{{< callout info >}} **Heads up!** Be sure to [read the Grid page]({{< docsref "/layout/grid" >}}) first before diving into how to modify and customize your grid columns. {{< /callout >}}

How they work

- **Columns build on the grid's flexbox architecture.** Flexbox means we have options for changing individual columns and [modifying groups of columns at the row level]({{< docsref "/layout/grid#row-columns" >}}). You choose how columns grow, shrink, or otherwise change.
- **When building grid layouts, all content goes in columns.** The hierarchy of Bootstrap's grid goes from [container]({{< docsref "/layout/containers" >}}) to row to column to your content. On rare occasions, you may combine content and column, but be aware there can be unintended consequences.
- **Bootstrap includes predefined classes for creating fast, responsive layouts.** With [six breakpoints]({{< docsref "/layout/breakpoints" >}}) and a dozen columns at each grid tier, we have dozens of classes already built for you to create your desired layouts. This can be disabled via Sass if you wish.

Alignment

Use flexbox alignment utilities to vertically and horizontally align columns.

Vertical alignment

{{< example class="bd-example-row bd-example-row-flex-cols" >}}

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 >}}

{{< example class="bd-example-row bd-example-row-flex-cols" >}}

One of three columns
One of three columns
One of three columns
{{< /example >}}

Horizontal alignment

{{< example class="bd-example-row" >}}

One of two columns
One of two columns
One of two columns
One of two columns
One of two columns
One of two columns
One of two columns

One of two columns
One of two columns
One of two columns
One of two columns
One of two columns
{{< /example >}}

Column wrapping

If more than 12 columns are placed within a single row, each group of extra columns will, as one unit, wrap onto a new line.

{{< example class="bd-example-row" >}}

.col-9
.col-4
Since 9 + 4 = 13 > 12, this 4-column-wide div gets wrapped onto a new line as one contiguous unit.
.col-6
Subsequent columns continue along the new line.
{{< /example >}}

Column breaks

Breaking columns to a new line in flexbox requires a small hack: add an element with `width: 100%` wherever you want to wrap your columns to a new line. Normally this is accomplished with multiple `.row`s, but not every implementation method can account for this.

{{< example class="bd-example-row" >}}

.col-6 .col-sm-3
.col-6 .col-sm-3

```
<!-- Force next columns to break to new line -->
<div class="w-100"></div>

<div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
<div class="col-6 col-sm-3">.col-6 .col-sm-3</div>
```

{{< /example >}}
You may also apply this break at specific breakpoints with our [responsive display utilities]({{< docsref "/utilities/display" >}}).

{{< example class="bd-example-row" >}}

.col-6 .col-sm-4
.col-6 .col-sm-4

```
<!-- Force next columns to break to new line at md breakpoint and up -->
<div class="w-100 d-none d-md-block"></div>

<div class="col-6 col-sm-4">.col-6 .col-sm-4</div>
<div class="col-6 col-sm-4">.col-6 .col-sm-4</div>
```

{{< /example >}}

Reordering

Order classes

Use `.order-` classes for controlling the **visual order** of your content. These classes are responsive, so you can set the `order` by breakpoint (e.g., `.order-1.order-md-2`). Includes support for `1` through `5` across all six grid tiers.

{{< example class="bd-example-row" >}}

First in DOM, no order applied
Second in DOM, with a larger order
Third in DOM, with an order of 1
{{< /example >}}

There are also responsive `.order-first` and `.order-last` classes that change the `order` of an element by applying `order: -1` and `order: 6`, respectively. These classes can also be intermixed with the numbered `.order-*` classes as needed.

{{< example class="bd-example-row" >}}

First in DOM, ordered last
Second in DOM, unordered
Third in DOM, ordered first
{{< /example >}}

Offsetting columns

You can offset grid columns in two ways: our responsive `.offset-` grid classes and our [margin utilities]({{< docsref "/utilities/spacing" >}}). Grid classes are sized to match columns while margins are more useful for quick layouts where the width of the offset is variable.

Offset classes

Move columns to the right using `.offset-md-*` classes. These classes increase the left margin of a column by `*` columns. For example, `.offset-md-4` moves `.col-md-4` over four columns.

{{< example class="bd-example-row" >}}

.col-md-4
.col-md-4 .offset-md-4
.col-md-3 .offset-md-3
.col-md-3 .offset-md-3
.col-md-6 .offset-md-3
{{< /example >}}

In addition to column clearing at responsive breakpoints, you may need to reset offsets. See this in action in [the grid example]({{< docsref "/examples/grid" >}}).

{{< example class="bd-example-row" >}}

.col-sm-5 .col-md-6
.col-sm-5 .offset-sm-2 .col-md-6 .offset-md-0
.col-sm-6 .col-md-5 .col-lg-6
.col-sm-6 .col-md-5 .offset-md-2 .col-lg-6 .offset-lg-0
{{< /example >}}

Margin utilities

With the move to flexbox in v4, you can use margin utilities like `.me-auto` to force sibling columns away from one another.

{{< example class="bd-example-row" >}}

.col-md-4
.col-md-4 .ms-auto
.col-md-3 .ms-md-auto
.col-md-3 .ms-md-auto
.col-auto .me-auto
.col-auto
{{< /example >}}

Standalone column classes

The `.col-*` classes can also be used outside a `.row` to give an element a specific width. Whenever column classes are used as non direct children of a row, the paddings are omitted.

{{< example >}}

.col-3: width of 25%
.col-sm-9: width of 75% above sm breakpoint
{{< /example >}}

The classes can be used together with utilities to create responsive floated images. Make sure to wrap the content in a [`.clearfix`]({{< docsref "/helpers/clearfix" >}}) wrapper to clear the float if the text is shorter.

{{< example >}}

{{< placeholder width="100%" height="210" class="col-md-6 float-md-end mb-3 ms-md-3" text="Responsive floated image" >}}

A paragraph of placeholder text. We're using it here to show the use of the clearfix class. We're adding quite a few meaningless phrases here to demonstrate how the columns interact here with the floated image.

As you can see the paragraphs gracefully wrap around the floated image. Now imagine how this would look with some actual content in here, rather than just this boring placeholder text that goes on and on, but actually conveys no tangible information at. It simply takes up space and should not really be read.

And yet, here you are, still persevering in reading this placeholder text, hoping for some more insights, or some hidden easter egg of content. A joke, perhaps. Unfortunately, there's none of that here.

{{< /example >}}