**What is Bootstrap Grid System?**

Bootstrap grid system provides an easy and powerful way to create responsive layouts of all shapes and sizes. It is built with [flexbox](https://www.tutorialrepublic.com/css-tutorial/css3-flexible-box-layouts.php) with mobile-first approach. Also, it is fully responsive and uses twelve column system (12 columns available per row) and six default responsive tiers.

You can use the Bootstrap's predefined grid classes for quickly making the layouts for different types of devices like mobile phones, tablets, laptops, desktops, and so on. For example, you can use the .col-\* classes to create grid columns for extra small devices like mobile phones in portrait mode, and the .col-sm-\* classes for mobile phones in landscape mode.

Similarly, you can use the .col-md-\* classes to create grid columns for medium screen devices like tablets, the .col-lg-\* classes for devices like small laptops, the .col-xl-\* classes for laptops and desktops, and the .col-xxl-\* classes for large desktop screens.

The following table summarizes the key features of the Bootstrap's grid system.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Features**  **Bootstrap  Grid System** | **X-Small (xs)**  **<576px** | **Small (sm)**  **≥576px** | **Medium (md)**  **≥768px** | **Large (lg)**  **≥992px** | **X-Large (xl)**  **≥1200px** | **XX-Large (xxl)**  **≥1400px** |
| Container max-width | None (auto) | 540px | 720px | 960px | 1140px | 1320px |
| Class prefix | .col- | .col-sm- | .col-md- | .col-lg- | .col-xl- | .col-xxl- |

## Creating Two Column Layouts

The following example will show you how to create two column layouts for medium, large and extra large devices like tables, laptops and desktops etc. However, on mobile phones (screen width less than 768px), the columns will automatically become horizontal (2 rows, 1 column).

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Two Column Grid Layouts for Tablets and Desktops</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

<style>

/\* Some custom styles to beautify this example \*/

.demo-content{

padding: 15px;

font-size: 18px;

background: #dbdfe5;

margin-bottom: 15px;

}

.demo-content.bg-alt{

background: #abb1b8;

}

</style>

</head>

<body>

<h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>

<div class="text-center my-3">Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.</div>

<div class="container">

<!--Row with two equal columns-->

<div class="row">

<div class="col-md-6">

<div class="demo-content">.col-md-6</div>

</div>

<div class="col-md-6">

<div class="demo-content bg-alt">.col-md-6</div>

</div>

</div>

<!--Row with two columns divided in 1:2 ratio-->

<div class="row">

<div class="col-md-4">

<div class="demo-content">.col-md-4</div>

</div>

<div class="col-md-8">

<div class="demo-content bg-alt">.col-md-8</div>

</div>

</div>

<!--Row with two columns divided in 1:3 ratio-->

<div class="row">

<div class="col-md-3">

<div class="demo-content">.col-md-3</div>

</div>

<div class="col-md-9">

<div class="demo-content bg-alt">.col-md-9</div>

</div>

</div>

</div>

</body>

</html>

**Note:** In a grid layout, content must be placed inside the columns (.col and .col-\*) and only columns may be the immediate children of rows (.row). Also, rows should be placed inside a [container (either fixed or fluid)](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-containers.php) for proper padding and alignment.

Since the Bootstrap grid system is based on 12 columns, therefore to keep the columns in a one line (i.e. side by side), the sum of the grid column numbers within a single row should not be greater than 12. If you go through the above example code carefully you will find the numbers of grid columns (i.e. col-md-\*) add up to twelve (6+6, 4+8 and 3+9) for every row.

!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Three Column Grid Layouts for Tablets (landscape) and Desktops</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

<style>

/\* Some custom styles to beautify this example \*/

.demo-content{

padding: 15px;

font-size: 18px;

background: #dbdfe5;

margin-bottom: 15px;

}

.demo-content.bg-alt{

background: #abb1b8;

}

</style>

</head>

<body>

<h2 class="text-center mt-3">Bootstrap Responsive Layout</h2>

<div class="text-center my-3">Open the output in a new blank tab (Click the arrow next to "Show Output" button) and resize the browser window to understand how the Bootstrap responsive grid system works.</div>

<div class="container">

<!--Row with three equal columns-->

<div class="row">

<div class="col-lg-4">

<div class="demo-content">.col-lg-4</div>

</div>

<div class="col-lg-4">

<div class="demo-content bg-alt">.col-lg-4</div>

</div>

<div class="col-lg-4">

<div class="demo-content">.col-lg-4</div>

</div>

</div>

<!--Row with three columns divided in 1:4:1 ratio-->

<div class="row">

<div class="col-lg-2">

<div class="demo-content">.col-lg-2</div>

</div>

<div class="col-lg-8">

<div class="demo-content bg-alt">.col-lg-8</div>

</div>

<div class="col-lg-2">

<div class="demo-content">.col-lg-2</div>

</div>

</div>

<!--Row with three columns divided unevenly-->

<div class="row">

<div class="col-lg-3">

<div class="demo-content">.col-lg-3</div>

</div>

<div class="col-lg-7">

<div class="demo-content bg-alt">.col-lg-7</div>

</div>

<div class="col-lg-2">

<div class="demo-content">.col-lg-2</div>

</div>

</div>

</div>

</body>

</html>

## Bootstrap Auto-layout Columns

You can also create equal width columns for all devices (x-small, small, medium, large, x-large, and xx-large) through simply using the class .col, without specifying any column number.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Auto-layout Columns</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

<style>

/\* Some custom styles to beautify this example \*/

.row{

margin-top: 1rem;

background: #dbdfe5;

}

.col{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<!--Row with two equal columns-->

<div class="row">

<div class="col">.col</div>

<div class="col">.col</div>

</div>

<!--Row with three equal columns-->

<div class="row">

<div class="col">.col</div>

<div class="col">.col</div>

<div class="col">.col</div>

</div>

</div>

</body>

</html>

**Creating Multi-Column Layouts with Bootstrap**

With the new Bootstrap mobile first flexbox grid system you can easily control how your website layout will render on different types of devices that have different screen or viewport sizes like mobile phones, tablets, desktops, etc. Let's consider the following illustration.



In the above illustration there are total 12 content boxes in all devices, but its placement varies according to the device screen size, like in mobile device the layout is rendered as one column grid layout which has 1 column and 12 rows placed above one another, whereas in tablet it is rendered as two column grid layout which has 2 columns and 6 rows.

Further, in large screen size devices like laptops and desktops it is rendered as three column grid layout which has 3 columns and 4 rows and finally in extra large screen devices like large desktops it is rendered as four column grid layout which has 4 columns and 3 rows.

## Alignment of Grid Columns

You can use the flexbox alignment utilities to vertically and horizontally align grid columns inside a container.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Vertical Alignments of Grid Columns</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

<style>

/\* Some custom styles to beautify this example \*/

.row{

min-height: 10rem;

margin-top: 1rem;

background: #dbdfe5;

}

.col{

padding: 10px 15px;

background: #abb1b8;

border: 1px solid rgba(0,0,0,0.5)

}

</style>

</head>

<body>

<div class="container">

<div class="row align-items-start">

<div class="col">Column one</div>

<div class="col">Column two</div>

<div class="col">Column three</div>

</div>

<div class="row align-items-center">

<div class="col">Column one</div>

<div class="col">Column two</div>

<div class="col">Column three</div>

</div>

<div class="row align-items-end">

<div class="col">Column one</div>

<div class="col">Column two</div>

<div class="col">Column three</div>

</div>

</div>

</body>

</html>

# Bootstrap Typography

You can define all [HTML headings](https://www.tutorialrepublic.com/html-tutorial/html-headings.php), <h1> through <h6> — In the same way you define in simple HTML document. You can also utilize the heading classes .h1 through .h6 on other elements, if you want to apply the style on element's text same as headings.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Headings</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="m-4">

<h1>h1. Bootstrap heading</h1>

<h2>h2. Bootstrap heading</h2>

<h3>h3. Bootstrap heading</h3>

<h4>h4. Bootstrap heading</h4>

<h5>h5. Bootstrap heading</h5>

<h6>h6. Bootstrap heading</h6>

</div>

</body>

</html>

## Display Headings

Bootstrap also provides display headings that can be used when you need a heading to stand out. Display headings are displayed in larger [font-size](https://www.tutorialrepublic.com/css-reference/css-font-size-property.php) but lighter [font-weight](https://www.tutorialrepublic.com/css-reference/css-font-weight-property.php).

Six different display headings are available. Here's is an example:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Display Headings</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="m-4">

<h1 class="display-1">Display Heading 1</h1>

<h1 class="display-2">Display Heading 2</h1>

<h1 class="display-3">Display Heading 3</h1>

<h1 class="display-4">Display Heading 4</h1>

<h1 class="display-5">Display Heading 5</h1>

<h1 class="display-6">Display Heading 6</h1>

</div>

</body>

</html>

## What is Table?

The HTML tables are used to present data in grid manner like row and columns. Using Bootstrap you can greatly improve the appearance of table in a quick and easy way.

you can create tables with basic styling that has horizontal dividers and small cell padding (8px by default), by just adding the Bootstrap's class .table to the [<table>](https://www.tutorialrepublic.com/html-reference/html-table-tag.php) element.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Simple Table</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="m-4">

<table class="table">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

<td>clarkkent@mail.com</td>

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

<td>peterparker@mail.com</td>

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

<td>johncarter@mail.com</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>

## Creating Accented Tables

Bootstrap even provides a handful of contextual classes such as .table-primary, .table-secondary, .table-success, .table-danger, .table-warning, .table-info, .table-light and .table-dark to color tables, table rows or individual cells.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta name="viewport" content="width=device-width, initial-scale=1">

<title>Bootstrap Dark Table</title>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet">

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>

<div class="m-4">

<table class="table table-dark">

<thead>

<tr>

<th>#</th>

<th>First Name</th>

<th>Last Name</th>

<th>Email</th>

</tr>

</thead>

<tbody>

<tr>

<td>1</td>

<td>Clark</td>

<td>Kent</td>

<td>clarkkent@mail.com</td>

</tr>

<tr>

<td>2</td>

<td>Peter</td>

<td>Parker</td>

<td>peterparker@mail.com</td>

</tr>

<tr>

<td>3</td>

<td>John</td>

<td>Carter</td>

<td>johncarter@mail.com</td>

</tr>

</tbody>

</table>

</div>

</body>

</html>