

Bootstrap 4 Utilities

[< Previous](#)[Next >](#)

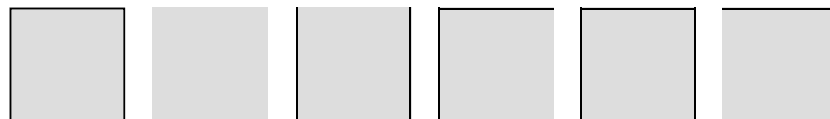
Bootstrap 4 Utilities

Bootstrap 4 has a lot of utility/helper classes to quickly style elements without using any CSS code.

Borders

Use the `border` classes to add or remove borders from an element:

Example



Example

```

<span class="attribute" style="color:red" > class="attributevalue"
style="color:mediumblue" >="border"="tagcolor" style="color:mediumblue" >></span
class="tagcolor" style="color:mediumblue" >>
<span class="attribute" style="color:red" > class="attributevalue"
style="color:mediumblue" >="border border-0"="tagcolor" style="color:mediumblue" >>
</span class="tagcolor" style="color:mediumblue" >>
<span class="attribute" style="color:red" > class="attributevalue"
style="color:mediumblue" >="border border-top-0"="tagcolor"
style="color:mediumblue" >></span class="tagcolor" style="color:mediumblue" >>
<span class="attribute" style="color:red" > class="attributevalue"
style="color:mediumblue" >="border border-right-0"="tagcolor"
style="color:mediumblue" >></span class="tagcolor" style="color:mediumblue" >>
<span class="attribute" style="color:red" > class="attributevalue"
style="color:mediumblue" >="border border-bottom-0"="tagcolor"
style="color:mediumblue" >></span class="tagcolor" style="color:mediumblue" >>
<span class="attribute" style="color:red" > class="attributevalue"
style="color:mediumblue" >="border border-left-0"="tagcolor"
style="color:mediumblue" >></span class="tagcolor" style="color:mediumblue" >>

```

Try it Yourself »

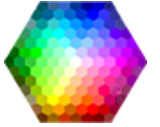
Border Color

Add a color to the border with any of the contextual border color classes:

Example



COLOR
PICKER



 Get Certified

ADVERTISEMENT

Example

```
<span class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="border border-primary"="tagcolor"
style="color:mediumblue">></span class="tagcolor" style="color:mediumblue">>
<span class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="border border-secondary"="tagcolor"
style="color:mediumblue">></span class="tagcolor" style="color:mediumblue">>
<span class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="border border-success"="tagcolor"
style="color:mediumblue">></span class="tagcolor" style="color:mediumblue">>
<span class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="border border-danger"="tagcolor"
style="color:mediumblue">></span class="tagcolor" style="color:mediumblue">>
<span class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="border border-warning"="tagcolor"
style="color:mediumblue">></span class="tagcolor" style="color:mediumblue">>
<span class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="border border-info"="tagcolor"
style="color:mediumblue">></span class="tagcolor" style="color:mediumblue">>
<span class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="border border-light"="tagcolor"
style="color:mediumblue">></span class="tagcolor" style="color:mediumblue">>
<span class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="border border-dark"="tagcolor"
style="color:mediumblue">></span class="tagcolor" style="color:mediumblue">>
<span class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="border border-white"="tagcolor"
style="color:mediumblue">></span class="tagcolor" style="color:mediumblue">>
```

Try it Yourself »

Border Radius

Add rounded corners to an element with the `rounded` classes:

Example



Example

```
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="rounded-sm"="tag" style="color:mediumblue">></span
class="tag" style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="rounded"="tag" style="color:mediumblue">></span
class="tag" style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="rounded-lg"="tag" style="color:mediumblue">></span
class="tag" style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="rounded-top"="tag" style="color:mediumblue">>
</span class="tag" style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="rounded-right"="tag" style="color:mediumblue">>
</span class="tag" style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="rounded-bottom"="tag" style="color:mediumblue">>
</span class="tag" style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="rounded-left"="tag" style="color:mediumblue">>
</span class="tag" style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="rounded-circle"="tag" style="color:mediumblue">>
</span class="tag" style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
```

```
style="color:mediumblue">="rounded-0"="tagcolor" style="color:mediumblue">></span
class="tagcolor" style="color:mediumblue">>
```

Try it Yourself »

ADVERTISEMENT

Float and Clearfix

Float an element to the right with the `.float-right` class or to the left with `.float-left`, and clear floats with the `.clearfix` class:

Example

Float left

Float right

Example

```
<div class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="clearfix"="tag" style="color:mediumblue">>
  <span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="float-left"="tag" style="color:mediumblue">>Float
left</span class="tag" style="color:mediumblue">>
  <span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="float-right"="tag" style="color:mediumblue">>Float
right</span class="tag" style="color:mediumblue">>
</div class="tag" style="color:mediumblue">>
```

Try it Yourself »

Responsive Floats

Float an element to the left or to the right depending on screen width, with the responsive float classes (`.float-*-left|right` - where * is `sm` ($\geq 576\text{px}$), `md` ($\geq 768\text{px}$), `lg` ($\geq 992\text{px}$) or `xl` ($\geq 1200\text{px}$)):

Example

Float none

Float right on small screens or wider
 Float right on medium screens or wider
 Float right on large screens or wider
 Float right on extra large screens or wider

Example

```
<div class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="float-sm-right"="tag"
style="color:mediumblue">>Float right on small screens or wider</div
class="tag" style="color:mediumblue">><br class="tag"
```

```

style="color:mediumblue">>
<div class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="float-md-right"="tagcolor"
style="color:mediumblue">>Float right on medium screens or wider</div
class="tagcolor" style="color:mediumblue">><br class="tagcolor"
style="color:mediumblue">>
<div class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="float-lg-right"="tagcolor"
style="color:mediumblue">>Float right on large screens or wider</div
class="tagcolor" style="color:mediumblue">><br class="tagcolor"
style="color:mediumblue">>
<div class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="float-xl-right"="tagcolor"
style="color:mediumblue">>Float right on extra large screens or wider</div
class="tagcolor" style="color:mediumblue">><br class="tagcolor"
style="color:mediumblue">>
<div class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="float-none"="tagcolor" style="color:mediumblue">>Float
none</div class="tagcolor" style="color:mediumblue">>

```

Try it Yourself »

Center Align

Center an element with the `.mx-auto` class (adds margin-left and margin-right: auto):

Example

Centered

Example

```
<div class="attribute" style="color:red" class="attributevalue" style="color:mediumblue">="mx-auto bg-warning" style="attributevalue" style="color:mediumblue">="width:150px"="tagcolor" style="color:mediumblue">>Centered</div class="tagcolor" style="color:mediumblue">>
```

Try it Yourself »

Width

Set the width of an element with the w-* classes (`.w-25` , `.w-50` , `.w-75` , `.w-100` , `.mw-100`):

Example

Width 25%

Width 50%

Width 75%

Width 100%

Max Width 100%

Example

```
<div class="attribute" style="color:red" class="attributevalue" style="color:mediumblue">="w-25 bg-warning"="tagcolor" style="color:mediumblue">>Width 25%</div class="tagcolor" style="color:mediumblue">>  
<div class="attribute" style="color:red" class="attributevalue" style="color:mediumblue">="w-50 bg-warning"="tagcolor"
```



```

style="color:mediumblue">>Width 50%</div class="tagcolor"
style="color:mediumblue">>
<div class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="w-75 bg-warning"="tagcolor"
style="color:mediumblue">>Width 75%</div class="tagcolor"
style="color:mediumblue">>
<div class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="w-100 bg-warning"="tagcolor"
style="color:mediumblue">>Width 100%</div class="tagcolor"
style="color:mediumblue">>
<div class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="mw-100 bg-warning"="tagcolor"
style="color:mediumblue">>Max Width 100%</div class="tagcolor"
style="color:mediumblue">>

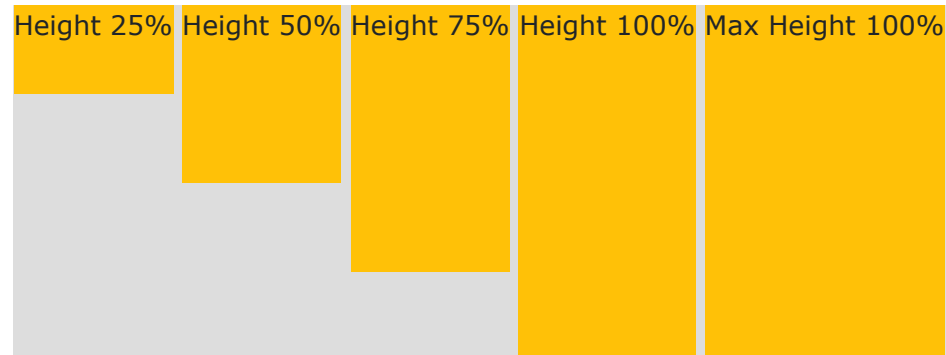
```

Try it Yourself »

Height

Set the height of an element with the h-* classes (`.h-25` , `.h-50` , `.h-75` , `.h-100` , `.mh-100`):

Example



Example

```
<div class="attributecolor" style="color:red"> style="attributevaluecolor"
style="color:mediumblue">="height:200px;background-color:#ddd"="tagcolor"
style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="h-25 bg-warning"="tagcolor"
style="color:mediumblue">>Height 25%</div class="tagcolor"
style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="h-50 bg-warning"="tagcolor"
style="color:mediumblue">>Height 50%</div class="tagcolor"
style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="h-75 bg-warning"="tagcolor"
style="color:mediumblue">>Height 75%</div class="tagcolor"
style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="h-100 bg-warning"="tagcolor"
style="color:mediumblue">>Height 100%</div class="tagcolor"
style="color:mediumblue">>
  <div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="mh-100 bg-warning" style="attributevaluecolor"
style="color:mediumblue">="height:500px"="tagcolor" style="color:mediumblue">>Max
Height 100%</div class="tagcolor" style="color:mediumblue">>
</div class="tagcolor" style="color:mediumblue">>
```

Try it Yourself »

Spacing

Bootstrap 4 has a wide range of responsive margin and padding utility classes. They work for all breakpoints: `xs` ($\leq 576\text{px}$), `sm` ($\geq 576\text{px}$), `md` ($\geq 768\text{px}$), `lg` ($\geq 992\text{px}$) or `xl`

(>=1200px)):

The classes are used in the format: `{property}{sides}-{size}` for `xs` and `{property}{sides}-{breakpoint}-{size}` for `sm`, `md`, `lg`, and `xl`.

Where *property* is one of:

- `m` - sets `margin`
- `p` - sets `padding`

Where *sides* is one of:

- `t` - sets `margin-top` or `padding-top`
- `b` - sets `margin-bottom` or `padding-bottom`
- `l` - sets `margin-left` or `padding-left`
- `r` - sets `margin-right` or `padding-right`
- `x` - sets both `padding-left` and `padding-right` or `margin-left` and `margin-right`
- `y` - sets both `padding-top` and `padding-bottom` or `margin-top` and `margin-bottom`
- blank - sets a `margin` or `padding` on all 4 sides of the element

Where *size* is one of:

- `0` - sets `margin` or `padding` to `0`
- `1` - sets `margin` or `padding` to `.25rem` (4px if font-size is 16px)
- `2` - sets `margin` or `padding` to `.5rem` (8px if font-size is 16px)
- `3` - sets `margin` or `padding` to `1rem` (16px if font-size is 16px)
- `4` - sets `margin` or `padding` to `1.5rem` (24px if font-size is 16px)
- `5` - sets `margin` or `padding` to `3rem` (48px if font-size is 16px)
- `auto` - sets `margin` to auto

Note: margins can also be negative, by adding an "n" in front of *size*:

- `n1` - sets `margin` to `-.25rem` (-4px if font-size is 16px)
- `n2` - sets `margin` to `-.5rem` (-8px if font-size is 16px)
- `n3` - sets `margin` to `-1rem` (-16px if font-size is 16px)
- `n4` - sets `margin` to `-1.5rem` (-24px if font-size is 16px)
- `n5` - sets `margin` to `-3rem` (-48px if font-size is 16px)

Example

I only have a top padding (1.5rem = 24px)

I have a padding on all sides (3rem = 48px)

I have a margin on all sides (3rem = 48px) and a bottom padding (3rem = 48px)

Example

```
<div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">= "pt-4 bg-warning"="tagcolor" style="color:mediumblue">>I
only have a top padding (1.5rem = 24px)</div class="tagcolor"
style="color:mediumblue">>
<div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">= "p-5 bg-success"="tagcolor" style="color:mediumblue">>I
have a padding on all sides (3rem = 48px)</div class="tagcolor"
style="color:mediumblue">>
<div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">= "m-5 pb-5 bg-info"="tagcolor"
style="color:mediumblue">>I have a margin on all sides (3rem = 48px) and a bottom
padding (3rem = 48px)</div class="tagcolor" style="color:mediumblue">>
```

Try it Yourself »

More Spacing Examples

<code>.m-# / m-*-#</code>	margin on all sides	Try it
<code>.mt-# / mt-*-#</code>	margin top	Try it
<code>.mb-# / mb-*-#</code>	margin bottom	Try it
<code>.ml-# / ml-*-#</code>	margin left	Try it
<code>.mr-# / mr-*-#</code>	margin right	Try it
<code>.mx-# / mx-*-#</code>	margin left and right	Try it
<code>.my-# / my-*-#</code>	margin top and bottom	Try it
<code>.p-# / p-*-#</code>	padding on all sides	Try it
<code>.pt-# / pt-*-#</code>	padding top	Try it
<code>.pb-# / pb-*-#</code>	padding bottom	Try it
<code>.pl-# / pl-*-#</code>	padding left	Try it
<code>.pr-# / pr-*-#</code>	padding right	Try it
<code>.py-# / py-*-#</code>	padding top and bottom	Try it
<code>.px-# / px-*-#</code>	padding left and right	Try it

Shadows

Use the `shadow-` classes to add shadows to an element:

Example

No shadow

Small shadow

Default shadow

Large shadow

Example

```
<div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">"shadow-none p-4 mb-4 bg-light"="tagcolor"
style="color:mediumblue">>No shadow</div class="tagcolor"
style="color:mediumblue">>
<div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">"shadow-sm p-4 mb-4 bg-white"="tagcolor"
style="color:mediumblue">>Small shadow</div class="tagcolor"
style="color:mediumblue">>
<div class="attributecolor" style="color:red"> class="attributevaluecolor"
```

```

style="color:mediumblue">="shadow p-4 mb-4 bg-white"="tagcolor"
style="color:mediumblue">>Default shadow</div class="tagcolor"
style="color:mediumblue">>
<div class="attributeicolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="shadow-lg p-4 mb-4 bg-white"="tagcolor"
style="color:mediumblue">>Large shadow</div class="tagcolor"
style="color:mediumblue">>

```

Try it Yourself »

Vertical Align

Use the `align-` classes to change the alignment of elements (only works on inline, inline-block, inline-table and table cell elements):

Example

baseline top middle bottom text-top text-bottom

Example

```

<span class="attributeicolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="align-baseline"="tagcolor"
style="color:mediumblue">>baseline</span class="tagcolor"
style="color:mediumblue">>
<span class="attributeicolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="align-top"="tagcolor"
style="color:mediumblue">>top</span class="tagcolor" style="color:mediumblue">>
<span class="attributeicolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="align-middle"="tagcolor"
style="color:mediumblue">>middle</span class="tagcolor" style="color:mediumblue">>
<span class="attributeicolor" style="color:red"> class="attributevaluecolor"

```

```

style="color:mediumblue">="align-bottom"="tagcolor"
style="color:mediumblue">>bottom</span class="tagcolor" style="color:mediumblue">>
<span class="attributevaluecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="align-text-top"="tagcolor"
style="color:mediumblue">>text-top</span class="tagcolor"
style="color:mediumblue">>
<span class="attributevaluecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="align-text-bottom"="tagcolor"
style="color:mediumblue">>text-bottom</span class="tagcolor"
style="color:mediumblue">>

```

Try it Yourself »

Responsive Embeds

Create responsive video or slideshow embeds based on the width of the parent.

Add the `.embed-responsive-item` to any embed elements (like `<iframe>` or `<video>`) in a parent element with `.embed-responsive` and an aspect ratio of your choice:

Example

Example

```

<!-- 21:9 aspect ratio -->
<div class="attributevaluecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="embed-responsive embed-responsive-21by9"="tagcolor"
style="color:mediumblue">>
  <iframe class="attributevaluecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="embed-responsive-item" src="attributevaluecolor"
style="color:mediumblue">="..."="tagcolor" style="color:mediumblue">></iframe
class="tagcolor" style="color:mediumblue">>

```



```

</div class="tagcolor" style="color:mediumblue">>

<!-- 16:9 aspect ratio -->
<div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="embed-responsive embed-responsive-16by9"="tagcolor"
style="color:mediumblue">>
  <iframe class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="embed-responsive-item" src="attributevaluecolor"
style="color:mediumblue">="..."="tagcolor" style="color:mediumblue">></iframe
class="tagcolor" style="color:mediumblue">>
</div class="tagcolor" style="color:mediumblue">>

<!-- 4:3 aspect ratio -->
<div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="embed-responsive embed-responsive-4by3"="tagcolor"
style="color:mediumblue">>
  <iframe class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="embed-responsive-item" src="attributevaluecolor"
style="color:mediumblue">="..."="tagcolor" style="color:mediumblue">></iframe
class="tagcolor" style="color:mediumblue">>
</div class="tagcolor" style="color:mediumblue">>

<!-- 1:1 aspect ratio -->
<div class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="embed-responsive embed-responsive-1by1"="tagcolor"
style="color:mediumblue">>
  <iframe class="attributecolor" style="color:red"> class="attributevaluecolor"
style="color:mediumblue">="embed-responsive-item" src="attributevaluecolor"
style="color:mediumblue">="..."="tagcolor" style="color:mediumblue">></iframe
class="tagcolor" style="color:mediumblue">>
</div class="tagcolor" style="color:mediumblue">>

```

Try it Yourself »

Visibility

Use the `.visible` or `.invisible` classes to control the visibility of elements. **Note:** These classes do not change the CSS display value. They only add `visibility:visible` or `visibility:hidden`:

Example

I am visible

Example

```
<div class="attribute" style="color:red"> class="attributevalue" style="color:mediumblue">= "visible"= "tagcolor" style="color:mediumblue">>I am visible</div class="tagcolor" style="color:mediumblue">>
<div class="attribute" style="color:red"> class="attributevalue" style="color:mediumblue">= "invisible"= "tagcolor" style="color:mediumblue">>I am invisible</div class="tagcolor" style="color:mediumblue">>
```

Try it Yourself »

Position

Use the `.fixed-top` class to make any element fixed/stay at the **top** of the page:

Example

```
<nav class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">= "navbar navbar-expand-sm bg-dark navbar-dark fixed-
top"="tag" style="color:mediumblue">>
...
</nav class="tag" style="color:mediumblue">>
```

[Try it Yourself »](#)

Use the `.fixed-bottom` class to make any element fixed/stay at the **bottom** of the page:

Example

```
<nav class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue"> navbar navbar-expand-sm bg-dark navbar-dark fixed-
bottom="tag" style="color:mediumblue">>
...
</nav class="tag" style="color:mediumblue">>
```

Try it Yourself »

Use the `.sticky-top` class to make any element fixed/stay at the **top** of the page when you scroll past it. **Note:** This class does not work in IE11 and earlier (will treat it as `position:relative`).

Example

```
<nav class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">= "navbar navbar-expand-sm bg-dark navbar-dark sticky-
top"="tagcolor" style="color:mediumblue">>
...
</nav class="tagcolor" style="color:mediumblue">>
```

Try it Yourself »

Close icon

Use the `.close` class to style a close icon. This is often used for alerts and modals. Note that we use the `×` symbol to create the actual icon (a better looking "x"). Also note that it floats right by default:

Example



Example

```
<button class="attribute" style="color:red"> type="attributevalue"
style="color:mediumblue">button class="attributevalue"
style="color:mediumblue">close="tag" style="color:mediumblue">>&times;
</button class="tag" style="color:mediumblue">>
```

Try it Yourself »

Screenreaders

Use the `.sr-only` class to hide an element on all devices, except screen readers:

Example

```
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">sr-only="tag" style="color:mediumblue">>I will be
hidden on all screens except for screen readers.</span class="tag"
style="color:mediumblue">>
```

Try it Yourself »

Colors

As described in the [Colors](#) chapter, here is a list of all text and background color classes:

The classes for text colors are: `.text-muted`, `.text-primary`, `.text-success`, `.text-info`, `.text-warning`, `.text-danger`, `.text-secondary`, `.text-white`, `.text-dark`, `.text-body` (default body color/often black) and `.text-light`:

Example

This text is muted.

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary text.

Dark grey text.

Body text.

Light grey text.

Try it Yourself »

Contextual text classes can also be used on links, which will add a darker hover color:

Example

Muted link. [Primary link](#). [Success link](#). [Info link](#). [Warning link](#). [Danger link](#). Secondary link. Dark grey link. Body/black link. [Light grey link](#).

[Try it Yourself »](#)

You can also add 50% opacity for black or white text with the `.text-black-50` or `.text-white-50` classes:

Example

Black text with 50% opacity on white background

White text with 50% opacity on black background

[Try it Yourself »](#)

Background Colors

The classes for background colors are: `.bg-primary`, `.bg-success`, `.bg-info`, `.bg-warning`, `.bg-danger`, `.bg-secondary`, `.bg-dark` and `.bg-light`.

Note that background colors do not set the text color, so in some cases you'll want to use them together with a `.text-*` class.

Example

This text is important.

This text indicates success.

This text represents some information.

This text represents a warning.

This text represents danger.

Secondary background color.

Dark grey background color.

Light grey background color.

Try it Yourself »

Typography/Text Classes

As described in the [Typography](#) chapter, here is a list of all typography/text classes:

Class	Description	Example
<code>.display-*</code>	Display headings are used to stand out more than normal headings (larger font-size and lighter font-weight), and there are four classes to choose from: <code>.display-1</code> , <code>.display-2</code> , <code>.display-3</code> , <code>.display-4</code>	Try it
<code>.font-weight-bold</code>	Bold text	Try it
<code>.font-weight-bolder</code>	Bolder bold text	Try it
<code>.font-weight-normal</code>	Normal text	Try it
<code>.font-weight-light</code>	Light weight text	Try it
<code>.font-weight-lighter</code>	Lighter weight text	Try it
<code>.font-italic</code>	Italic text	Try it

<code>.lead</code>	Makes a paragraph stand out	Try it
<code>.small</code>	Indicates smaller text (set to 85% of the size of the parent)	Try it
<code>.text-break</code>	Prevents long text from breaking layout	Try it
<code>.text-center</code>	Indicates center-aligned text	Try it
<code>.text-decoration-none</code>	Removes the underline from a link	Try it
<code>.text-left</code>	Indicates left-aligned text	Try it
<code>.text-justify</code>	Indicates justified text	Try it
<code>.text-monospace</code>	Monospaced text	Try it
<code>.text-nowrap</code>	Indicates no wrap text	Try it
<code>.text-lowercase</code>	Indicates lowercased text	Try it
<code>.text-reset</code>	Resets the color of a text or a link (inherits the color from its parent)	Try it
<code>.text-right</code>	Indicates right-aligned text	Try it
<code>.text-uppercase</code>	Indicates uppercased text	Try it
<code>.text-capitalize</code>	Indicates capitalized text	Try it
<code>.initialism</code>	Displays the text inside an <code><abbr></code> element in a slightly smaller font size	Try it
<code>.list-unstyled</code>	Removes the default list-style and left margin on list items (works on both <code></code> and <code></code>). This class only applies to immediate children list items (to remove the default	Try it

	list-style from any nested lists, apply this class to any nested lists as well)	
<code>.list-inline</code>	Places all list items on a single line (used together with <code>.list-inline-item</code> on each <code></code> elements)	Try it
<code>.pre-scrollable</code>	Makes a <code><pre></code> element scrollable	Try it

Block Elements

To make an element into a block element, add the `.d-block` class. Use any of the `d-*-block` classes to control WHEN the element should be a block element on a specific screen width:

Example

```
d-block
d-sm-block
d-md-block
d-lg-block
d-xl-block
```

Example

```
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="d-block bg-success"="tagcolor"
style="color:mediumblue">>d-block</span class="tagcolor"
style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="d-sm-block bg-success"="tagcolor"
style="color:mediumblue">>d-sm-block</span class="tagcolor"
style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">="d-md-block bg-success"="tagcolor"
style="color:mediumblue">>d-md-block</span class="tagcolor"
```

```
style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">=d-lg-block bg-success="tagcolor"
style="color:mediumblue">>d-lg-block</span class="tagcolor"
style="color:mediumblue">>
<span class="attribute" style="color:red"> class="attributevalue"
style="color:mediumblue">=d-xl-block bg-success="tagcolor"
style="color:mediumblue">>d-xl-block</span class="tagcolor"
style="color:mediumblue">>
```

[Try it Yourself »](#)

Other Display Classes

Other display classes are also available:

Class	Description	Example
<code>.d-none</code>	Hides an element	Try it
<code>.d-*-none</code>	Hides an element on a specific screen size	Try it
<code>.d-inline</code>	Makes an element inline	Try it
<code>.d-*-inline</code>	Makes an element inline on a specific screen size	Try it
<code>.d-inline-block</code>	Makes an element inline block	Try it
<code>.d-*-inline-block</code>	Makes an element inline block on a specific screen size	Try it
<code>.d-table</code>	Makes an element display as a table	Try it
<code>.d-*-table</code>	Makes an element display as a table on a specific screen size	Try it

<code>.d-table-cell</code>	Makes an element display as a table cell	Try it
<code>.d-*-table-cell</code>	Makes an element display as a table cell on a specific screen size	Try it
<code>.d-table-row</code>	Makes an element display as a table row	Try it
<code>.d-*-table-row</code>	Makes an element display as a table row on a specific screen size	Try it
<code>.d-flex</code>	Creates a flexbox container and transforms direct children into flex items	Try it
<code>.d-*-flex</code>	Creates a flexbox container on a specific screen size	Try it
<code>.d-inline-flex</code>	Creates an inline flexbox container	Try it
<code>.d-*-inline-flex</code>	Creates an inline flexbox container on a specific screen size	Try it

Flex

Use `.flex-*` classes to control the layout with flexbox.

Read more about [Bootstrap 4 Flex](#), in our next chapter.

Example

Horizontal:



Vertical:

Flex item 1

Flex item 2

Flex item 3

Flex item 3

Flex item 2

Flex item 1

◀ Previous

Log in to track progress

Next ▶

ADVERTISEMENT

ADVERTISEMENT

