin="anonymous" referrerpolicy="no-referrer"></script> -->
```

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

O/P:

Hello	Dev
Ops	
<p>Here are two grid layouts that apply to every device and viewport, from xs to xl. Add any number of unit-less classes for each breakpoint you need and every column will be the same width.</p>	

```
4.<!-- Width of the columns can be restricted to the actual content using ".col-{breakpoint}-auto" class, for example, ".col-md-auto". -->
<!-- The middle columns in this example is automatically adjusted to the width of the content. -->
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet" href="css/bootstrap.css">
    <title>Variable Width Content</title>
    <style>
        [class*="col"] {
            padding: 1rem;
            background-color: #000000;
            border: 2px solid #fff;
            color: #fff;
        }
    </style>
</head>

<body>
```

```
<div class="container">
  <div class="row justify-content-md-center">
    <div class="col col-lg-2">
      .col .col-lg-2
    </div>
    <div class="col-md-auto">
      .col-md-auto (variable width)
    </div>
    <div class="col col-lg-2">
      .col .col-lg-2
    </div>
  </div>
  <div class="row">
    <div class="col">
      .col
    </div>
    <div class="col-md-auto">
      .col-md-auto (variable width)
    </div>
    <div class="col col-lg-2">
      .col .col-lg-2
    </div>
  </div>
  <div class="row">
    <div class="col">
      <div class="alert alert-primary">
        Use col-{breakpoint}-auto classes to size columns
        based on the natural width of their content.
      </div>
    </div>
  </div>
<!-- <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/2.9.3/umd/popper.min.js"
integrity="sha512-XLo6bQe08irJObCc86rFEKQdcFYbGGIHVXcfMsxpbvF8ompmd1SNJjqVY5hmjQ01Ts0UmmSQGfqpt3fGjm6pGA=="
crossorigin="anonymous" referrerPolicy="no-referrer"></script>
-->
<script src="js/bootstrap.js"></script>
</body>
```

```
</html>
```

O/P:

.col .col-lg-2	.col-md-auto (variable width)	.col .col-lg-2
.col	.col-md-auto (variable width)	.col .col-lg-2
Use col-{breakpoint}-auto classes to size columns based on the natural width of their content.		

5.

```
<!-- Bootstrap grid system allows you to define the width for one column and the remaining columns within the same row will automatically adjusted. For example, when you define "col + col-6 + col" in a row then it will be automatically divided like "1/3+1/6+1/3" columns. Below are two examples for defining one column and then the remaining two other columns will be adjusted accordingly. -->
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet" href="css/bootstrap.css">
    <title>One Column Width and Resize Other Columns</title>
    <style>
        [class*="col"] {
            padding: 1rem;
            background-color: #e533cd;
            border: 2px solid #fff;
            color: #fff;
        }
    </style>
</head>

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

```
<div class="row">
    <div class="col">
        Unne
    </div>
    <div class="col-6">
        cess
    </div>
    <div class="col">
        ary
    </div>
</div>
<div class="row">
    <div class="col">
        .col
    </div>
    <div class="col-5">
        tags
    </div>
    <div class="col">
        in file
    </div>
</div>
<div class="row">
    <div class="col">
        <div class="alert alert-primary">
            Auto-layout for flexbox grid columns also means you
            can set the width of one column and have the sibling
            columns automatically resize around it.
        </div>
    </div>
</div>
<!-- <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/2.9.3/umd/popper.min.js"
integrity="sha512-XLo6bQe08irJObCc86rFEKQdcFYbGGIHVXcfMsxpbvF8ompmd1SNJjqVY5hmjQ01Ts0UmmSQGfqpt3fGjm6pGA=="
crossorigin="anonymous" referrerPolicy="no-referrer"></script>
-->
<script src="js/bootstrap.js"></script>
</body>
```

```
</html>
```

O/P:

Unne	cess	ary
.col	tags	in file
Auto-layout for flexbox grid columns also means you can set the width of one column and have the sibling columns automatically resize around it.		

6.<!-- Simply create a multiple columns layout within a row using ".col" classes continuously. You just need to insert ".w-100" class where a breakpoint is required. -->

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

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <link rel="stylesheet" href="css/bootstrap.css">
    <title>Equal Width Multiple Rows Grid</title>
    <style>
        [class*="col"] {
            padding: 1rem;
            background-color: #33b5e5;
            border: 2px solid #fff;
            color: #fff;
        }
    </style>
</head>

<body>
    <div class="container">
        <div class="row">
            <div class="col">.col</div>
            <div class="col">.col</div>
            <div class="w-100"></div>
        </div>
    </div>
</body>
```

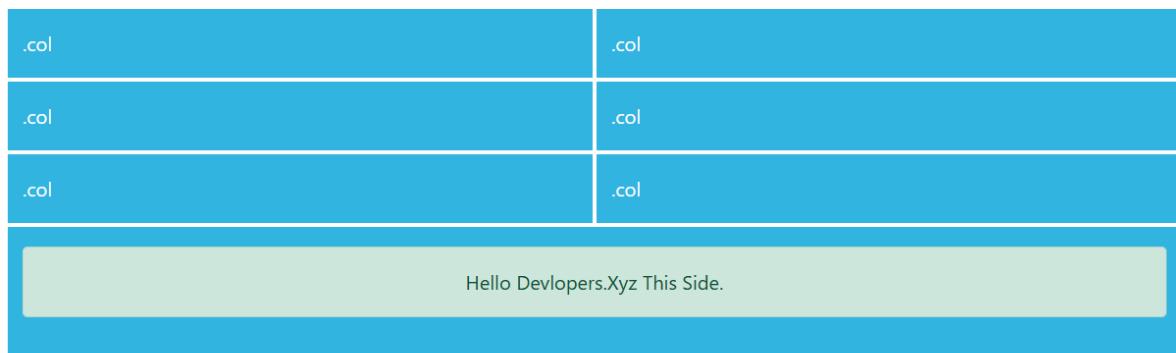
```

<div class="col">.col</div>
<div class="col">.col</div>
<div class="w-100"></div>
<div class="col">.col</div>
<div class="col">.col</div>
</div>
<div class="row">
    <div class="col">
        <div class="alert alert-success">
            Hello Developers.Xyz This Side.
        </div>
    </div>
</div>
<!-- <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/2.9.3/umd/popper.min.js"

integrity="sha512-XLo6bQe08irJObCc86rFEKQdcFYbGGIHVXcfMsxpbvF8ompmd1SNJjqVY5hmjQ01Ts0UmmSQGfqpt3fGjm6pGA=="
crossorigin="anonymous" referrerpolicy="no-referrer"></script>
-->
    <script src="js/bootstrap.js"></script>
</body>

</html>

```



7.<!-- You can create stacked grid layout using ".col-sm-" classes which will be expanded to horizontal grid on desktop. -->

```

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

<head>

```

```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<link rel="stylesheet" href="css/bootstrap.css">
<title>Stacked Horizontal Grid</title>
<style>
    [class*="col"] {
        padding: 1rem;
        background-color: #33b5e5;
        border: 2px solid #fff;
        color: #fff;
    }
</style>
</head>

<body>
    <div class="container">
        <div class="row">
            <div class="col-sm-8".col-sm-8</div>
            <div class="col-sm-4".col-sm-4</div>
        </div>
        <div class="row">
            <div class="col-sm".col-sm</div>
            <div class="col-sm".col-sm</div>
            <div class="col-sm".col-sm</div>
        </div>
        <div class="row">
            <div class="col">
                <div class="alert alert-primary">
                    Using a single set of .col-sm-* classes, you can
                    create a basic grid system that starts out stacked
                    before becoming horizontal with at the small
                    breakpoint (sm).
                </div>
            </div>
        </div>
    </div>
    <!-- <script
src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/2.9.3/umd/popper.
min.js">
```

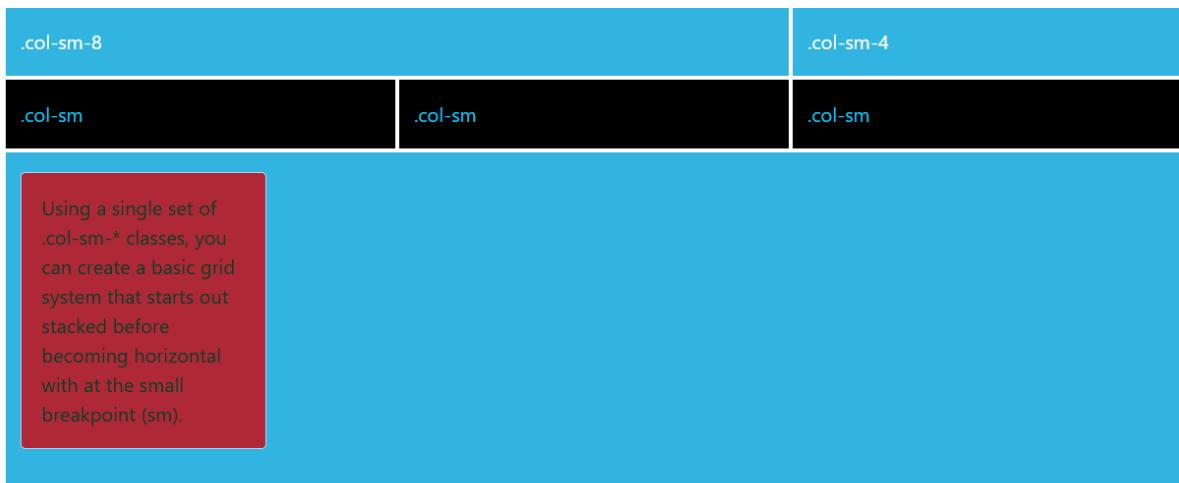
```

integrity="sha512-XLo6bQe08irJObCc86rFEKQdcFYbGGIHVXcfMsxpbvF8ompmd1SNJ
jqVY5hmjQ01Ts0UmmSQGfqpt3fGjm6pGA=="
    crossorigin="anonymous" referrerpolicy="no-referrer">></script>
-->
<script src="js/bootstrap.js"></script>
</body>

</html>

```

O/P:



8. Wrapped Column Grid.

Code:

```

<!DOCTYPE html>

<html lang="en">

    <head>

        <meta charset="UTF-8">

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width,
initial-scale=1.0">

```

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

<title>Wrapped Column Grid</title>

<style>

[class*="col"] {

    padding: 1rem;

    background-color: #33b5e5;

    border: 2px solid #fff;

    color: #fff;

    text-align: center; /* Center text */

    border-radius: 0.5rem; /* Rounded corners */

    box-shadow: 1px 1px 5px rgba(0, 0, 0, 0.2); /* Subtle shadow
*/



}

</style>

</head>



<body class="bg-light">

    <div class="container mt-4">

        <h3 class="text-center mb-4 text-primary">Wrapped Column Grid Example</h3>

        <div class="row mb-3">

            <div class="col-9">.col-9 (Row 1)</div>

            <div class="col-2">.col-2 (Row 1)</div>

            <div class="col-4">.col-4 (Wrapped Row 2)</div>

            <div class="col-5">.col-5 (Wrapped Row 2)</div>

        </div>

    </div>

</body>
```

```
</div>

<div class="row">
    <div class="col">
        <div class="alert alert-info">
             Columns exceeding 12 in a single row automatically wrap to a new line. This ensures responsive alignment without breaking the layout.
        </div>
    </div>
</div>

<script src="js/bootstrap.js"></script>

</body>

</html>
```

Output:

Wrapped Column Grid Example

The example illustrates a wrapped column grid layout. It consists of two rows of columns. Row 1 contains a single column with class .col-9, which has the label ".col-9 (Row 1)". To its right is another column with class .col-2, which has the label ".col-2 (Row 1)". Row 2 contains two columns: one with class .col-4 labeled ".col-4 (Wrapped Row 2)" and another with class .col-5 labeled ".col-5 (Wrapped Row 2)". A callout box with a light blue background and a white border contains the text: "💡 Columns exceeding 12 in a single row automatically wrap to a new line. This ensures responsive alignment without breaking the layout."