

# BOOTSTRAP 4

LAYOUT

([HTTPS://GETBOOTSTRAP.COM/DOCS/4.0/LAYOUT/OVERVIEW](https://getbootstrap.com/docs/4.0/layout/overview))

# CONTENT

1. Overview
2. Grid system
3. Media object
4. Utilities for layout

# 1. OVERVIEW

1.1. Containers

1.2. Responsive breakpoints

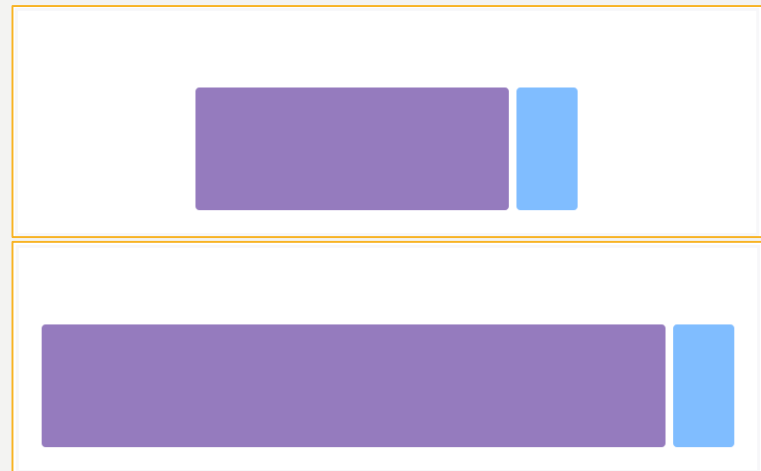
1.3. Z-index

# 1.1. CONTAINERS

- Containers are the most basic layout element in Bootstrap and are **required when using our default grid system**. Choose from a responsive, **fixed-width** container (meaning its **max-width** changes at each breakpoint) or **fluid-width** (meaning it's 100% wide all the time).

- Fixed-width

```
<div class="container">  
  <!-- Content here -->  
</div>  
<div class="container-fluid">  
  <!-- Content here -->  
</div>
```



# 1.2. RESPONSIVE BREAKPOINTS

```
// Extra small devices (portrait phones, less than 576px)  
// No media query 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) { ... }
```

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

```
// Extra small devices (portrait phones, less than 576px)
```

```
@media (max-width: 575px) { ... }
```

```
// Small devices (landscape phones, less than 768px)
```

```
@media (max-width: 767px) { ... }
```

```
// Medium devices (tablets, less than 992px)
```

```
@media (max-width: 991px) { ... }
```

```
// Large devices (desktops, less than 1200px)
```

```
@media (max-width: 1199px) { ... }
```

```
// Extra large devices (large desktops)
```

```
// No media query since the extra-large breakpoint has no upper bound on its width
```

```
// Extra small devices (portrait phones, less than 576px)
@media (max-width: 575px) { ... }

// Small devices (landscape phones, 576px and up)
@media (min-width: 576px) and (max-width: 767px) { ... }

// Medium devices (tablets, 768px and up)
@media (min-width: 768px) and (max-width: 991px) { ... }

// Large devices (desktops, 992px and up)
@media (min-width: 992px) and (max-width: 1199px) { ... }

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

# 1.3. Z-INDEX

```
$zindex-dropdown-backdrop: 990 !default;  
$zindex-dropdown: 1000 !default;  
$zindex-fixed: 1030 !default;  
$zindex-sticky: 1030 !default;  
$zindex-modal-backdrop: 1040 !default;  
$zindex-modal: 1050 !default;  
$zindex-popover: 1060 !default;  
$zindex-tooltip: 1070 !default;
```



# 2. GRID SYSTEM

2.1. How it works

2.2. Grid options

2.3. Auto-layout columns

2.4. Responsive classes

2.5. Alignment

2.6. Reordering

2.7. Nesting

# 2.1. HOW IT WORKS

- Bootstrap's grid system uses a series of containers, rows, and columns to layout and align content. It's built with flexbox and is fully responsive.
- Breaking it down
  - Containers provide a means to center and horizontally pad your site's contents. Use `.container` for a responsive pixel width or `.container-fluid` for width: `100%` across all viewport and device sizes.
  - Rows are wrappers for columns. Each column has horizontal `padding` (called a gutter) for controlling the space between them. This `padding` is then counteracted on the rows with negative margins. This way, all the content in your columns is visually aligned down the left side.
  - In a grid layout, content must be placed within columns and only columns may be immediate children of rows.
  - Thanks to flexbox, grid columns without a specified width will automatically layout as equal width columns. For example, four instances of `.col-sm` will each automatically be `25%` wide from the small breakpoint and up.

- Column classes indicate the number of columns you'd like to use out of the possible 12 per row. So, if you want three equal-width columns across, you can use `.col-4`.
- Column widths are set in percentages, so they're always fluid and sized relative to their parent element.
- Columns have horizontal padding to create the gutters between individual columns, however, you can remove the margin from rows and padding from columns with `.no-gutters` on the `.row`.
- To make the grid responsive, there are five grid breakpoints, one for each responsive breakpoint: all breakpoints (extra small), small, medium, large, and extra large.
- Grid breakpoints are based on minimum width media queries, meaning they apply to that one breakpoint and all those above it (e.g., `.col-sm-4` applies to small, medium, large, and extra large devices, but not the first `xs` breakpoint).
- You can use predefined grid classes (like `.col-4`).

## 2.2. GRID OPTIONS

	<b>Extra small</b> <576px	<b>Small</b> ≥576px	<b>Medium</b> ≥768px	<b>Large</b> ≥992px	<b>Extra large</b> ≥1200px
<b>Max container width</b>	None (auto)	540px	720px	960px	1140px
<b>Class prefix</b>	<code>.col-</code>	<code>.col-sm-</code>	<code>.col-md-</code>	<code>.col-lg-</code>	<code>.col-xl-</code>
<b># of columns</b>	12				
<b>Gutter width</b>	30px (15px on each side of a column)				
<b>Nestable</b>	Yes				
<b>Column ordering</b>	Yes				

## 2.3. AUTO-LAYOUT COLUMNS

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

2.3.1. Equal-width

2.3.2. Setting one column width

2.3.3. Variable width content

2.3.4. Equal-width multi-row

## 2.3.1. EQUAL-WIDTH

```
<div class="container">
  <div class="row">
    <div class="col">
      1 of 2
    </div>
    <div class="col">
      2 of 2
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col">
      2 of 3
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
```

1 of 2

2 of 2

1 of 3

2 of 3

3 of 3

## 2.3.2. SETTING ONE COLUMN WIDTH

```
<div class="container">
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-6">
      2 of 3 (wider)
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
  <div class="row">
    <div class="col">
      1 of 3
    </div>
    <div class="col-5">
      2 of 3 (wider)
    </div>
    <div class="col">
      3 of 3
    </div>
  </div>
</div>
```

1 of 3	2 of 3 (wider)	3 of 3
1 of 3	2 of 3 (wider)	3 of 3

## 2.3.3. 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>
```

1 of 3	Variable width content	3 of 3
1 of 3	Variable width content	3 of 3



## 2.3.4. EQUAL-WIDTH MULTI-ROW

- Create equal-width columns that span multiple rows by inserting a `.w-100` where you want the columns to break to a new line. Make the breaks responsive by mixing the `.w-100` with some responsive display utilities.

```
<div class="container">
  <div class="row">
    <div class="col">Column</div>
    <div class="col">Column</div>
    <div class="w-100"></div>
    <div class="col">Column</div>
    <div class="col">Column</div>
  </div>
</div>
```

Column	Column
Column	Column

## **2.4. RESPONSIVE CLASSES**

2.4.1. All breakpoints

2.4.2. Stacked to horizontal

2.4.3. Mix and match

## 2.4.1. 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

## 2.4.2. STACKED TO HORIZONTAL

Using a single set of `.col-sm-*` classes, you can create a basic grid system that starts out stacked on extra small devices before becoming horizontal on desktop (medium) devices.

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

col-sm-8		col-sm-4
col-sm	col-sm	col-sm

## 2.4.3. 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.

```
<!-- Stack the columns on mobile by making one full-width and the other half-width  
<div class="row">  
  <div class="col-12 col-md-8">.col-12 .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 mobile and desktop -->  
<div class="row">  
  <div class="col-6">.col-6</div>  
  <div class="col-6">.col-6</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	

# 2.5. ALIGNMENT

Use flexbox alignment utilities to vertically and horizontally align columns.

2.5.1. Vertical alignment

2.5.2. Horizontal alignment

2.5.3. No gutters

2.5.4. Column wrapping

2.5.5. Column breaks

# 2.5.1. 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>
```

One of three columns

One of three columns

One of three columns



## 2.5.2. HORIZONTAL ALIGNMENT

```
<div class="container">
  <div class="row justify-content-start">
    <div class="col-4">
      One of two columns
    </div>
    <div class="col-4">
      One of two columns
    </div>
  </div>
  <div class="row justify-content-center">
    <div class="col-4">
      One of two columns
    </div>
    <div class="col-4">
      One of two columns
    </div>
  </div>
  <div class="row justify-content-end">
    <div class="col-4">
      One of two columns
    </div>
    <div class="col-4">
      One of two columns
    </div>
  </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

```
<div class="container">
  <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>
```

One of two columns

One of two columns

One of two columns

One of two columns

`.col-12 .col-sm-6 .col-md-8`

`.col-6 .col-md-4`





# 2.6. REORDERING

2.6.1. Order classes

2.6.2. Offsetting columns

## 2.6.1. 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 1 through 12 across all five grid tiers.

```
<div class="container">
  <div class="row">
    <div class="col">
      First, but unordered
    </div>
    <div class="col order-12">
      Second, but last
    </div>
    <div class="col order-1">
      Third, but first
    </div>
  </div>
</div>
```

First, but unordered

Third, but first

Second, but last

# **2.6.2. OFFSETTING COLUMNS**

2.6.2.1. Offset classes

2.6.2.2. Margin utilities



## 2.6.2.1. 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.

`.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`

```
<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>
```

## 2.6.2.2. MARGIN UTILITIES

Use margin utilities like `.mr-auto` to force sibling columns away from one another.

```
<div class="row">
  <div class="col-md-4">.col-md-4</div>
  <div class="col-md-4 ml-auto">.col-md-4 .ml-auto</div>
</div>
<div class="row">
  <div class="col-md-3 ml-md-auto">.col-md-3 .ml-md-auto</div>
  <div class="col-md-3 ml-md-auto">.col-md-3 .ml-md-auto</div>
</div>
<div class="row">
  <div class="col-auto mr-auto">.col-auto .mr-auto</div>
  <div class="col-auto">.col-auto</div>
</div>
```

.col-md-4

.col-md-4 .ml-auto

.col-md-3 .ml-md-  
auto

.col-md-3 .ml-md-  
auto

.col-auto .mr-auto

.col-auto

## 2.7. 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. Nested rows should include a set of columns that add up to 12 or fewer

```
<div class="row">
  <div class="col-sm-9">
    Level 1: .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-9

Level 2: .col-8 .col-sm-6

Level 2: .col-4 .col-sm-6

# 3. MEDIA OBJECT

3.1. Example

3.2. Nesting

3.3. Alignment

3.4. Order

3.5. Media list

# 3.1. EXAMPLE

The media object helps build complex and repetitive components where some media is positioned alongside content that doesn't wrap around said media. It requires 2 classes: `.media`, `.media-body`



## Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

```
<div class="media">
  
  <div class="media-body">
    <h5 class="mt-0">Media heading</h5>
    Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante
    sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.
    Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in
    faucibus.
  </div>
</div>
```

## 3.2. NESTING

Media objects can be infinitely nested, though we suggest you stop at some point. Place nested `.media` within the `.media-body` of a parent media object.

```
<div class="media">
  
  <div class="media-body">
    <h5 class="mt-0">Media heading</h5>
    Cras sit amet nibh libero, in gravida nulla...

    <div class="media mt-3">
      <a class="d-flex pr-3" href="#">
        
      </a>
      <div class="media-body">
        <h5 class="mt-0">Media heading</h5>
        Cras sit amet nibh libero, in gravida nulla...
      </div>
    </div>
  </div>
</div>
```

64x64

### Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

64x64

### Media heading

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

## 3.3. ALIGNMENT

Media in a media object can be aligned with flexbox utilities to the top (default), middle, or end of your .media-body content.

64x64

### Top-aligned media

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

```
<div class="media">
  
  <div class="media-body">
    <h5 class="mt-0">Top-aligned media</h5>
    <p>Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollic:
    <p>Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis
  </div>
</div>
```

Copy

64x64

## Center-aligned media

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Copy

```
<div class="media">
  
  <div class="media-body">
    <h5 class="mt-0">Center-aligned media</h5>
    <p>Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.
    <p class="mb-0">Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo.
  </div>
</div>
```

<

>



## Bottom-aligned media

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

64x64

Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Copy

```
<div class="media">
  
  <div class="media-body">
    <h5 class="mt-0">Bottom-aligned media</h5>
    <p>Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollic:
    <p class="mb-0">Donec sed odio dui. Nullam quis risus eget urna mollis ornare vel eu leo
  </div>
</div>
```

<

>

## 3.4. ORDER

Change the order of content in media objects by modifying the HTML itself, or by adding some custom flexbox CSS to set the order property (to an integer of your choosing).

### Media object

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

64x64

```
<div class="media">
  <div class="media-body">
    <h5 class="mt-0 mb-1">Media object</h5>
    Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante
    sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis.
    Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in
    faucibus.
  </div>
  
</div>
```

## 3.5. MEDIA LIST

Because the media object has so few structural requirements, you can also use these classes on list HTML elements.

On your `<ul>` or `<ol>`, add the `.list-unstyled` to remove any browser default list styles, and then apply `.media` to your `<li>`s.

As always, use spacing utilities wherever needed to fine tune.

```
<ul class="list-unstyled">
  <li class="media">
    
    <div class="media-body">
      <h5 class="mt-0 mb-1">List-based media object</h5>
      Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin.
    </div>
  </li>
  <li class="media my-4">
    
    <div class="media-body">
      <h5 class="mt-0 mb-1">List-based media object</h5>
      Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin.
    </div>
  </li>
  <li class="media">
    
    <div class="media-body">
      <h5 class="mt-0 mb-1">List-based media object</h5>
      Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin.
    </div>
  </li>
</ul>
```



64x64

### List-based media object

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

64x64

### List-based media object

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

64x64

### List-based media object

Cras sit amet nibh libero, in gravida nulla. Nulla vel metus scelerisque ante sollicitudin. Cras purus odio, vestibulum in vulputate at, tempus viverra turpis. Fusce condimentum nunc ac nisi vulputate fringilla. Donec lacinia congue felis in faucibus.

# 4. UTILITIES FOR LAYOUT

4.1. Changing display

4.2. Margin and padding

4.3. Toggle visibility

# 4.1. 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.

4.1.1. How it works

4.1.2. Notation

4.1.3. Examples

4.1.4. Hiding elements

4.1.5. Display in print

## 4.1.1. HOW IT WORKS

Change the value of the display property with our responsive display utility classes. We purposely support only a subset of all possible values for display. Classes can be combined for various effects as you need.



## 4.1.2. NOTATION

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

the classes are named using the format:

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

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

Where value is one of:

`none`

`inline`

`inline-block`

`block`

`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>
```

## 4.1.4. 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}-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` 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</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</code>

# 4.1.5. DISPLAY IN PRINT

- Change the display value of elements when printing with our print display utility classes. Includes support for the same display values as our responsive `.d-*` utilities.

`.d-print-none`

`.d-print-inline`

`.d-print-inline-block`

`.d-print-block`

`.d-print-table`

`.d-print-table-row`

`.d-print-table-cell`

`.d-print-flex`

`.d-print-inline-flex`

Screen Only (Hide on print only)

Hide up to large on screen, but always show on print

```
<div class="d-print-none">Screen Only (Hide on print only)</div>
<div class="d-none d-print-block">Print Only (Hide on screen only)</div>
<div class="d-none d-lg-block d-print-block">Hide up to large on screen,
but always show on print</div>
```

## 4.2. MARGIN AND PADDING

Use the margin and padding spacing utilities to control how elements and components are spaced and sized. Bootstrap 4 includes a five-level scale for spacing utilities, based on a 1rem value default \$spacer variable. Choose values for all viewports (e.g., `.mr-3` for `margin-right: 1rem`), or pick responsive variants to target specific viewports (e.g., `.mr-md-3` for `margin-right: 1rem` starting at the md breakpoint).

4.2.1. How it works

4.2.2. Notation

4.2.3. Horizontal centering

## 4.2.1. HOW IT WORKS

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`.

## 4.2.2. NOTATION

The classes are named using the format:

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

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

- 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**

**l** - for classes that set **margin-left** or **padding-left**

**r** - for classes that set **margin-right** or **padding-right**

**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**

## 4.2.3. 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>
```

## 4.3. TOGGLE VISIBILITY

When toggling display isn't needed, you can toggle the visibility of an element with our visibility utilities. Invisible elements will still affect the layout of the page, but are visually hidden from visitors.

`.visible`

`.invisible`