

A decorative wavy line in orange and white, running vertically along the left side of the image.

BOOTSTRAP 5

Layout

([HTTPS://GETBOOTSTRAP.COM/DOCS/5.3/LAYOUT/](https://getbootstrap.com/docs/5.3/layout/))

CONTENT

1. Breakpoints
2. Containers
3. Grid system
4. Columns
5. Gutters
6. Utilities for layout

1. BREAKPOINTS

- Breakpoints are customizable widths that determine how your responsive layout behaves across device or viewport sizes in Bootstrap.

Breakpoint	Class infix	Dimensions
X-Small	<i>None</i>	<576px
Small	sm	≥576px
Medium	md	≥768px
Large	lg	≥992px
Extra large	xl	≥1200px
Extra extra large	xxl	≥1400px

MEDIA QUERIES

```
// X-Small devices (portrait phones, less than 576px)
// No media query for `xs` since this is the default in Bootstrap

// Small devices (landscape phones, 576px and up)
@media (min-width: 576px) { ... }

// Medium devices (tablets, 768px and up)
@media (min-width: 768px) { ... }

// Large devices (desktops, 992px and up)
@media (min-width: 992px) { ... }

// X-Large devices (large desktops, 1200px and up)
@media (min-width: 1200px) { ... }

// XX-Large devices (larger desktops, 1400px and up)
@media (min-width: 1400px) { ... }
```

```
// X-Small devices (portrait phones, less than 576px)
@media (max-width: 575.98px) { ... }

// Small devices (landscape phones, less than 768px)
@media (max-width: 767.98px) { ... }

// Medium devices (tablets, less than 992px)
@media (max-width: 991.98px) { ... }

// Large devices (desktops, less than 1200px)
@media (max-width: 1199.98px) { ... }

// X-Large devices (large desktops, less than 1400px)
@media (max-width: 1399.98px) { ... }

// XX-Large devices (larger desktops)
// No media query since the xxl breakpoint has no upper bound on its width
```

2. CONTAINERS

- Containers are a fundamental building block of Bootstrap that contain, pad, and align your content within a given device or viewport.

	Extra small <576px	Small ≥576px	Medium ≥768px	Large ≥992px	X-Large ≥1200px	XX-Large ≥1400px
<code>.container</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-sm</code>	100%	540px	720px	960px	1140px	1320px
<code>.container-md</code>	100%	100%	720px	960px	1140px	1320px
<code>.container-lg</code>	100%	100%	100%	960px	1140px	1320px
<code>.container-xl</code>	100%	100%	100%	100%	1140px	1320px
<code>.container-xxl</code>	100%	100%	100%	100%	100%	1320px
<code>.container-fluid</code>	100%	100%	100%	100%	100%	100%

3. GRID SYSTEM

3.1. Example

3.2. Grid options

3.3. Auto-layout columns

3.4. Responsive classes

3.5. Nesting

3.2. GRID OPTIONS

Bootstrap's grid system can adapt across all six default breakpoints, and any breakpoints you customize. The six default grid tiers are as follow:

- Extra small (**xs**)
- Small (**sm**)
- Medium (**md**)
- Large (**lg**)
- Extra large (**xl**)
- Extra extra large (**xxl**)

	xs <576px	sm ≥576px	md ≥768px	lg ≥992px	xl ≥1200px	xxl ≥1400px
Container <i>max-width</i>	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	<i>.col-</i>	<i>.col-sm-</i>	<i>.col-md-</i>	<i>.col-lg-</i>	<i>.col-xl-</i>	<i>.col-xxl-</i>
# of columns	12					
Gutter width	1.5rem (.75rem on left and right)					
Custom gutters	Yes					
Nestable	Yes					
Column ordering	Yes					

3.3. AUTO-LAYOUT COLUMNS

Utilize breakpoint-specific column classes for easy column sizing without an explicit numbered class like `.col-sm-6`.

3.3.1. Equal-width

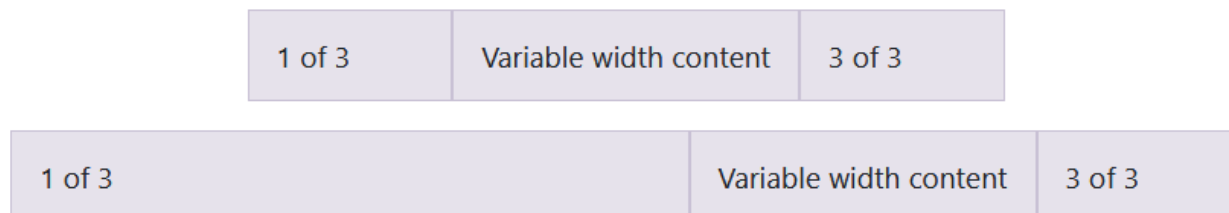
3.3.2. Setting one column width

3.3.3. Variable width content

3.3.3. VARIABLE WIDTH CONTENT

- Use `col-{breakpoint}-auto` classes to size columns based on the natural width of their content.

```
<div class="container">
  <div class="row justify-content-md-center">
    <div class="col col-lg-2">
      1 of 3
    </div>
    <div class="col-md-auto">
      Variable width content
    </div>
    <div class="col col-lg-2">
      3 of 3
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-md-auto">
      Variable width content
    </div>
    <div class="col col-lg-2">
      3 of 3
    </div>
  </div>
</div>
```



3.4. RESPONSIVE CLASSES

3.4.1. All breakpoints

3.4.2. Mix and match

3.4.3. Row columns

3.4.1. ALL BREAKPOINTS

For grids that are the same from the smallest of devices to the largest, use the `.col` and `.col-*` classes. Specify a numbered class when you need a particularly sized column; otherwise, feel free to stick to `.col`.

```
<div class="row">
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="col">col</div>
  <div class="col">col</div>
</div>
<div class="row">
  <div class="col-8">col-8</div>
  <div class="col-4">col-4</div>
</div>
```

col	col	col	col
col-8			col-4

3.4.2. MIX AND MATCH

Use a combination of different classes for each tier as needed. See the example below for a better idea of how it all works.

```
<div class="container text-center">
  <!-- Stack the columns on mobile by making one full-width and the other half-width -->
  <div class="row">
    <div class="col-md-8">.col-md-8</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns start at 50% wide on mobile and bump up to 33.3% wide on desktop -->
  <div class="row">
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>

  <!-- Columns are always 50% wide, on -->
  <div class="row">
    <div class="col-6">.col-6</div>
    <div class="col-6">.col-6</div>
  </div>
</div>
```

.col-12 .col-md-8		.col-6 .col-md-4	
.col-6 .col-md-4	.col-6 .col-md-4	.col-6 .col-md-4	
.col-6		.col-6	

3.4.3. ROW COLUMNS

- Use the responsive `.row-cols-*` classes to quickly set the number of columns that best render your content and layout

```
<div class="container">
  <div class="row row-cols-2">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

Column	Column
Column	Column

```
<div class="container">  
  <div class="row row-cols-auto">  
    <div class="col">Column</div>  
    <div class="col">Column</div>  
    <div class="col">Column</div>  
    <div class="col">Column</div>  
  </div>  
</div>
```

Column

Column

Column

Column

```
<div class="container">
  <div class="row row-cols-4">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col-6">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

Column	Column	Column
Column		

```
<div class="container">
  <div class="row row-cols-1 row-cols-sm-2 row-cols-md-4">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

Column	Column	Column	Column
--------	--------	--------	--------

3.5. NESTING

To nest your content with the default grid, add a new `.row` and set of `.col-sm-*` columns within an existing `.col-sm-*` column.

```
<div class="row">
  <div class="col-sm-3">
    Level 1: .col-sm-3
  </div>
  <div class="col-sm-9">
    <div class="row">
      <div class="col-8 col-sm-6">
        Level 2: .col-8 .col-sm-6
      </div>
      <div class="col-4 col-sm-6">
        Level 2: .col-4 .col-sm-6
      </div>
    </div>
  </div>
</div>
```

Level 1: .col-sm-3	Level 2: .col-8 .col-sm-6	Level 2: .col-4 .col-sm-6
--------------------	---------------------------	---------------------------

4. COLUMN

Learn how to modify columns with a handful of options for alignment, ordering, and offsetting thanks to our flexbox grid system

4.1. Alignment

4.2. Reordering

4.1. ALIGNMENT

Use flexbox alignment utilities to vertically and horizontally align columns.

4.1.1. Vertical alignment

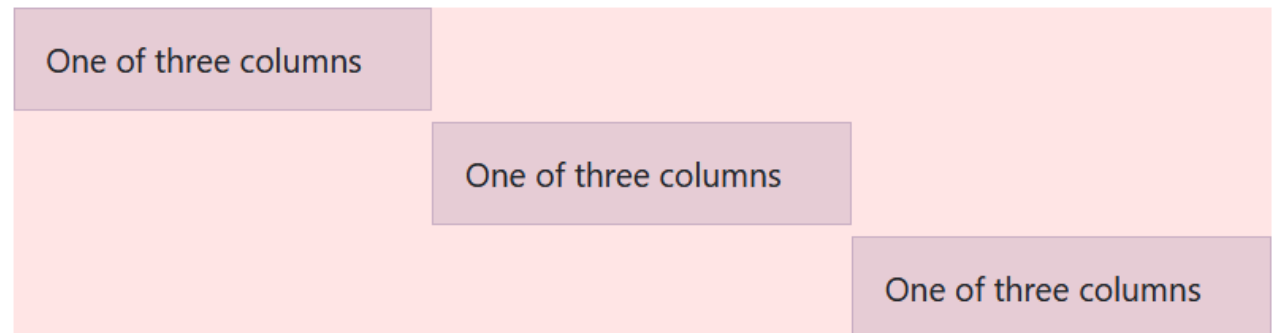
4.1.2. Horizontal alignment

4.1.1. VERTICAL ALIGNMENT

```
<div class="container">
  <div class="row align-items-start">
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
  </div>
  <div class="row align-items-center">
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
  </div>
  <div class="row align-items-end">
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
    <div class="col">
      One of three columns
    </div>
  </div>
</div>
```

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

```
<div class="container">  
  <div class="row">  
    <div class="col align-self-start">  
      One of three columns  
    </div>  
    <div class="col align-self-center">  
      One of three columns  
    </div>  
    <div class="col align-self-end">  
      One of three columns  
    </div>  
  </div>  
</div>
```




```
<div class="row justify-content-around">
  <div class="col-4">
    One of two columns
  </div>
  <div class="col-4">
    One of two columns
  </div>
</div>
<div class="row justify-content-between">
  <div class="col-4">
    One of two columns
  </div>
  <div class="col-4">
    One of two columns
  </div>
</div>
<div class="row justify-content-evenly">
  <div class="col-4">
    One of two columns
  </div>
  <div class="col-4">
    One of two columns
  </div>
</div>
```

One of two columns

One of two columns

One of two columns

One of two columns

One of two columns

One of two columns

4.2. REORDERING

4.2.1. Order classes

4.2.2. Offsetting columns

4.2.1. 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 **0** through **5** across all **six** grid tiers.

```
<div class="row">  
  <div class="col">  
    First in DOM, no order applied  
  </div>  
  <div class="col order-5">  
    Second in DOM, with a larger order  
  </div>  
  <div class="col order-1">  
    Third in DOM, with an order of 1  
  </div>  
</div>
```

First in DOM, no order applied

Third in DOM, with an order of 1

Second in DOM, with a larger order

```
<div class="container">
  <div class="row">
    <div class="col order-last">
      First in DOM, ordered last
    </div>
    <div class="col">
      Second in DOM, unordered
    </div>
    <div class="col order-first">
      Third in DOM, ordered first
    </div>
  </div>
</div>
```

Third in DOM, ordered
first

Second in DOM,
unordered

First in DOM, ordered
last

4.2.2. OFFSETTING COLUMNS

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.


```
<div class="container">
  <div class="row">
    <div class="col-md-4">.col-md-4</div>
    <div class="col-md-4 offset-md-4">.col-md-4 .offset-md-4</div>
  </div>
  <div class="row">
    <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
    <div class="col-md-3 offset-md-3">.col-md-3 .offset-md-3</div>
  </div>
  <div class="row">
    <div class="col-md-6 offset-md-3">.col-md-6 .offset-md-3</div>
  </div>
</div>
```

.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

5. GUTTERS

Gutters are the padding between your columns, used to responsively space and align content in the Bootstrap grid system.

5.1. Horizontal gutters

5.2. Vertical gutters

5.3. Horizontal & vertical gutters

5.4. No gutters

5.1. HORIZONTAL GUTTERS

`.gx-*` classes can be used to control the horizontal gutter widths. Includes support for 0 through 5 across all six grid tiers.

```
<div class="container px-4">
  <div class="row gx-5">
    <div class="col">
      <div class="p-3 border bg-light">Custom column padding</div>
    </div>
    <div class="col">
      <div class="p-3 border bg-light">Custom column padding</div>
    </div>
  </div>
</div>
```

Custom column padding

Custom column padding

5.3. HORIZONTAL & VERTICAL GUTTERS

`.g-*` classes can be used to control the horizontal gutter widths.

```
<div class="container">
  <div class="row g-2">
    <div class="col-6">
      <div class="p-3 border bg-light">Custom column padding</div>
    </div>
    <div class="col-6">
      <div class="p-3 border bg-light">Custom column padding</div>
    </div>
    <div class="col-6">
      <div class="p-3 border bg-light">Custom column padding</div>
    </div>
    <div class="col-6">
      <div class="p-3 border bg-light">Custom column padding</div>
    </div>
  </div>
</div>
```

Custom column padding

Custom column padding

Custom column padding

Custom column padding

5.4. NO GUTTERS

The gutters between columns in our predefined grid classes can be removed with `.g-0`

```
<div class="row g-0">  
  <div class="col-sm-6 col-md-8">.col-sm-6 .col-md-8</div>  
  <div class="col-6 col-md-4">.col-6 .col-md-4</div>  
</div>
```

.col-sm-6 .col-md-8

.col-6 .col-md-4

6. UTILITIES FOR LAYOUT

6.1. Changing display

6.2. Margin & padding

6.1. CHANGING DISPLAY

Use our display utilities for responsively toggling common values of the display property. Mix it with our grid system, content, or components to show or hide them across specific viewports.

6.1.1. Notation

6.1.2. Examples

6.1.3. Hiding elements

6.1.1. NOTATION

Display utility classes that apply to all breakpoints, from **xs** to **xxl**, have no breakpoint abbreviation in them.

the classes are named using the format:

.d-{value} for **xs**

.d-{breakpoint}-{value} for **sm**, **md**, **lg**, **xl** and **xxl**.

Where value is one of:

none	inline	inline-block	block
grid	table	table-cell	
table-row	flex	inline-flex	

d-inline

d-inline

```
<div class="d-inline p-2 bg-primary text-white">d-inline</div>  
<div class="d-inline p-2 bg-dark text-white">d-inline</div>
```

d-block

d-block

```
<span class="d-block p-2 bg-primary text-white">d-block</span>  
<span class="d-block p-2 bg-dark text-white">d-block</span>
```

6.1.3. HIDING ELEMENTS

- For faster mobile-friendly development, use responsive display classes for showing and hiding elements by device. Avoid creating entirely different versions of the same site, instead hide element responsively for each screen size.
- To hide elements simply use the `.d-none` class or one of the `.d-{sm,md,lg,xl,xxl}-none` classes for any responsive screen variation.
- To show an element only on a given interval of screen sizes you can combine one `.d-*-none` class with a `.d-*-*` class, for example `.d-none .d-md-block .d-xl-none .d-xxl-none` will hide the element for all screen sizes except on medium and large devices.

hide on screens smaller than lg

```
<div class="d-lg-none">hide on screens wider than lg</div>  
<div class="d-none d-lg-block">hide on screens smaller than lg</div>
```

Screen size	Class
Hidden on all	<code>.d-none</code>
Hidden only on xs	<code>.d-none .d-sm-block</code>
Hidden only on sm	<code>.d-sm-none .d-md-block</code>
Hidden only on md	<code>.d-md-none .d-lg-block</code>
Hidden only on lg	<code>.d-lg-none .d-xl-block</code>
Hidden only on xl	<code>.d-xl-none .d-xxl-block</code>
Hidden only on xxl	<code>.d-xxl-none</code>
Visible on all	<code>.d-block</code>
Visible only on xs	<code>.d-block .d-sm-none</code>
Visible only on sm	<code>.d-none .d-sm-block .d-md-none</code>
Visible only on md	<code>.d-none .d-md-block .d-lg-none</code>
Visible only on lg	<code>.d-none .d-lg-block .d-xl-none</code>
Visible only on xl	<code>.d-none .d-xl-block .d-xxl-none</code>
Visible only on xxl	<code>.d-none .d-xxl-block</code>

6.2. MARGIN & PADDING

Assign responsive-friendly margin or padding values to an element or a subset of its sides with shorthand classes.

Includes support for individual properties, all properties, and vertical and horizontal properties. Classes are built from a default Sass map ranging from **.25rem** to **3rem**.

6.2.1. Notation

6.2.2. Example

6.2.3. Horizontal centering

6.2.1. NOTATION

The classes are named using the format:

{property}{sides}-{size} for xs

{property}{sides}-{breakpoint}-{size} for sm, md, lg, xl and xxl.

- Where **property** is one of:

m - for classes that set margin

p - for classes that set padding

- Where **sides** is one of:

t - for classes that set **margin-top** or **padding-top**

b - for classes that set **margin-bottom** or **padding-bottom**

s - for classes that set **margin-left** or **padding-left** in LTR & vice versa in RTL

e - for classes that set **margin-right** or **padding-right** in LTR & vice versa in RTL

x - for classes that set both ***-left** and ***-right**

y - for classes that set both ***-top** and ***-bottom**

blank - for classes that set a margin or padding on all **4 sides** of the element

Where **size** is one of:

- 0** - for classes that eliminate the **margin** or **padding** by setting it to **0**
- 1** - (by default) for classes that set the **margin** or **padding** to $\$spacer * .25$
- 2** - (by default) for classes that set the **margin** or **padding** to $\$spacer * .5$
- 3** - (by default) for classes that set the **margin** or **padding** to $\$spacer$
- 4** - (by default) for classes that set the **margin** or **padding** to $\$spacer * 1.5$
- 5** - (by default) for classes that set the **margin** or **padding** to $\$spacer * 3$
- auto** - for classes that set the **margin** to **auto**

Block with padding & margin bottom

Block with margin left & padding left-right

6.2.3. HORIZONTAL CENTERING

Additionally, Bootstrap also includes an `.mx-auto` class for horizontally centering fixed-width block level content—that is, content that has `display: block` and a width set—by setting the horizontal margins to auto.

Centered element

```
<div class="mx-auto" style="width: 200px;">  
  Centered element  
</div>
```



THE END