Bootstrap 4 Grid System

❮ Previous Next ❯

Bootstrap 4 Grid System

Bootstrap's grid system allows up to 12 columns across the page.

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

span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1	span 1
span 4			span 4			span 4					
span 4			span 8								
span 6						span 6					
span 12											

Bootstrap's grid system is responsive, and the columns will re-arrange depending on the screen size: On a big screen it might look better with the content organized in three columns, but on a small

screen it would be better if the content items were stacked on top of each other.

Grid Classes

The Bootstrap 4 grid system has five 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)

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

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

Grid System Rules

Some Bootstrap 4 grid system rules:

- Rows must be placed within a `.container` (fixed-width) or `.container-fluid` (full-width) for proper alignment and padding
- Use rows to create horizontal groups of columns
- Content should be placed within columns, and only columns may be immediate children of rows
- Predefined classes like `.row` and `.col-sm-4` are available for quickly making grid layouts
- Columns create gutters (gaps between column content) via padding. That padding is offset in rows for the first and last column via negative margin on `.rows`
- Grid columns are created by specifying the number of 12 available columns you wish to span. For example, three equal columns would use three `.col-sm-4`
- Column widths are in percentage, so they are always fluid and sized relative to their parent element
- The biggest **difference between Bootstrap 3 and Bootstrap 4** is that Bootstrap 4 now uses flexbox, instead of floats. One big advantage with flexbox is that grid columns without a

specified width will automatically layout as "equal width columns" (and equal height). Example: Three elements with `.col-sm` will each automatically be 33.33% wide from the small breakpoint and up. **Tip:** If you want to learn more about Flexbox, you can read our CSS Flexbox Tutorial.

Note that Flexbox is not supported in IE9 and earlier versions.

If you require IE8-9 support, use Bootstrap 3. It is the most stable version of Bootstrap, and it is still supported by the team for critical bugfixes and documentation changes. However, no new features will be added to it.

Basic Structure of a Bootstrap 4 Grid

The following is a basic structure of a Bootstrap 4 grid:

```
<!-- Control the column width, and how they should appear on different devices -->
<div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="row"="tagcolor" style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="col-*-*"="tagcolor" style="color:mediumblue">></div
class="tagcolor" style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="col-*-*"="tagcolor" style="color:mediumblue">></div
class="tagcolor" style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="col-*-*"="tagcolor" style="color:mediumblue">></div
class="tagcolor" style="color:mediumblue">>
</div class="tagcolor" style="color:mediumblue">>

<!-- Or let Bootstrap automatically handle the layout -->
<div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="row"="tagcolor" style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="col"="tagcolor" style="color:mediumblue">></div
class="tagcolor" style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="col"="tagcolor" style="color:mediumblue">></div
class="tagcolor" style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="col"="tagcolor" style="color:mediumblue">></div
class="tagcolor" style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="col"="tagcolor" style="color:mediumblue">></div
class="tagcolor" style="color:mediumblue">>
</div class="tagcolor" style="color:mediumblue">>
```

Try it Yourself »

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 or xl, while the second star represents a number, which should always 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. three cols = 33.33% width to each col. four cols = 25% width, etc. You can also use `.col-sm|md|lg|xl` to make the columns responsive.

Grid Options

The following table summarizes how the Bootstrap 4 grid system works across different screen sizes:

	Extra small (<576px)	**Small (>=576px)**	**Medium (>=768px)**	**Large (>=992px)**	**Extra Large (>=1200px)**
Class prefix	`.col-`	`.col-sm-`	`.col-md-`	`.col-lg-`	`.col-xl-`
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
Container width	None (auto)	540px	720px	960px	1140px
Suitable for	Portrait phones	Landscape phones	Tablets	Laptops	Laptops and Desktops
# of columns	12	12	12	12	12
Gutter width	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)

Nestable	Yes	Yes	Yes	Yes	Yes
Offsets	Yes	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes	Yes

Examples

The next chapters shows examples of grid systems for different devices and screen widths:

- Stacked-to-horizontal
- Extra Small Layout
- Small devices
- Medium devices
- Large devices
- Extra large devices
- More grid examples

❮ Previous

Log in to track progress

Next ❯