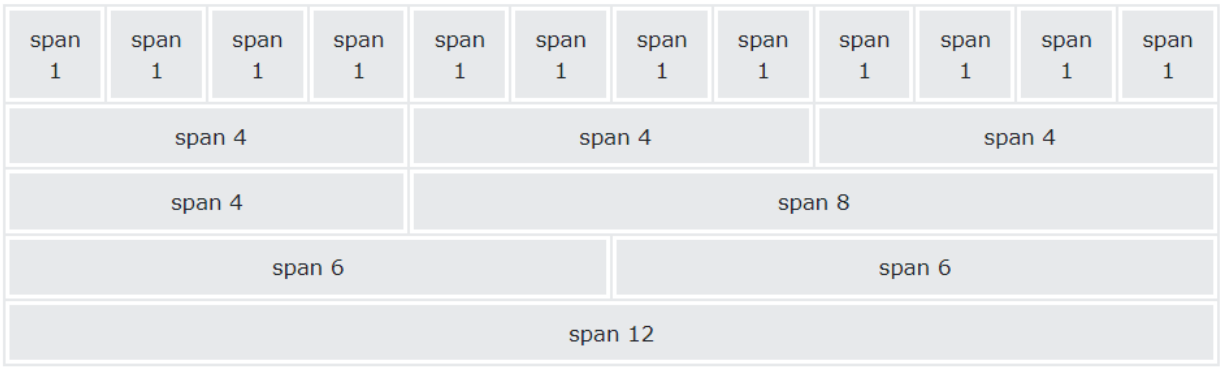
Bootstrap 5 Grid Systems

The Grid System

Bootstrap's grid system is built with flexbox and allows up to 12 columns across the page.

If you do not want to use all 12 columns individually, you can group the columns together to create wider columns:



The grid system is responsive, and the columns will re-arrange automatically depending on the screen size.

Make sure that the sum adds up to 12 or fewer (it is not required that you use all 12 available columns).

Grid Classes

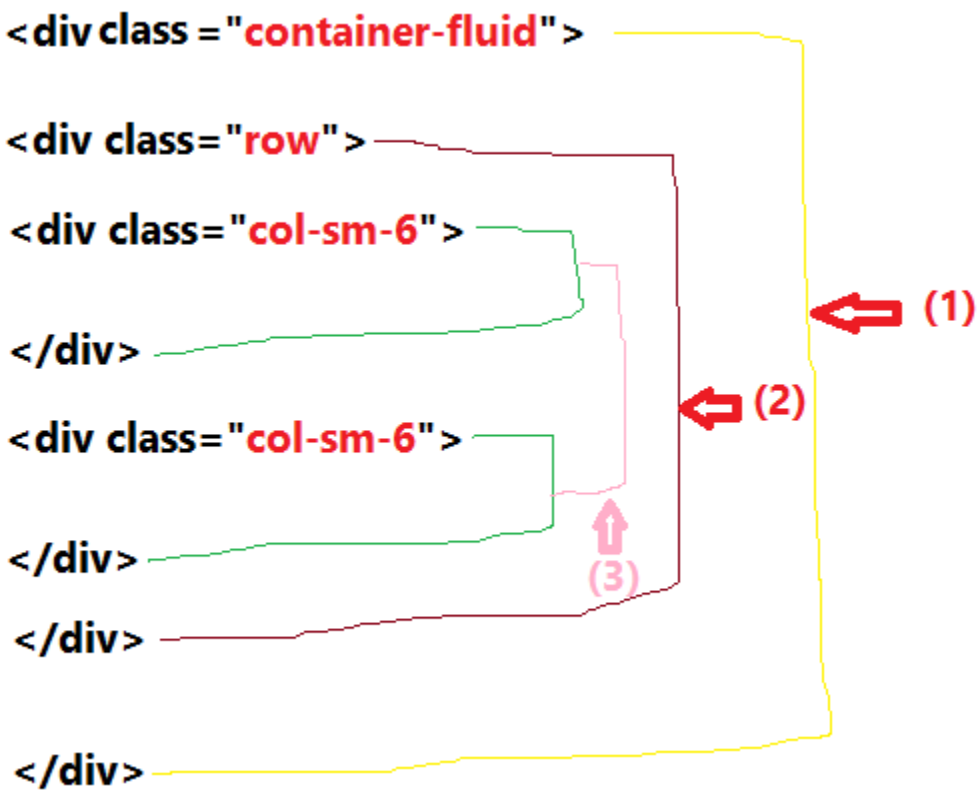
The Bootstrap 5 grid system has six classes:

- `.col-` (extra small devices - screen width less than 576px)
- `.col-sm-` (small devices - screen width equal to or greater than 576px)
- `.col-md-` (medium devices - screen width equal to or greater than 768px)
- `.col-lg-` (large devices - screen width equal to or greater than 992px)
- `.col-xl-` (xlarge devices - screen width equal to or greater than 1200px)
- `.col-xxl-` (xxlarge devices - screen width equal to or greater than 1400px)

The classes above can be combined to create more dynamic and flexible layouts.

Tip: Each class scales up, so if you want to set the same widths for `sm` and `md`, you only need to specify `sm`.

Basic Structure of a Bootstrap 5 Grid



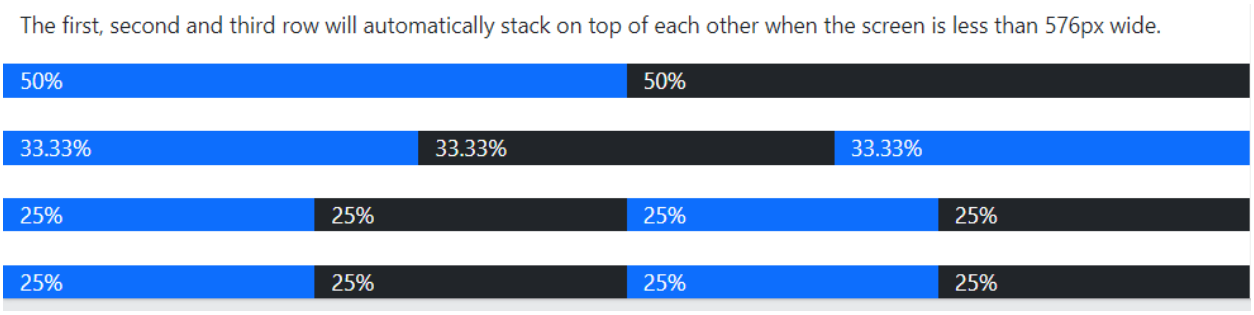
The following is a basic structure example of a Bootstrap 5 grid:-

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>
</head>
```

```
<body>

<div class="container-fluid mt-3">
  <h1>Basic Grid Structure</h1>
  <p>Resize the browser window to see the effect.</p>
  <p>The first, second and third row will automatically stack on top of each other when the screen is less than 576px wide.</p>

  <!-- Control the column width, and how they should appear on different devices -->
  <div class="row">
    <div class="col-sm-6 bg-primary text-white">50%</div>
    <div class="col-sm-6 bg-dark text-white">50%</div>
  </div>
  <br>

  <div class="row">
    <div class="col-sm-4 bg-primary text-white">33.33%</div>
    <div class="col-sm-4 bg-dark text-white">33.33%</div>
    <div class="col-sm-4 bg-primary text-white">33.33%</div>
  </div>
  <br>

  <!-- Or let Bootstrap automatically handle the layout -->
  <div class="row">
    <div class="col-sm bg-primary text-white">25%</div>
    <div class="col-sm bg-dark text-white">25%</div>
```

```
  <div class="col-sm bg-primary text-white">25%</div>
  <div class="col-sm bg-dark text-white">25%</div>
 </div>
 <br>

 <div class="row">
  <div class="col bg-primary text-white">25%</div>
  <div class="col bg-dark text-white">25%</div>
  <div class="col bg-primary text-white">25%</div>
  <div class="col bg-dark text-white">25%</div>
 </div>
</div>

</body>
</html>
```

Output:-



First example: create a row (`<div class="row">`). Then, add the desired number of columns (tags with appropriate `.col-*-*` classes). The first star (*) represents the responsiveness: sm, md, lg, xl or xxl, while the second star represents a number, which should add up to 12 for each row.

Second example: instead of adding a number to each `col`, let bootstrap handle the layout, to create equal width columns: two `"col"` elements = 50% width to each col, while three cols = 33.33% width to each col. Four cols = 25% width, etc. You can also use `.col-sm|md|lg|xl|xxl` to make the columns responsive.

Grid Options

The following table summarizes how the Bootstrap 5 grid system works across different screen sizes:

	Extra small (<576px)	Small (>=576px)	Medium (>=768px)	Large (>=992px)	Extra Large (>=1200px)	XXL (>=1400px)
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`	`.col-xxl-`
Grid behaviour	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Container width	None (auto)	540px	720px	960px	1140px	1320px
Suitable for	Portrait phones	Landscape phones	Tablets	Laptops	Laptops and Desktops	Laptops and Desktops
# of columns	12	12	12	12	12	12
Gutter width	1.5rem (.75rem on	1.5rem (.75rem on	1.5rem (.75rem on	1.5rem (.75rem on	1.5rem (.75rem on	1.5rem (.75rem on

Bootstrap 5 Grid Stacked to horizontal

Grid Example: Stacked-to-horizontal

Let's create a basic grid system that starts out stacked on extra small devices, before becoming horizontal on larger devices.

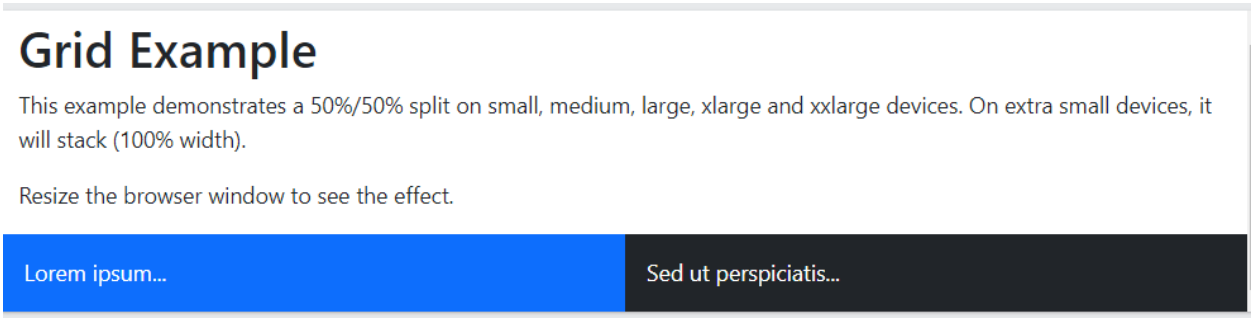
The following example shows a simple "stacked-to-horizontal" two-column layout, meaning it will result in a 50%/50% split on all screens, except for extra small screens, which it will automatically stack (100%):

Example :-

```
<!DOCTYPE html>

<html lang="en">

<head>

  <title>Bootstrap Example</title>

  <meta charset="utf-8">

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

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>



<div class="container-fluid mt-3">

  <h1>Grid Example</h1>

  <p>This example demonstrates a 50%/50% split on small, medium, large, xlarge and xxlarge devices.
On extra small devices, it will stack (100% width).</p>
```

```
 <p>Resize the browser window to see the effect.</p>

 <div class="row">

  <div class="col-sm-6 bg-primary text-white p-3">

    Lorem ipsum...

  </div>

  <div class="col-sm-6 bg-dark text-white p-3">

    Sed ut perspiciatis...

  </div>

 </div>

</div>


</body>

</html>
```

Output:-



Tip: The numbers in the `.col-sm-*` classes indicates how many columns the div should span (out of 12). So, `.col-sm-1` spans 1 column, `.col-sm-4` spans 4 columns, `.col-sm-6` spans 6 columns, etc.

Note: Make sure that the sum adds up to 12 or fewer (it is not required that you use all 12 available columns):

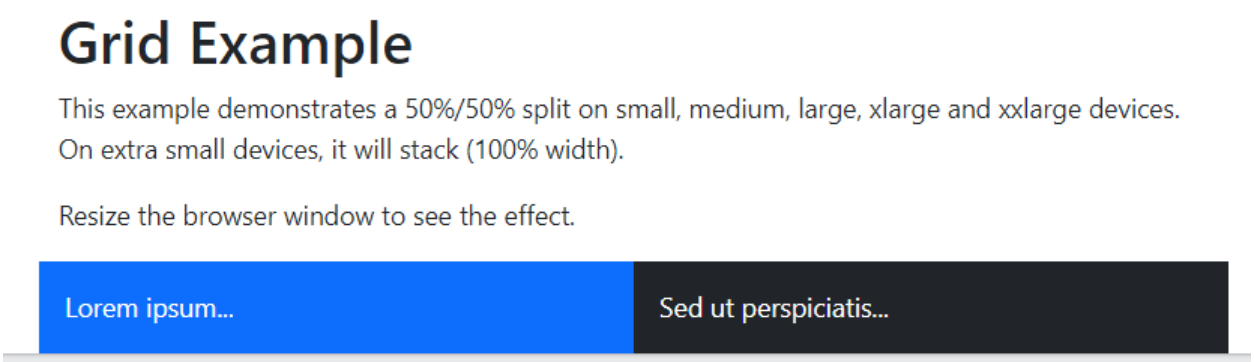
Tip: You can turn any **full-width** layout into a **fixed-width responsive** layout, by changing the `.container-fluid` class to `.container`:

Example: Responsive Container

```
<!DOCTYPE html>

<html lang="en">

<head>

  <title>Bootstrap Example</title>

  <meta charset="utf-8">

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

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>



<div class="container mt-3">

  <h1>Grid Example</h1>

  <p>This example demonstrates a 50%/50% split on small, medium, large, xlarge and xxlarge devices.
On extra small devices, it will stack (100% width).</p>

  <p>Resize the browser window to see the effect.</p>

  <div class="row">

    <div class="col-sm-6 bg-primary text-white p-3">

      Lorem ipsum...

    </div>

    <div class="col-sm-6 bg-dark text-white p-3">

      Sed ut perspiciatis...
```

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


</body>
</html>
```

Output:-



Auto Layout Columns

In Bootstrap 5, there is an easy way to create equal width columns for all devices: just remove the number from `.col-size-*` and only use the `.col-size` class on a specified number of **col elements**. Bootstrap will recognize how many columns there are, and each column will get the same width.
The *size* classes (sm, md, etc.) determines **when** the columns should be responsive:

Example :-

```
<!DOCTYPE html>

<html lang="en">

<head>

 <title>Bootstrap Example</title>

 <meta charset="utf-8">
```

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

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css"
rel="stylesheet">

  <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js"></script>

</head>

<body>


<div class="container-fluid mt-3">

  <h1>Auto Layout Columns</h1>

  <p>In Bootstrap 5, there is an easy way to create equal width columns: just use the <code>.col-
size</code> class on a specified number of col elements. Bootstrap will recognize how many columns
there are, and each column will get the same width.</p>

  <p>Two columns: 50% width on all screens, except for extra small (100% width on screens less than
<strong>576px</strong> wide)</p>

  <div class="container-fluid">

    <div class="row">

      <div class="col-sm bg-primary text-white p-3">1 of 2</div>

      <div class="col-sm bg-dark text-white p-3">2 of 2</div>

    </div>

  </div>

  <br>


  <p>Four columns: 25% width on all screens, except for extra small (100% width on screens less than
<strong>576px</strong> wide)</p>

  <div class="container-fluid">

    <div class="row">

      <div class="col-sm bg-primary text-white p-3">1 of 4</div>
```

```
    <div class="col-sm bg-dark text-white p-3">2 of 4</div>
    <div class="col-sm bg-primary text-white p-3">3 of 4</div>
    <div class="col-sm bg-dark text-white p-3">4 of 4</div>
   </div>
  </div>
</div>


</body>
</html>
```

Output:-