Bootstrap Cheat Sheet

Acknowledgment

Table of Contents

elevvo

Key Bootstrap Components

Bootstrap.css

This is the basic Bootstrap package that you will need to download. CSS is a style sheet language for static information.

```
<link   rel="stylesheet"   href="https://stackpath.bootstrapcdn.com/
bootstrap/4.4.1/css/bootstrap.min.css" integrity="sha384-Vkoo8x4CGsO3+Hhxv8T/
Q5PaXtkKtu6ug5TOeNV6gBiFeWPGFN9MuhOf23Q9Ifjh"  crossorigin="anonymous">
```

Bootstrap.js

A JavaScript/JQuery library that powers up certain components of Bootstrap such as animation, scrolling, and interactivity.

```
<script     src="https://code.jquery.com/jquery-3.4.1.slim.min.js"     integrity="sha384-
J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7imGFAV0wwj1yYfoRSJoZ+n"
crossorigin="anonymous"></script>

<script    src="https://cdn.jsdelivr.net/npm/popper.js@1.16.0/dist/umd/popper.min.js"
integrity="sha384-Q6E9RHvbIyZFJoft+2mJbHaEWldlvI9IOYy5n3zV9zzTtmI3UksdQRVvoxMfooAo"
crossorigin="anonymous"></script>

<script     src="https://stackpath.bootstrapcdn.com/bootstrap/4.4.1/js/bootstrap.min.js"
integrity="sha384-wfSDF2E50Y2D1uUdj0O3uMBJnjuUD4Ih7YwaYd1iqfktj0Uod8GCExl3Og8ifwB6"
crossorigin="anonymous"></script>
```

Glyphicons

Glyphs are elemental symbols with typography, such as the English Pound symbol (£). Bootstrap has a huge list of embedded glyph icons that are available for free.

Bootstrap source code elements

The Bootstrap source code download includes the precompiled CSS, JavaScript, and font assets, along with source Less, JavaScript, and documentation.

less/ - a preprocessor style sheet for CSS that eliminate repetitive coding tasks

sass/ - a newer version of the preprocessor that is more popular

js/ - simply refers to the source code JavaScript, which allows Bootstrap components to work

fonts/ - these are icon fonts that come with the download

dist/ - a folder that contains precompiled files for drop-in use in website development

Note: Bootstrap also requires jQuery installation for Bootstrap's JavaScript plugins. jQuery is a feature-rich component of the JavaScript library, and it whittles down lots of JavaScript code and wraps them into actions you can accomplish with a single line.
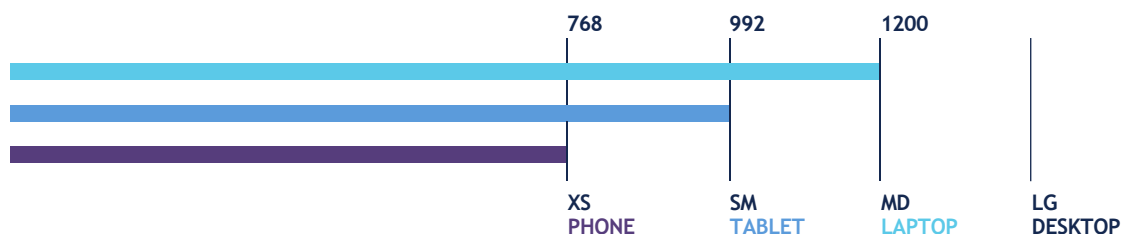
```
<script src="https://code.jquery.com/jquery.js"></script>
```

To install plug-ins:

```
<script src="js/bootstrap.min.js"></script>
```

Bootstrap Screen Sizes

Bootstrap 4 is a mobile-first responsive framework. But still you need to provide screen size instructions when you are creating page layout grids. Below are the standard screen sizes for reference:

768	992	1200

XS	SM	MD	LG
PHONE	TABLET	LAPTOP	DESKTOP

Min:
```
@media (min-width: @screen-sm-min) // >= 768px (small tablet)
@media (min-width: @screen-md-min) // >= 992px (medium laptop)
@media (min-width: @screen-lg-min) // >= 1200px (large desktop)
```

Max:
```
@media (max-width: @screen-xs-max) { // < 768px (xsmall phone)
@media (max-width: @screen-sm-max) { // < 992px (small tablet)
@media (max-width: @screen-md-max) { // < 1200px (medium laptop)
```

elevvo

Bootstrap Key Components

Alerts

Alerts are messages to users when something is wrong. For example, if a user types in an incorrect email address or credit card number, they'll receive an error alert message, prompting them to make corrections.

.alert-primary

You can create colorful alerts for any texts. Primary alert (more important message) is in light blue color.

```
<div class="alert alert-primary" role="alert">Primary alert</div>
```

.alert-secondary

Add a secondary alert (less important message) in light gray color.

```
<div class="alert alert-secondary" role="alert">I'm a secondary alert</div>
```

.alert-success

This will alert a user that their action has been successful

```
<div class="alert alert-success" role="alert">Success alert!</div>
```

.alert-warning

This will send a message of an upcoming action.

```
<div class="alert alert-warning" role="alert">Warning! Warning!</div>
```

.alert-danger

A danger alert is similar to a warning alert, but for more negative actions (e.g., getting locked out after too many password fails)

```
<div class="alert alert-danger" role="alert"> You are in grave danger, my friend!</div>
```

.alert-link

So you want to add another message and a link to that message in the original alert? You can embed that message and in the same color.

```
<div class="alert alert-success"> <strong>Success!</strong> You should <a href="#" class="alert-link">read this cool message</a>. </div>
```

.alert-dismissible

Provides an option to close the alert after reading it.

```
<div class="alert alert-success alert-dismissible"> <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a><strong>Success! You did well</strong> </div>
```

.alert-heading

Add a quick header to your alert. (e.g., "shipping address" or "billing address"). It could relate to an entire page or just a piece of content within that page.

```
<div class="alert alert-info">
<h4 class="alert-heading"><i class="fa fa-info"></i>
```

.alert-light and .alert-dark

Changes the alert style to an in either a light or dark gray color.

```
<div class="alert alert-light" role="alert"> I'm the light alert </div>
<div class="alert alert-dark" role="alert">And I'm the dark (side) alert!</div>
```

elevvo

Badge

Use badges to display extra information next to a category, button, or other element. You can create and style them with other context elements (e.g., .badge-warning). Also, badges will scale to match the size of the parent element (e.g. headings). Lastly, you can use badges as a part of buttons or links to provide counters.

Example:Headings

Suppose you have a number of headings and you are adding a badge. That badge will adjust in size to match the heading.

```
<h1>Example heading <span class="badge badge-secondary">New</span></h1>
<h2>Example heading <span class="badge badge-secondary">New</span></h2>
```

.badge-pill

Use this command to modify the shape of your badges, making them more rounded.

```
<span class="badge badge-pill badge-light">Light</span>
```

.badge-primary + .badge-secondary

You may want to add a badge to primary (more important) and secondary (less important) messages.

```
<span class="badge badge-primary">Primary</span>
<span class="badge badge-badge-secondary">Secondary</span>
```

.badge-transparent

Suppose you want to make a button transparent to make it stand out from the rest. With Bootcamp 4, you do not have to use in-line styling. Instead, use the following command:

```
<button class="btn bg-transparent"></button>
```

.badge-warning, -success, -info, or -danger

Adds a badge to a warning, success, info or danger alert.

```
<span class="badge badge-warning">Warning</span>
<span class="badge badge-success">Success</span>
<span class="badge badge-danger">Danger</span>
<span class="badge badge-info">Info</span>
```

Create Actionable Badges with Hover and Focus States

When users move their cursors to a button but have not yet activated it, you can provide hover and focus elements to encourage action.

To do this, use the contextual .badge-* classes on an `<a>` element. Here are a few examples:

```
<a href="#" class="badge badge-primary">Primary</a>
<a href="#" class="badge badge-secondary">Secondary</a>
<a href="#" class="badge badge-success">Success</a>
<a href="#" class="badge badge-danger">Danger</a>
<a href="#" class="badge badge-warning">Warning</a>
```

elevvo

.Breadcrumbs

Breadcrumbs are navigational components that will help users move from page to page without getting lost and give them the way to pedal back to a previous page.

```html
<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item active" aria-current="page">Home</li>
  </ol>
</nav>

<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="#">Home</a></li>
    <li class="breadcrumb-item active" aria-current="page">Library</li>
  </ol>
</nav>

<nav aria-label="breadcrumb">
  <ol class="breadcrumb">
    <li class="breadcrumb-item"><a href="#">Home</a></li>
    <li class="breadcrumb-item active" aria-current="page">Library</li>
    <li class="breadcrumb-item active" aria-current="page">Data</li>
  </ol>
</nav>
```

.buttons

As the name hints, .button command lets you create and style … a button.

.button-primary + .button-secondary

A primary button is commonly used for a user action; a secondary button may then be used to close out.

```html
<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-secondary">Secondary</button>
```

.btn-light, -dark .btn-primary, -secondary, -transparent, -white, -warning, -success, -info, -danger

Design your buttons using the standard predefined styling options:

```html
<button type="button" class="btn btn-light">Light</button>
<button type="button" class="btn btn-dark">Dark</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="button-transparent">Transparent</button>
<button type="button" class="button-white">White</button>
```

.btn-outline

Choose a button outline following these snippet samples:

```html
<button type="button" class="btn btn-outline-primary">Primary</button>
```

.btn-lg + .btn-sm

Change the size of your buttons.

```html
<button type="button" class="btn btn-primary btn-lg">I'm the large button</button>
<button type="button" class="btn btn-primary btn-sm">I'm a small button</button>
```

elevvo

.btn-block

Group your buttons into a block. All grouped buttons will span the full width of a parent.
```
<button type="button" class="btn btn-primary btn-lg btn-block">Block level button</button>
```

.active

By default, all buttons will be displayed as pressed - with a dark border, darker background and inset shadow - when active.

You don't need to add a class to `<button>`s as they use a pseudo-class.

But, if for some reason, you need to force that same active look, use the following code snippet:
```
<a href="#!" class="btn btn-primary btn-lg active" role="button" aria-pressed="true">Primary link</a>
```

Note: You can also add the disabled Boolean attribute to any button to make it look inactive.
```
<button type="button" class="btn btn-lg btn-primary" disabled>Disabled button</button>
```

Button Group

Use this element to make a group of similarly-sized buttons without coding each separately.

.btn-group
```
<div class="btn-group" role="group" aria-label="Basic example">
    <button type="button" class="btn btn-secondary">Left</button>
    <button type="button" class="btn btn-secondary">Middle</button>
    <button type="button" class="btn btn-secondary">Right</button>
</div>
```

.btn-group-vertical

Style a vertical group of buttons:
```
<div class="btn-group-vertical" role="group" aria-label="Basic example">
    <button type="button" class="btn btn-secondary">Left</button>
    <button type="button" class="btn btn-secondary">Middle</button>
    <button type="button" class="btn btn-secondary">Right</button>
</div>
```

.btn-group (Nested)

You can also create nested buttons with drop down menus.
```
<div class="btn-group" role="group" aria-label="Button group with nested dropdown">
  <button type="button" class="btn btn-secondary">1</button>
  <button type="button" class="btn btn-secondary">2</button>

  <div class="btn-group" role="group">
    <button id="btnGroupDrop1" type="button"
            class="btn btn-secondary dropdown-toggle"
            data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
      Dropdown
    </button>
    <div class="dropdown-menu" aria-labelledby="btnGroupDrop1">
      <a class="dropdown-item" href="#!">Dropdown link</a>
      <a class="dropdown-item" href="#!">Dropdown link</a>
    </div>
  </div>
</div>
```

elevvo

.btn-toolbar

Arrange button groups into a toolbar to make more complex components. You can apply different utility classes for additional styling.

```
<div class="btn-toolbar" role="toolbar" aria-label="Toolbar with button groups">
  <div class="btn-group" role="group" aria-label="First group">
    <button type="button" class="btn btn-secondary">1</button>
    <button type="button" class="btn btn-secondary">2</button>
    <button type="button" class="btn btn-secondary">3</button>
  </div>
  <div class="btn-group" role="group" aria-label="Second group">
    <button type="button" class="btn btn-secondary">5</button>
    <button type="button" class="btn btn-secondary">6</button>
  </div>
  <div class="btn-group" role="group" aria-label="Third group">
    <button type="button" class="btn btn-secondary">8</button>
  </div>
</div>
```

.btn-group-toggle

Install Bootstrap Toggle plugin to modify checkboxes into toggles. You can then add data-toggle="buttons" to a button group with modified buttons to enable their toggling behavior via JavaScrip. Afterwards, use .btn-group-toggle to style different inputs within your buttons.

```
<div class="btn-group-toggle" data-toggle="buttons">
  <label class="btn btn-secondary active">
    <input type="checkbox" checked autocomplete="off"> Checked
  </label>
</div>
```

Cards

Cards are flexible containers with options for headers/footers, colors and display options, and more. They replaced several earlier components (panels, wells, and thumbnails) from Bootstrap 3.

.card-body

The main element of the card. Use it to add a padded section within your card.

```
<div class="card">
  <div class="card-body">
  Your awesome text
  </div>
</div>
```

.card-title

Code a **title** for your card. Add this to a <h*> tag.

```
 <h4 class="card-title">Big title</h4>
```

.card-subtitle

You can also add subtitles to every card for some extra fanciness.

```
<h6 class="card-subtitle mb-2 text-muted">Fancy card subtitle</h6>
```

.card-link

Embed a link inside your card. Add this class to an <a> tag.

```
 <a href="#!" class="card-link">Link to something fun</a>
```

elevvo

.card-text

Add some words to your card. As many or as few as you want.

```
<p class="card-text">
Roses are red, violets are blue,
I'm learning Bootstrap,
What about you?
</p>
```

.card-img-top

You can also enclose an image to your card. This snippet will add one atop of it.

```
<div class="card">
  <img class="card-img-top" src="/images/pathToYourImage.png" alt="Card image cap">
  <div class="card-body"> What an epic image!</div>
</div>
```

.card-img-bottom

Or you can have the image displayed as a bottom of the card. Your call.

```
<div class="card">
  <div class="card-body"> Look at that pic!</div>
  <img class="card-img-bottom" src="/images/pathToYourImage.png" alt="Card image cap">
</div>
```

.card-img-overlay

Use an image as a background and overlay all the texts.

```
<div class="card">
  <img class="card-img" src="/images/pathToYourImage.png" alt="Card image">
  <div class="card-img-overlay">
    <p class="card-text">I'm text that has a background image!</p>
  </div>
</div>
```

.card-header

Place a custom header at the top of your card. It will be displayed above all the titles and subtitles.

```
<div class="card">
  <div class="card-header">
    Some big header!
  </div>
```

.card-footer

Also, you can code a footer for your card to communicate some extra info. It will go right after the .card-body.

```
<div class="card">
  <div class="card-body">
    <p class="card-text">Some more card content</p>
  </div>
  <div class="card-footer">
    Updated 2 days ago
  </div>
</div>
```

elevvo

.card-group

Play around with card layout and build a group of cards.A group will act as a single, attached element, with every card having the same width and height. You can also apply display: flex; to improve sizing.

Note: Group card layouts are not responsive!

```html
<div class="card-group">
  <div class="card">
    <div class="card-body">
      <h4 class="card-title">Card number one</h4>
      <p class="card-text">I'm the first card in the group, displaying some cool
        info.</p>
    </div>
  </div>
  <div class="card">
    <div class="card-body">
      <h4 class="card-title">Card number two</h4>
      <p class="card-text">I'm the middle card in the group and probably I offer the
        best deal</p>
    </div>
  </div>
  <div class="card">
    <div class="card-body">
      <h4 class="card-title">Card three</h4>
      <p class="card-text">I'm the third card trying to be as cool as the first
        one.</p>
    </div>
  </div>
</div>
```

.card-columns

You can organize your cards into Masonry-like columns. This allows you to build some creative patterns using only CSS.

NB: If your cards are breaking across columns, set them to display: inline-block.

```html
<div class="card-columns">
  <div class="card">
    <div class="card-body">
      <!-- Card content -->
    </div>
  </div>
  <div class="card p-3">
    <!-- Card content -->
  </div>
  <div class="card">
    <div class="card-body">
      <!-- Card content -->
    </div>
  </div>
  <div class="card bg-primary p-3 text-center">
      <!-- Card content -->
  </div>
</div>
```

.card-deck

Assemble a set of non-attached cards with equal height and width.

```html
<div class="card-deck">
  <div class="card">
    <div class="card-body">
      <h4 class="card-title">Some title</h4>
      <p class="card-text">Your texts </p>
    </div>
  </div>
  <div class="card">
    <div class="card-body">
      <h4 class="card-title">Another card</h4>
      <p class="card-text">Even more texts that someone will need to write.</p>
    </div>
  </div>
  <div class="card">
    <div class="card-body">
      <h4 class="card-title">I'm a card too!</h4>
      <p class="card-text">Some words to add and arrange</p>
    </div>
  </div>
</div>
```

Carousel

Set up a slideshow to cycle through a series of slides, text, or images. Built with CSS 3D and some JS. You can add images, text or custom markup, as well as add or remove previous/next controls.

Carousel Slide - creating a single slide

```html
<div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
  <div class="carousel-inner" role="listbox">
    <div class="carousel-item active">
      <img class="d-block w-100" src="#" data-src="holder.js/900x400?theme=social" alt="First slide">
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="#" data-src="holder.js/900x400?theme=industrial" alt="Second slide">
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```

elevvo

.carousel-fade

Add this cool fade out effect for a slide before the next one.

```html
<div id="carouselFadeExampleIndicators" class="carousel slide carousel-fade" data-ride="carousel">
  <div class="carousel-inner" role="listbox">
    <div class="carousel-item active">
      <img class="d-block w-100" src="#" data-src="holder.js/900x400?theme=social" alt="First slide">
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="#" data-src="holder.js/900x400?theme=industrial" alt="Second slide">
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselFadeExampleIndicators" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselFadeExampleIndicators" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```

.carousel-indicators

Add control and support for next/previous and indicators such as slide number.

```html
<div id="carouselExampleIndicators" class="carousel slide" data-ride="carousel">
  <ol class="carousel-indicators">
    <li data-target="#carouselExampleIndicators" data-slide-to="0" class="active"></li>
    <li data-target="#carouselExampleIndicators" data-slide-to="1"></li>
  </ol>
  <div class="carousel-inner" role="listbox">
    <div class="carousel-item active">
      <img class="d-block w-100" src="#" data-src="holder.js/900x400?theme=social" alt="First slide">
    </div>
    <div class="carousel-item">
      <img class="d-block w-100" src="#" data-src="holder.js/900x400?theme=industrial" alt="Second slide">
    </div>
  </div>
  <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button" data-slide="prev">
    <span class="carousel-control-prev-icon" aria-hidden="true"></span>
    <span class="sr-only">Previous</span>
  </a>
  <a class="carousel-control-next" href="#carouselExampleIndicators" role="button" data-slide="next">
    <span class="carousel-control-next-icon" aria-hidden="true"></span>
    <span class="sr-only">Next</span>
  </a>
</div>
```

.carousel-caption

Add a funky caption to each or several slides.

```html
<div class="bd-example">
  <div id="carouselExampleCaptions" class="carousel slide" data-ride="carousel">
    <ol class="carousel-indicators">
      <li data-target="#carouselExampleCaptions" data-slide-to="0" class="active"></li>
```

```
          <li data-target="#carouselExampleCaptions" data-slide-to="1"></li>
      </ol>
      <div class="carousel-inner" role="listbox">
        <div class="carousel-item active">
          <img class="d-block w-100" data-src="holder.
js/900x300?auto=yes&bg=777&fg=555&text=First slide" alt="First slide">
          <div class="carousel-caption d-none d-md-block">
            <h3>Cool Slide Label</h3>
            <p>I'm so proud of myself for coding this slide.</p>
          </div>
        </div>
        <div class="carousel-item">
          <img class="d-block w-100" data-src="holder.
js/900x300?auto=yes&bg=666&fg=444&text=Second slide" alt="Second slide">
          <div class="carousel-caption d-none d-md-block">
            <h3>Even Cooler label</h3>
            <p>Yep, totally rocking this whole coding thing.</p>
          </div>
        </div>
      </div>
      <a class="carousel-control-prev" href="#carouselExampleCaptions" role="button"
data-slide="prev">
        <span class="carousel-control-prev-icon" aria-hidden="true"></span>
        <span class="sr-only">Previous</span>
      </a>
      <a class="carousel-control-next" href="#carouselExampleCaptions" role="button"
data-slide="next">
        <span class="carousel-control-next-icon" aria-hidden="true"></span>
        <span class="sr-only">Next</span>
      </a>
    </div>
  </div>
</div>
```

Collapse & Accordion

Collapse is a JavaScript plugin you can use to hide content under the "collapse" menu. The collapsed element will animate it's height from 0 to its normal value when triggered and vice versa.

.collapse
Hide your content.
```
<div class="collapse" id="collapseExample">
  <div class="card card-body">
    Anim pariatur cliche reprehenderit, enim eiusmod high life accusamus terry richardson ad squid. Nihil anim keffiyeh helvetica, craft beer labore wes anderson cred nesciunt sapiente ea proident.
  </div>
</div>
```

.collapse show
Display the collapsed content.
```
<div id="demo" class="collapse show"> Some text you've decided to hide </div>
```

.collapse.options
Activate content as collapsible element; accepts optional objects
```
$('#myCollapsible').collapse({
  toggle: false
})
```

elevvo

.accordion

An extension of collabsaple behavior to cards. You can use this feature to create an accordion.

Note: You must use .accordion as a wrapper

Accordion Snippet Example with 2 group items

```
<div id="accordion" role="tablist">
  <div class="card">
    <div class="card-header" role="tab" id="headingOne">
      <h5 class="mb-0">
        <a data-toggle="collapse" href="#collapseOne" aria-expanded="true" aria-controls="collapseOne">
          Collapsible  Group  Item  #1
        </a>
      </h5>
    </div>

    <div id="collapseOne" class="collapse show" role="tabpanel" aria-labelledby="headingOne">
      <div class="card-body">
        Here's the first thing I want to hide in this card.
      </div>
    </div>
  </div>
  <div class="card">
    <div class="card-header" role="tab" id="headingTwo">
      <h5 class="mb-0">
        <a class="collapsed" data-toggle="collapse" href="#collapseTwo" aria-expanded="false" aria-controls="collapseTwo">
          Collapsible  Group  Item  #2
        </a>
      </h5>
    </div>
    <div id="collapseTwo" class="collapse" role="tabpanel" aria-labelledby="headingTwo">
      <div class="card-body">
        And here's some other text I'm styling with accordion.
      </div>
    </div>
  </div>
</div>
```

Custom Forms

Bootstrap 4 has several customized form elements that replace the browser defaults.

.custom-checkbox

As the name implies, you can build a custom checkbox for your form.

```
<div class="custom-control custom-checkbox">
  <input type="checkbox" class="custom-control-input" id="customCheck1">
  <label class="custom-control-label" for="customCheck1">I have a cool custom checkbox</label>
</div>
```

.custom-radio

…and you can style a custom radio too!

```
<div class="custom-control custom-radio">
  <input type="radio" id="customRadio1" name="customRadio" class="custom-control-input">
  <label class="custom-control-label" for="customRadio1">Give this custom radio a toggle</label>
</div>
```

.custom-switch

Lastly, you can also create a stylish custom switch.

```
<div class="custom-control custom-switch">
  <input type="checkbox" class="custom-control-input" id="customSwitch1">
  <label class="custom-control-label" for="customSwitch1">Toggle this switch element</label>
</div>
```

.custom-select

Use this command to add a custom select menu.

```
<select class="custom-select">
  <option selected>Open this select menu</option>
  <option value="1">One</option>
  <option value="2">Two</option>
  <option value="3">Three</option>
</select>
```

.custom-file

Customize user file upload. To do so, add .custom-file class around the input with type="file". Then add the .custom-file-input to it.

```
<div class="custom-file">
  <input type="file" class="custom-file-input" id="customFile">
  <label class="custom-file-label" for="customFile">Choose file</label>
</div>
```

.custom-range

Design a custom range menu.

```
<label for="customRange1">Basic range</label>
<input type="range" class="custom-range" id="customRange1">
<label class="mt-3" for="customRange3">Range with steps</label>
<input type="range" class="custom-range" min="0" max="5" step="0.5" id="customRange3">
```

Dropdowns

Use this plugin to create contextual overlays for displaying lists of user links. It's a handy option for creating menus. Dropdowns are built through Popper.js, part of a third-party library. Thus, Make sure you include popper.min.js before Bootstrap's own JavaScript. Or you can just use bootstrap.bundle.min.js/ bootstrap.bundle.js. Both contain Popper.js.

Quick styling tip: all dropdowns in Bootstrap are toggled by clicking, not hovering.

.dropdown

Add a simple dropdown menu with buttons.

```html
<div class="dropdown">
  <button class="btn btn-secondary dropdown-toggle"
          type="button" id="dropdownMenu1" data-toggle="dropdown"
          aria-haspopup="true" aria-expanded="false">
  Dropdown
  </button>
  <div class="dropdown-menu" aria-labelledby="dropdownMenu1">
    <a class="dropdown-item" href="#!">Action</a>
    <a class="dropdown-item" href="#!">Another action</a>
  </div>
</div>
```

.dropdown-toggle-split

Create split button dropdowns with proper spacing around the dropdown caret.

```html
<div class="btn-group">
  <button type="button" class="btn btn-secondary">Dropdown</button>
  <button type="button" class="btn btn-secondary dropdown-toggle dropdown-toggle-split" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
    <span class="sr-only">Toggle Dropdown</span>
  </button>
  <div class="dropdown-menu">
    <a class="dropdown-item" href="#!">Action</a>
    <a class="dropdown-item" href="#!">Another action</a>
  </div>
</div>
```

.dropup

Did you know that you can style a menu coming up rather than down? Now you do!

```html
<!-- Default dropup button -->
<div class="btn-group dropup">
  <button type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
  Dropup
  </button>
  <div class="dropdown-menu">
    <!-- Dropdown menu links -->
  </div>
</div>
```

.dropright

Provide the menu to the right of the button.

```html
<!-- Default dropright button -->
<div class="btn-group dropright">
  <button type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
  Dropright
  </button>
  <div class="dropdown-menu">
    <!-- Dropdown menu links -->
```

elevvo

```
  </div>
</div>
```

.dropleft

…And you can also display the menu on the left.
```
<!-- Default dropleft button -->
<div class="btn-group dropleft">
  <button type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
  Dropleft
  </button>
  <div class="dropdown-menu">
    <!-- Dropdown menu links -->
  </div>
</div>
```

.dropdown-item-text

Add non-interactive dropdown items to your menu.
```
<div class="dropdown-menu">
  <span class="dropdown-item-text">Plain text</span>
  <a class="dropdown-item" href="#">Clickable action</a>
  <a class="dropdown-item" href="#">Another action</a>
  <a class="dropdown-item" href="#">Whatever else you need</a>
</div>
```

.dropdown-item disabled

You can also choose to disable any menu item(s).
```
<div class="dropdown-menu">
  <a class="dropdown-item" href="#">Active link</a>
  <a class="dropdown-item disabled" href="#">Disabled link</a>
  <a class="dropdown-item" href="#">One more link</a>
</div>
```

.dropdown-divider

Add a simple divider between menu elements to draw extra attention.
```
  <div class="dropdown-divider"></div>
```

.dropdown-menu-right

Align the entire menu to the right
```
<div class="btn-group">
  <button type="button" class="btn btn-secondary dropdown-toggle" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
  This dropdown's menu is right-aligned
  </button>
  <div class="dropdown-menu dropdown-menu-right">
    <button class="dropdown-item" type="button">Action</button>
    <button class="dropdown-item" type="button">Another action</button>
    <button class="dropdown-item" type="button">Something else here</button>
  </div>
</div>
```

Forms

You can easily build attractive forms and code custom styles, layouts and additional elements. In Bootstrap 4, forms also received some new input controls such as number election, email verification and others, along with a bunch of new responsive classes.

Example of .form-group

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1" aria-describedby="emailHelp" placeholder="Provide email">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Your password</label>
    <input type="password" class="form-control" id="exampleInputPassword1" placeholder="Password">
  </div>
  <div class="form-group form-check">
    <input type="checkbox" class="form-check-input" id="exampleCheck1">
    <label class="form-check-label" for="exampleCheck1">Remember me</label>
  </div>
  <button type="submit" class="btn btn-primary">Submit</button>
</form>
```

.form-control

Use this class to style all textual form controls such as user input, titles, etc.

```
<form>
  <div class="form-group">
    <label for="exampleFormControlInput1">Email address</label>
    <input type="email" class="form-control" id="exampleFormControlInput1" placeholder="name@example.com">
  </div>
```

.form-control-file

Add this class whenever you need to provide the user with an option to upload a file to the form.

```
<input type="file" class="form-control-file" id="exampleFormControlFile1">
```

.form-control-lg and .form-control-sm.

Create a visual hierarchy within your form by adding .form-control-lg to make bigger input areas and .form-control-sm to make smaller ones.

Email	col-form-label-sm
Email	col-form-label
Email	col-form-label-lg

```
<input class="form-control form-control-lg" type="text" placeholder=".form-control-lg">
<input class="form-control form-control-sm" type="text" placeholder=".form-control-sm">
```

.form-control-plaintext

Use this class to correctly display <input readonly> elements in your form. It will replace the default form field styling with plain text, while keeping the correct margin and padding.

```
<form>
  <div  class="form-group  row">
    <label  for="staticEmail"  class="col-sm-2  col-form-label">Email</label>
    <div  class="col-sm-10">
      <input  type="text"  readonly  class="form-control-plaintext"  id="staticEmail"
value="email@example.com">
    </div>
  </div>
  <div  class="form-group  row">
    <label  for="inputPassword"  class="col-sm-2  col-form-label">Pass</label>
    <div  class="col-sm-10">
      <input  type="password"  class="form-control"  id="inputPassword"
placeholder="Password">
    </div>
  </div>
</form>
```

.form-control-range

Set horizontally scrollable range inputs for your form.

```
<form>
  <div  class="form-group">
    <label  for="formControlRange">Some  range  input</label>
    <input  type="range"  class="form-control-range"  id="formControlRange">
  </div>
</form>
```

.form-check

Add checkboxes to your form.

```
<div class="form-check">
  <input  class="form-check-input"  type="checkbox"  value=""  id="defaultCheck1">
  <label  class="form-check-label"  for="defaultCheck1">

  My first checkbox
  </label>
</div>
```

Note: You can also add radio buttons instead of checkboxes using form-check-input" type="radio".

.form-check-inline

Create a horizontal list of checkboxes.

```
<div  class="form-check  form-check-inline">
  <input class="form-check-input"  type="checkbox"  id="inlineCheckbox1"
value="option1">
  <label class="form-check-label"  for="inlineCheckbox1">1</label>
</div>
<div  class="form-check  form-check-inline">
  <input class="form-check-input"  type="checkbox"  id="inlineCheckbox2"
value="option2">
  <label class="form-check-label"  for="inlineCheckbox2">2</label>
</div>
```

.readonly boolean attribute

Specify that certain form input is read only. This will prevent modification of the input's value.

```
<input  class="form-control"  type="text"  placeholder="Readonly  input  here…"  readonly>
```

elevvo

Input Group

Input group element lets you create more interactive and attractive form controls. Use it to add texts, icons or buttons on both sides of the input field.

.input.group - Basic Example

```
<div class="input-group">
   <div class="input-group-prepend">
      <span class="input-group-text">First and last name</span>
   </div>
   <input type="text" aria-label="First name" class="form-control">
   <input type="text" aria-label="Last name" class="form-control">
</div>
```

.input-group-prepend

Provide additional texts in front of the input.

```
<form>
   <div class="input-group mb-3">
      <div class="input-group-prepend">
         <span class="input-group-text">@</span>
      </div>
      <input type="text" class="form-control" placeholder="Name">
   </div>
```

.input-group-append

…Or list them behind the input.

```
<div class="input-group mb-3">
      <input type="text" class="form-control" placeholder="Password">
      <div class="input-group-append">
         <span class="input-group-text">@example.com</span>
      </div>
   </div>
</form>
```

.input-group-text

Use this class to style specified texts.

```
<div class="input-group mb-3">
   <input type="text" class="form-control" placeholder="1000" aria-label="Amount
(rounded to the nearest dollar)" aria-describedby="basic-addon">
   <div class="input-group-append">
      <span class="input-group-text" id="basic-addon">.00</span>
   </div>
</div>
```

elevvo

Jumbotron

A flexible component that will help you create big boxes to command more attention to featured content or message. In Bootstrap, jumbotron looks like a grey box with rounded corners that automatically enlarges all the font sizes and texts inside of it.

You can add any HTML and other Bootstrap classes inside a jumbotron.

.jumbotron

```
<div class="jumbotron">
  <h1 class="display-3">Hey, awesome you!</h1>
  <p class="lead">This is a simple hero unit, showing that anyone can be a real hero!.</p>
  <hr class="my-2">
  <p>Use utility classes for typography and spacing</p>
  <p class="lead">
    <a class="btn btn-primary btn-lg" href="#!" role="button">Ready for action</a>
  </p>
</div>
```

.jumbotron-fluid

Slightly changes the look of jumbotron and makes it full-page wide without rounded corners.

```
<div class="jumbotron jumbotron-fluid">
  <div class="container">
    <h1 class="display-3">Fluid jumbotron</h1>
    <p class="lead">I'm taking up the entire horizontal space on the page.</p>
  </div>
</div>
```

List Group

Use list groups to display series of content, with lots of options for styling and layouts.

.list-group - Example

Create a basic list group with several items.

```
<ul class="list-group">
  <li class="list-group-item">Milk</li>
  <li class="list-group-item">Tea</li>
  <li class="list-group-item">Toffees</li>
</ul>
```

.list-group-item active

Add .active class to highlight the current active selection in the list.

```
<ul class="list-group">
  <li class="list-group-item active">I want this</li>
  <li class="list-group-item">Not this</li>
  <li class="list-group-item">Or this</li>
</ul>
```

.list-group-item disabled

Show that one of the list items is not available/disaibled. Some active elements (e.g. links) will require custom JavaScript on top of .disable to become fully inactive.

```
<ul class="list-group">
  <li class="list-group-item disabled">This product is out of stock</li>
  <li class="list-group-item">But we have this!</li>
  <li class="list-group-item">And also this cool thing</li>
</ul>
```

elevvo

.list-group-item-action

Add more interactivity to your list by adding styling effects (disabeled, hover, active, etc.) to individual items.

```
<div class="list-group">
   <a href="#" class="list-group-item list-group-item-action active">
  Our prime choice for you!
   </a>
   <a href="#" class="list-group-item list-group-item-action">Some good offer</a>
   <a href="#" class="list-group-item list-group-item-action">Also an option</a>
   <a href="#" class="list-group-item list-group-item-action">Something else on the
list</a>
   <a href="#" class="list-group-item list-group-item-action disabled" tabindex="-1"
aria-disabled="true">And this one's not available</a>
</div>
```

.list-group-flush

Change the look of our list by removing borders and rounded corners. All the items will be placed edge-to-edge.

```
<ul class="list-group list-group-flush">
   <li class="list-group-item">Chai Latte</li>
   <li class="list-group-item">Matcha Latte</li>
   <li class="list-group-item">Earl Grey Tea</li>
   <li class="list-group-item">Vanilla Rooibos</li>
   <li class="list-group-item">Mate</li>
</ul>
```

.list-group-horizontal

You can also set your list up horizontally rather than vertically. You can also code the list group to become horizontal starting at a certain breakpoint's min-width using .list-group-horizontal-{sm|md|lg|xl}.

NB: You can't use horizontal list groups with flush list groups at the same time.

```
<ul class="list-group list-group-horizontal">
   <li class="list-group-item">Trains</li>
   <li class="list-group-item">Planes</li>
   <li class="list-group-item">Rockets</li>
</ul>
```

.list-group-item

(light, dark primary, secondary, transparent, white, warning, success, info, danger)

Apply standard styles to individual list items.

```
<ul class="list-group">
   <li class="list-group-item">All the colors that I have </li>
<li class="list-group-item list-group-item-primary">Light blue</li>
   <li class="list-group-item list-group-item-secondary">Light gray</li>
   <li class="list-group-item list-group-item-success">Green</li>
   <li class="list-group-item list-group-item-danger">Red</li>
   <li class="list-group-item list-group-item-warning">Yellow</li>
   <li class="list-group-item list-group-item-info">Teal</li>
   <li class="list-group-item list-group-item-light">White</li>
   <li class="list-group-item list-group-item-dark">Gray</li>
</ul>
```

elevvo

Media Object

Bootstrap 4 lets you build complex, repetitive components featuring texts and some media. Media objects are a cool tool to build tweet-like elements and featured boxes. Also, they are ridiculously easy to use as they require just two classes.

.media

Use the .media wrapping and .media-body around the content to create a single media object. Here is a sample for a heading.

```
<div class="media">
  <img class="d-flex mr-3" data-src="holder.js/64x64?theme=sky" alt="Some nice placeholder image">
  <div class="media-body">
    <h5 class="mt-0">Your heading</h5>
  A key message you want to share with the world!
  </div>
</div>
```
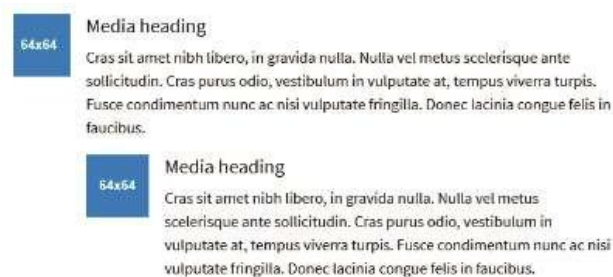
.media-body

A class specifying what would be inside of your object. You can code different alignments for your content. The default is top, but you can align in the middle or end.

```
<div class="media">
  <img src="..." class="align-self-start mr-3" alt="...">
  <div class="media-body">
    <h5 class="mt-0">Top-aligned media</h5>
    <p>Read the texts describing the picture above</p>
  </div>
```

Nested Media

You also have the option to include more than one media object. Objects are nested by beginning at left margin and tabbing each new object in.

Here is a snippet for nesting

```
<div class="media">
  <img class="d-flex mr-3" src="/images/pathToYourImage.png" alt="Generic placeholder image">
  <div class="media-body">
    <h5 class="mt-0">Media heading</h5>
  Here goes some text.
    <div class="media mt-3">
      <a class="d-flex pr-3" href="#!">
        <img src="/images/pathToYourImage.png" alt="Generic placeholder image">
      </a>
      <div class="media-body">
        <h5 class="mt-0">Another Media heading</h5>
  Some more texts explaining what's going on here.
      </div>
    </div>
  </div>
</div>
```

elevvo

Nav

.nav is a base class that helps you build all sorts of navigation components, even with style overrides. You have lots of options for customization.

.nav example

```
<ul class="nav">
  <li class="nav-item">
    <a class="nav-link disabled" href="#">You can't click this disabled link</a>
  </li>
<li class="nav-item">
    <a class="nav-link" href="#">Click this instead</a>
  </li>
</ul>
```

.nav-items

A class to specify a new item in the navigation menu.

```
</li>
<li class="nav-item">
    <a class="nav-link" href="#">Some item</a>
  </li>
```

.nav justify-content-center

Align your nav horizontally in the center.

```
<ul class="nav justify-content-center">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
  </ul>
```

.nav justify-content-end

Or justify your content to the right.
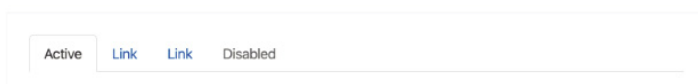
```
<ul class="nav justify-content-end">
  <li class="nav-item">
    <a class="nav-link active" href="#">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#">Link</a>
  </li>
 </ul>
```

.nav-tabs

Add some cool tabs to your navigation menu.

Note: You'll need tab JavaScript plugin installed.

```
<ul class="nav nav-tabs">
  <li class="nav-item">
    <a class="nav-link active" href="#!">Active</a>
  </li>
  <li class="nav-item">
```

```
    <a class="nav-link" href="#!">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#!">Link</a>
  </li>
</ul>
```

.nav-pills

Alternatively, you can style the menu components as pills.


```
<ul class="nav nav-pills">
  <li class="nav-item">
    <a class="nav-link active" href="#!">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#!">Link</a>
  </li>
  <li class="nav-item">
    <a class="nav-link disabled" href="#!">Disabled</a>
  </li>
</ul>
```

.Nav-justified

Equalize the widths of all tabs/pills by adding .nav-justified to .nav, .nav-tabs, or .nav, .nav-pills.

Justified Nav Elements Example
```
<ul class="nav nav-pills nav-justified">
    <li class="active"><a href="#">Home</a></li>
    <li><a href="#">HTML</a></li>
    <li><a href="#">CSS</a></li>
    <li><a href="#">PHP</a></li>
    <li><a href="#">Java</a></li>
</ul>
```

.nav-fill

Instead of justifying, you can also force your menu items to fill in all the available space using this command. However, all the items will not have the same width.
```
<ul class="nav nav-pills nav-fill">
  <li class="nav-item">
    <a class="nav-link active" href="#!">Active</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#!">This link is looooong</a>
  </li>
  <li class="nav-item">
    <a class="nav-link" href="#!">Short Link</a>
  </li>
</ul>
```

elevvo

Navbar

Navbar is a responsive navigation header with lots of flexibility and support for branding, forms, links, and more.

NavBar - basic navigation headers at top of page

```html
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <a class="navbar-brand" href="#">My first navbar</a>
  <button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <div class="collapse navbar-collapse" id="navbarSupportedContent">
    <ul class="navbar-nav mr-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#">Link 1</a>
      </li>
      <li class="nav-item">
        <a class="nav-link disabled" href="#">Link Disabled</a>
      </li>
    </ul>
    <form class="form-inline my-2 my-lg-0">
      <input class="form-control mr-sm-2" type="text" placeholder="Search" aria-label="Search">
      <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Find Stuff</button>
    </form>
  </div>
</nav>
```

.navbar-brand

Navbars come pre-furnished with support for some sub-components. This element will make your text stand out more. It was pre-designed to accommodate a product or company name.

```html
<nav class="navbar navbar-light bg-light">
  <a class="navbar-brand" href="#!">Your Company Name</a>
</nav>
```

.navbar-text

Use this class to add centered text strings vertically and horizontal spacing

```html
<nav class="navbar navbar-light bg-light">
  <span class="navbar-text">
    I'm a navbar text with an inline element
  </span>
</nav>
```

.navbar-expand (-sml -mdl -lgl -xl)

Enable responsive collapsing.

```html
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo02" aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <a class="navbar-brand" href="#!">Navbar</a>
  <div class="collapse navbar-collapse" id="navbarTogglerDemo02">
    <ul class="navbar-nav mr-auto mt-2 mt-md-0">
```

```
        <li class="nav-item active">
          <a class="nav-link" href="#!">Home <span class="sr-only">(current)</span></a>
        </li>
        <li class="nav-item">
          <a class="nav-link" href="#!">Link</a>
        </li>
      </ul>
    </div>
</nav>
```

.navbar-toggler

Program different navigation toggling behaviors (e.g. keep items active or collapse them).

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <button class="navbar-toggler navbar-toggler-right" type="button" data-
toggle="collapse" data-target="#navbarTogglerDemo02" aria-controls="navbarTogglerDemo02"
aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <a class="navbar-brand" href="#!">Navbar</a>

  <div class="collapse navbar-collapse" id="navbarTogglerDemo02">
    <ul class="navbar-nav mr-auto mt-2 mt-md-0">
      <li class="nav-item active">
        <a class="nav-link" href="#!">Home <span class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#!">Link</a>
      </li>
    </ul>
  </div>
</nav>
```

.form-inline

Place a form (e.g. search bar) in nav heading.

```
<nav class="navbar navbar-light bg-light">
  <form class="form-inline">
    <input class="form-control mr-sm-2" type="text" placeholder="Search">
    <button class="btn btn-outline-success my-2 my-sm-0" type="submit">Search</button>
  </form>
</nav>
```

.collapse.navbar-collapse

Group and collapse navbar contents by a parent breakpoint.

```html
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<a class="navbar-brand" href="#">Navbar</a>
<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#navbarSupportedContent" aria-controls="navbarSupportedContent" aria-expanded="false" aria-label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>

<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav mr-auto">
<li class="nav-item active">
<a class="nav-link" href="#">Home <span class="sr-only">(current)</span></a>
</li>
<li class="nav-item">
<a class="nav-link" href="#">Link</a>
</li>
</div>
</nav>
```

.navbar fixed-top

Set a fixed nav bar as primary and in dark color

```html
<nav class="navbar fixed-top navbar-dark bg-primary">
<a class="navbar-brand" href="#!">Fixed top</a>
</nav>
```

Bootstrap Modal (Plugin)

This is a JavaScript plugin used to add dialogs to a site, such as user notifications, custom content, and lightbox popups.

Modals consist of HTML, CSS, and JavaScript. And you can display one modal window at a time only, as Bootstrap creators deem nested models as a poor UX practice.

Modal Example

```
<div class="modal" tabindex="-1" role="dialog">
   <div class="modal-dialog" role="document">
     <div class="modal-content">
       <div class="modal-header">
         <h5 class="modal-title">Give it a title</h5>
         <button type="button" class="close" data-dismiss="modal" aria-label="Close">
           <span aria-hidden="true">&times;</span>
         </button>
       </div>
       <div class="modal-body">
         <p>Add some modal body text here.</p>
       </div>
       <div class="modal-footer">
         <button type="button" class="btn btn-primary">Agree</button>
         <button type="button" class="btn btn-secondary" data-dismiss="modal">Disagree</button>
       </div>
     </div>
   </div>
</div>
```

.modal-dialog-centered

You can choose to center all content vertically.

```
<div class="modal" tabindex="-1" role="dialog">
   <div class="modal-dialog modal-dialog-centered" role="document">
     <div class="modal-content">
       <div class="modal-header">
         <h5 class="modal-title" id="exampleModalCenterTitle">Modal title</h5>
         <button type="button" class="close" data-dismiss="modal" aria-label="Close">
           <span aria-hidden="true">&times;</span>
         </button>
       </div>
       <div class="modal-body">

List some texts.
       </div>
       <div class="modal-footer">
         <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
         <button type="button" class="btn btn-primary">Save changes</button>
       </div>
     </div>
   </div>
</div>
```

elevvo

.modal-dialog-scrollable

When you need to pack in more content into a modal (e.g. a lengthy privacy policy), you can add this class to make it scroll it independently from the page.

```
<div class="modal" id="exampleModalScrollable" tabindex="-1" role="dialog" aria-labelledby="exampleModalScrollableLabel" aria-hidden="true">
  <div class="modal-dialog modal-dialog-scrollable" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalCenteredLabel">Modal title</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
  This is a super long terms & conditions agreement  that you'll agree to without actually reading it.
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>
```

.modal fade

Enabled fading for the content.

```
<div class="modal fade" id="exampleModal3" tabindex="-1" role="dialog" aria-labelledby="exampleModal3Label" aria-hidden="true">
  <div class="modal-dialog" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModal3Label">Modal title</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      <div class="modal-body">
  Gotta right at least something here.
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-secondary" data-dismiss="modal">Close</button>
        <button type="button" class="btn btn-primary">Save changes</button>
      </div>
    </div>
  </div>
</div>
```

.modal-lg, -sm, -xl

You can also adjust the sizes of your modals.

```
<!-- Large modal -->
<button class="btn btn-primary" data-toggle="modal" data-target=".bd-example-modal-lg">
 Large modal
</button>
<div class="modal fade bd-example-modal-lg" tabindex="-1" role="dialog" aria-labelledby="myLargeModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-lg">
    <div class="modal-content">
  I'm a large modal!
    </div>
```

```
    </div>
  </div>
<!-- Small modal -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-target=".bd-example-modal-sm">

 Small modal
</button>


<div class="modal fade bd-example-modal-sm" tabindex="-1" role="dialog"
    aria-labelledby="mySmallModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-sm">
    <div class="modal-content">
  I'm a small and cute modal.
    </div>
  </div>
</div>
<!-- Huge modal -->
<button type="button" class="btn btn-primary" data-toggle="modal" data-target=".bd-example-modal-xl">

 Huge modal
</button>


<div class="modal fade bd-example-modal-xl" tabindex="-1" role="dialog"
    aria-labelledby="myHugeModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-xl">
    <div class="modal-content">
  I'm a jumbo-sized modal!
    </div>
  </div>
</div>
```

NOTE: Bootstrap offers a demo of all of these components at the Modal page. They are worth a look.

Paginators

If you know HTML, paginators are nothing new to you. With Bootstrap you have several options for styling them.

.pagination - basic pagination example

```
<nav aria-label="Page navigation example">
  <ul class="pagination">
    <li class="page-item">
      <a class="page-link" href="#!" aria-label="Previous">
        <span aria-hidden="true">&laquo;</span>
        <span class="sr-only">Previous</span>
      </a>
    </li>
    <li class="page-item"><a class="page-link" href="#!">1</a></li>
    <li class="page-item"><a class="page-link" href="#!">2</a></li>
    <li class="page-item"><a class="page-link" href="#!">3</a></li>
    <li class="page-item">
      <a class="page-link" href="#!" aria-label="Next">
        <span aria-hidden="true">&raquo;</span>
        <span class="sr-only">Next</span>
      </a>
    </li>
  </ul>
</nav>
```

elevvo

.page-item disabled

You can choose to disable one of the pagination elements or several ones.

| Previous | 1 | 2 | 3 | Next |

```
<nav aria-label="...">
  <ul class="pagination">
    <li class="page-item disabled">
      <a class="page-link" href="#" tabindex="-1" aria-disabled="true">Previous</a>
    </li>
    <li class="page-item"><a class="page-link" href="#">1</a></li>
    <li class="page-item active" aria-current="page">
      <a class="page-link" href="#">2 <span class="sr-only">(current)</span></a>
    </li>
    <li class="page-item"><a class="page-link" href="#">3</a></li>
    <li class="page-item">
      <a class="page-link" href="#">Next</a>
    </li>
  </ul>
</nav>
```

.page-item active

active indicates the current stage by highlighting it with blue.
```
<nav aria-label="...">
  <ul class="pagination">
    <li class="page-item disabled">
      <a class="page-link" href="#!" tabindex="-1">Previous</a>
    </li>
    <li class="page-item"><a class="page-link" href="#!">1</a></li>
    <li class="page-item active">
      <a class="page-link" href="#!">2 <span class="sr-only">(current)</span></a>
    </li>
    <li class="page-item"><a class="page-link" href="#!">3</a></li>
    <li class="page-item">
      <a class="page-link" href="#!">Next</a>
    </li>
  </ul>
</nav>
```

.pagination-lg

Make your pagination bigger and bolder.
```
<nav aria-label="...">
  <ul class="pagination pagination-lg">
    <li class="page-item disabled">
      <a class="page-link" href="#!" tabindex="-1">Previous</a>
    </li>
    <li class="page-item"><a class="page-link" href="#!">1</a></li>
    <li class="page-item"><a class="page-link" href="#!">2</a></li>
    <li class="page-item"><a class="page-link" href="#!">3</a></li>
    <li class="page-item">
      <a class="page-link" href="#!">Next</a>
    </li>
  </ul>
</nav>
```

elevvo

.pagination-sm

Or make it petite and less visible in size.

```
<nav aria-label="…">
  <ul class="pagination pagination-sm">
    <li class="page-item disabled">
      <a class="page-link" href="#!" tabindex="-1">Previous</a>
    </li>
    <li class="page-item"><a class="page-link" href="#!">1</a></li>
    <li class="page-item"><a class="page-link" href="#!">2</a></li>
    <li class="page-item"><a class="page-link" href="#!">3</a></li>
    <li class="page-item">
      <a class="page-link" href="#!">Next</a>
    </li>
  </ul>
</nav>
```

Popovers

Popover plugin enables you to create a pop-up box with content and other elements, activated whenever a user clicks on the element. Popovers are similar to tooltips, but fit more content.

Things to Know Before You Begin:

• You have to add the attribute, data-toggle="popover" to an element to create a popover.

• You must use the title attribute to specify header text

• Use the data-content attribute to specify what content is to be displayed in the body of the popover.

```
<a href="#" data-toggle="popover" title="Popover Header" data-content="Some content inside the popover">Toggle popover</a>
```

You have to use jQuery to initialize popovers - popover()

Popover

Here's the code that will enable popovers on your page:

```
<script>
$(document).ready(function(){
  $('[data-toggle="popover"]').popover();
});
</script>
```

Positioning Your Popovers - top, bottom, left, or right

Specify one of the four positions for your popover.

Popover on top

```
<button type="button" class="btn btn-secondary" data-container="body" data-toggle="popover" data-placement="top" data-content="Vivamus sagittis lacus vel augue laoreet rutrum faucibus.">
  Popover on top
</button>
```

Popover on right

```
<button type="button" class="btn btn-secondary" data-container="body" data-toggle="popover" data-placement="right" data-content="Vivamus sagittis lacus vel augue laoreet rutrum faucibus.">
  Popover on right
</button>
```

elevvo

Popover on bottom

```html
<button type="button" class="btn btn-secondary" data-container="body" data-toggle="popover" data-placement="bottom" data-content="Vivamus sagittis lacus vel augue laoreet rutrum faucibus.">
  Popover on bottom
</button>
```

Popover on left

```html
<button type="button" class="btn btn-secondary" data-container="body" data-toggle="popover" data-placement="left" data-content="Vivamus sagittis lacus vel augue laoreet rutrum faucibus.">
  Popover on left
</button>
```

Creating a Container Popover

Code a custom container whenever you feel that some styles on a parent element can mess up its look. Adding a container attribute will "wrap" the popover's HTML within the parent element.

```javascript
$(function () {
  $('.example-popover').popover({
    container: 'body'
  })
})
```

.popover-dismiss

Provide users with the option to close a popover when they click on the element the second time.You can arrange for the popover to dismiss when user clicks outside the element too: Use the data-trigger="focus" attribute.

By default, the popover is closed when you click on the element again. However, you can use the data-trigger="focus" attribute which will close the popover when clicking outside the element:

```html
<a href="#" title="Dismissible popover" data-toggle="popover" data-trigger="focus" data-content="Click anywhere in the document to close this popover">Click me</a>
```

Progress

Develop a custom progress bar and add additional styling elements such as animation and text labels if you like.

.progress

The class for setting up a basic progress bar. It acts as a wrapper, indicating the max value of your progress bar.

```html
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 41%" aria-valuenow="41" aria-valuemin="0" aria-valuemax="100"></div>
</div>
```

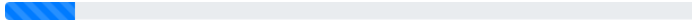## .progress-bar

Use this command to specify the current progress.

```html
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 55%" aria-valuenow="55" aria-valuemin="0" aria-valuemax="100"></div>
</div>
```

elevvo

.progress-bar-striped

Add some stripes to the progress bar section.

```
<div class="progress">
  <div class="progress-bar progress-bar-striped" role="progressbar" style="width: 10%" aria-valuenow="10" aria-valuemin="0" aria-valuemax="100"></div>
</div>
```

.progress-bar-animated

No get those stripes moving, barber's pole style.

```
<div class="progress">
  <div class="progress-bar progress-bar-striped progress-bar-animated" role="progressbar" aria-valuenow="75" aria-valuemin="0" aria-valuemax="100" style="width: 75%"></div>
</div>
```

Add Multiple Progress Bars

You can simultaneously add several bars to indicate progress for different elements. Adjust the allocation for each one up to your liking. .

```
<div class="progress">
  <div class="progress-bar" role="progressbar" style="width: 15%" aria-valuenow="15" aria-valuemin="0" aria-valuemax="100"></div>
  <div class="progress-bar bg-success" role="progressbar" style="width: 30%" aria-valuenow="30" aria-valuemin="0" aria-valuemax="100"></div>
  <div class="progress-bar bg-info" role="progressbar" style="width: 20%" aria-valuenow="20" aria-valuemin="0" aria-valuemax="100"></div>
</div>
```

elevvo

Spinner

Use this plugin to add that animated spinner. Spinners are built with HTML and CSS only, without any JS. But…you may need JavaScript for some styling options e.g. to toggle their visibility. Also, you have sveral nice options to choose from.

.spinner-border

Build a black and white loading indicator.

```
<div class="spinner-border" role="status">
 <span class="sr-only">Loading...</span>
</div>
```

.spinner-border-primary (-secondary, -light, -dark, -warning, -success, -info, -danger)

Or add a colorful border, using the standard Bootstrap styling attributes.

```
<div class="spinner-border text-primary" role="status">    <span class="sr-only">Loading...</span> </div>
<div class="spinner-border text-secondary" role="status"> <span class="sr-only">Loading...</span> </div>
<div class="spinner-border text-success" role="status"> <span class="sr-only">Loading...</span> </div>
<div class="spinner-border text-danger" role="status"><span class="sr-only">Loading...</span> </div>
<div class="spinner-border text-warning" role="status"> <span class="sr-only">Loading...</span> </div>
<div class="spinner-border text-info" role="status"><span class="sr-only">Loading...</span> </div>
<div class="spinner-border text-light" role="status"><span class="sr-only">Loading...</span></div>
<div class="spinner-border text-dark" role="status"><span class="sr-only">Loading...</span> </div>
```

Growing Spinner

Create a pulsating spinner that increases and reduces in size. A funky alternative to the spinning spinner :). Again, it can be designed in several colors.

```
<div class="spinner-grow text-primary" role="status"> <span class="sr-only">Loading...</span> </div>
```

Table

Bootstrap offers an easy-peasy way to create tables. Add base class .table to an `<table>` and add-on extra custom styles.

NB: All table styles are inherited in Bootstrap 4. Every nested table will be styled just as the parent.

.table - example
```
<table class="table">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">First</th>
      <th scope="col">Last</th>
      <th scope="col">Handle</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
```

elevvo

```html
      <th scope="row">2</th>
      <td>Jacob</td>
      <td>Thornton</td>
      <td>@fat</td>
    </tr>
    <tr>
      <th scope="row">3</th>
      <td>Larry</td>
      <td>the Bird</td>
      <td>@twitter</td>
    </tr>
  </tbody>
</table>
```

.table-dark

Create a table with a dark background and light texts.

```html
<table class="table table-dark">
  <thead>
    <tr>
      <th scope="col">#</th>
      <th scope="col">Nickname</th>
      <th scope="col">Followers</th>
      <th scope="col">Handle</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Minimark</td>
      <td>5,000</td>
      <td>@minik</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Gigi_B</td>
      <td>24K</td>
      <td>@gig</td>
    </tr>
    <tr>
      <th scope="row">3</th>
      <td>Birdie</td>
      <td>50K</td>
      <td>@bi)rdi</td>
    </tr>
  </tbody>
</table>
```

.thead-light and .thread-dark

You can also choose to highlight only one of the table entries using the dark/light attribute.

```html
<table class="table">
  <thead class="thead-dark">
    <tr>
      <th scope="col">#</th>
      <th scope="col">Episode</th>
      <th scope="col">Name</th>
      <th scope="col">Rating</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>#4</td>
      <td>New Hope</td>
      <td>10/10</td>
    </tr>
```

```html
    <tr>
      <th scope="row">2</th>
      <td>#6</td>
      <td>Return of the Jedi</td>
      <td>9/10</td>
    </tr>
    <tr>
      <th scope="row">3</th>
      <td>#9</td>
      <td>Rise of the Skywalker</td>
      <td>6/10</td>
    </tr>
  </tbody>
</table>

<table class="table">
  <thead class="thead-light">
    <tr>
      <th scope="col">#</th>
      <th scope="col">Episode</th>
      <th scope="col">Title</th>
      <th scope="col">Raiting</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Mark</td>
      <td>Otto</td>
      <td>@mdo</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>#2</td>
      <td>Revenge of the Sith</td>
      <td>7/10</td>
    </tr>
    <tr>
      <th scope="row">3</th>
      <td>#5</td>
      <td>The Empire Strikes Back</td>
      <td>10/10</td>
    </tr>
  </tbody>
</table>
```

.table-striped

Add this class to within the `<tbody>` to make your table zebra-stripey.

```html
<table class="table table-striped">
  <thead>
    <tr>
      <th>#</th>
      <th>First  Name</th>
      <th>Last  Name</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Darth</td>
      <td>Vader</td>
    </tr>
    <tr>
      <th scope="row">2</th>
```

```
      <td>Ray</td>
      <td>Just  Ray</td>
    </tr>
  </tbody>
</table>
```

.table-bordered

Style borders for all the cells and table sides.

```
<table class="table table-bordered">
  <thead>
    <tr>
      <th>#</th>
      <th>First  Name</th>
      <th>Last  Name</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Lil</td>
      <td>Kim</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Kanye</td>
      <td>West</td>
    </tr>
  </tbody>
</table>
```

.table-borderless

Ditch the borders to give your table a minimalistic flair.

```
<table class="table table-borderless">
  <thead>
    <tr>
      <th>#</th>
      <th>First  Name</th>
      <th>Last  Name</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Indiana</td>
      <td>Jones</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Han</td>
      <td>Solo</td>
    </tr>
  </tbody>
</table>
```

.table-hover

Add a hover state on table rows.

```
<table class="table table-hover">
  <thead>
    <tr>
      <th>#</th>
      <th>First  Name</th>
      <th>Last  Name</th>
    </tr>
```

elevvo

```
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Mark</td>
      <td>Twain</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Henry</td>
      <td>James</td>
    </tr>
  </tbody>
</table>
```

.table-sm

Minify your table. This class will cut the cell padding in half.

```
<table class="table table-sm">
  <thead>
    <tr>
      <th>#</th>
      <th>First Name</th>
      <th>Last Name</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">1</th>
      <td>Amy</td>
      <td>Winehouse</td>
    </tr>
    <tr>
      <th scope="row">2</th>
      <td>Kurt</td>
      <td>Cobain</td>
    </tr>
  </tbody>
</table>
```

.table-active (-primary, -default, -secondary, -success, -danger, -warning, -info, -light, -dark).

Pain individual table rows in different colors.

```
<tr class="table-active">...</tr>
<tr class="table-primary">...</tr>
<tr class="table-secondary">...</tr>
<tr class="table-success">...</tr>
<tr class="table-danger">...</tr>
<tr class="table-warning">...</tr>
<tr class="table-info">...</tr>
<tr class="table-light">...</tr>
<tr class="table-dark">...</tr>
```

elevvo

Toasts

A plugin that allows you to add "push notifications" with Flexbox - very easy to position and align. use these as an alternative to alerts.

.toast - example

```
<div class="toast" role="alert" aria-live="assertive" aria-atomic="true">
  <div class="toast-header">
    <img src="..." class="rounded mr-2" alt="...">
    <strong class="mr-auto">Bootstrap</strong>
    <small>11 mins ago</small>
    <button type="button" class="ml-2 mb-1 close" data-dismiss="toast" aria-label="Close">
      <span aria-hidden="true">&times;</span>
    </button>
  </div>
  <div class="toast-body">
  Kudos, I'm a toast message!
  </div>
</div>
```

Tooltips

Tooltips are small text pop-ups that provide users with some additional context on the button or another website element. In Bootstrap 4, tooltips use Popper.js library for positioning. That's why to use them you must include popper. min.js before bootstrap.js or use bootstrap.bundle.min.js / bootstrap.bundle.js.

Enable Tooltips

```
$(function () {$('[data-toggle="tooltip"]').tooltip()})
```

Align Tool Tips

Tooltip on top
```
<button type="button" class="btn btn-secondary" data-toggle="tooltip" data-placement="top" title="Tooltip on top"> </button>
```

Tooltip on right
```
<button type="button" class="btn btn-secondary" data-toggle="tooltip" data-placement="right" title="Tooltip on right"></button>
```

Tooltip on bottom
```
<button type="button" class="btn btn-secondary" data-toggle="tooltip" data-placement="bottom" title="Tooltip on bottom"></button>
```

Tooltip on left
```
<button type="button" class="btn btn-secondary" data-toggle="tooltip" data-placement="left" title="Tooltip on left">
```

For more advanced customizations and JavaScript methods, check the official Tooltip documentation.

elevvo

Bootstrap Styling Essentials

Breakpoints

For most components, layouts and grid systems Bootstrap uses the following breakpoint values:

Extra small < 544px

Small ≥ 544px

Medium ≥ 768px

Large ≥ 992px

Extra large ≥ 1200px

Grids and Columns

Easy to layout and align content through Flexbox; totally responsive. You get a 12-column system, 5 tiers by default, predefined classes and lots of variables.

Grid Sizing Options

.col-xs-: container width - auto/none; column width - auto.

.col-sm-: container width - 750px; column width - ~62px

.col-md-: container width - 970px ; column width - ~81px

.col-lg-: container width - 1170px; column width - ~97px

Basic .container parent grid

Create 3 equal columns with predefined grid classes

```
<div class="container">
  <div class="row">
    <div class="col-sm">
  One of three columns
    </div>
    <div class="col-sm">
  One of three columns
    </div>
    <div class="col-sm">
  One of three columns
    </div>
  </div>
</div>
```

.container-fluid

Scale your container to span over the entire viewpoint.

```
<div class="container-fluid">
 <!-- A fluid container that uses the full width -->
</div>
```

Column Sizes

Select among xs, sm, md, lg. Again, when you add columns, they have to fit the screen sizes of all devices or users will be frustrated

Here's the code snippet - just change the size element

```
<div class="container">
  <div class="row">
    <div class="col-xl-2">
      <!-- Content -->
```

elevvo

```
    </div>
    <div class="col-xl-2">
      <!-- Content -->
    </div>
    <div class="col-xl-8">
      <!-- Content -->
    </div>
  </div>
</div>
```

.no-gutters

The standard space between columns is 15px on each side. But you can remove it.

```
<div class="container">
  <div class="row no-gutters">
    <div class="col-12 col-sm-6 col-md-8">.col-12 .col-sm-6 .col-md-8</div>
    <div class="col-6 col-md-4">.col-6 .col-md-4</div>
  </div>
</div>
```

Typography: Everything you need for headings, body text, lists, and much more.

Text Alignment - left, right, centered, justified

```
<div> <p class="text-left">dotnetfunda is an online tutorial</p>
      <p class="text-center">dotnetfunda is an online tutorial</p>
      <p class="text-right">dotnetfunda is an online tutorial</p>     </div>
```

List inline - display a list in a line

```
<ul class="list-inline">
      <li>Tea</li>
      <li>Sugar</li>
      <li>Milk</li>
   </ul>
```

List-unstyled - remove the default list style.

```
<ul class="list-unstyled">
      <li>Coffee</li>
      <li>
    Tea
          <ul>
              <li>Black tea</li>
              <li>Green tea</li>
          </ul>
      </li>
      <li>Milk</li>
   </ul>
```

.text-lowercase

```
<p class="text-lowercase">lowercased text.</p>
```

.text-uppercase

```
<p class="text-uppercase">uppercased text.</p>
```

.text-capitalize
```
<p class="text-capitalize">capitalized  text.</p>
```

.text-truncate
```
<p class="text-truncate">truncated  text.</p>
```

.text-nowrap
```
<p class="text-nowrap">No  Wrap  Text.</p>
```

.text-monoscape
```
<p class="text-monospace">This is in monospace</p>
```

.text-hide
```
<h1 class="text-hide">Custom  heading</h1>
```

.text-decoration-none
```
<p class="text-decoration-none">this  text  is  not  decorated</p>
```

.initialism
Add a smaller font-size to style abbreviations.

Lead
When you want to add emphasis to a specific paragraph of content, you can make the font a bit larger and even thinner, so that the paragraph stands out visually.

```
<p class="lead">This is the info you should really pay attention to!</p>
```

Blockquote
If you are inserting a quote that is a bit lengthy, this will class allow you to set it off from the rest of the text.

```
<blockquote class="blockquote">
   <p class="mb-0">Some  cool  quote  that  you've  found</p>
</blockquote>
```

Blockquote Footer
Add the source of a quote by adding the footer element.

```
<blockquote class="blockquote">
   <p class="mb-0">Lorem  ipsum  dolor  sit  amet,  consectetur  adipiscing  elit.</p>
   <footer class="blockquote-footer">Someone  famous  in
     <cite title="Source Title">Source Title</cite>
   </footer>
</blockquote>
```

.text-break
This prevents super-long texts from breaking the look of your other design elements.
```
<p class="text-break">mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm
mmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmmm</p>
```

elevvo

Floats

.float-left - floats item left - all sizes

```
<div class="float-left">Float left on all viewport sizes</div>
<div class="float-sm-left">Float left on viewports sized SM (small) or wider</div>
<div class="float-md-left">Float left on viewports sized MD (medium) or wider</div>
<div class="float-lg-left">Float left on viewports sized LG (large) or wider</div>
<div class="float-xl-left">Float left on viewports sized XL (extra-large) or wider</div>
```

.float-right - Floats item right - all sizes

```
<div class="float-right">Float right on all viewport sizes</div>
<div class="float-sm-right">Float right on viewports sized SM (small) or wider</div>
<div class="float-md-right">Float right on viewports sized MD (medium) or wider</div>
<div class="float-lg-right">Float right on viewports sized LG (large) or wider</div>
<div class="float-xl-right">Float right on viewports sized XL (extra-large) or wider</div>
```

.float-none - Removes float - all sizes

```
<div class="float-none">Don't float on all viewport sizes</div>
<div class="float-sm-none">Don't float on viewports sized SM (small) or wider</div>
<div class="float-md-none">Don't float on viewports sized MD (medium) or wider</div>
<div class="float-lg-none">Don't float on viewports sized LG (large) or wider</div>
<div class="float-xl-none">Don't float on viewports sized XL (extra-large) or wider</div>
```

.clearfix - Removes float from parent element
```
<div class="clearfix">…</div>
```

Flex

You can easily and quickly manage your layouts, alignments, sizings, navs, displays, colors, alignment, and much more with this single Bootstrap utility tool.

Flex - sm-lg-md-xl
Size all of your elements using predetermined sizes and be fully responsive.

```
<div class="d-flex flex-sm-column">…</div>
<div class="d-flex flex-md-column">…</div>
<div class="d-flex flex-lg-column">…</div>
<div class="d-flex flex-xl-column">…</div>
```

.flex-row and .flex-row-reverse
Set horizontal direction or reverse it to start from the opposite side.

```
<div class="d-flex flex-row">
  <div class="p-2">Flex item 1</div>
  <div class="p-2">Flex item 2</div>
  <div class="p-2">Flex item 3</div>
</div>
```

elevvo

```
<div class="d-flex flex-row-reverse">
  <div class="p-2">Flex item 1</div>
  <div class="p-2">Flex item 2</div>
  <div class="p-2">Flex item 3</div>
</div>
```

.flex-column and .flex-column-reverse
Set direction of columns or reverse to set from opposite direction.

```
<div class="d-flex flex-column">
  <div class="p-2">Flex item 1</div>
  <div class="p-2">Flex item 2</div>
  <div class="p-2">Flex item 3</div>
</div>
```

```
<div class="d-flex flex-column-reverse">
  <div class="p-2">Flex item 1</div>
  <div class="p-2">Flex item 2</div>
  <div class="p-2">Flex item 3</div>
</div>
```

.Flex-fill
This utility forces all elements to occupy an equal width while taking up all the available horizontal space.
```
<div class="d-flex text-white">
  <div class="p-2 flex-fill bg-primary">Flex item</div>
  <div class="p-2 flex-fill bg-success">Flex item</div>
  <div class="p-2 flex-fill bg-primary">Flex item</div>
</div>
```

.Flex-grow-*
One of the flex items will fill all the available space, while ensuring that others also have minimally sufficient space.
```
<div class="d-flex text-white">
  <div class="p-2 flex-grow-1 bg-primary">Flex item</div>
  <div class="p-2 bg-success">Flex item</div>
  <div class="p-2 bg-primary">Third flex item</div>
</div>
```

.Flex-shrink-*
Activate the item's ability to shrink to the most minimal size to fill in the available space.

```
<div class="d-flex text-white">
  <div class="p-2 w-100 bg-primary">Flex item</div>
  <div class="p-2 flex-shrink-1 bg-success">Flex item</div>
</div>
```

.flex-nowrap, .flex-wrap-reverse   and .flex-wrap,
Use these classes when you want items either to wrap to the next line, or to fit on a single line (nowrap), or do the reverse wrapping act. (wrap-reverse).

```
<div class="d-flex flex-nowrap"> ... </div>
<div class="d-flex flex-wrap"> ... </div>
<div class="d-flex flex-wrap-reverse"> ... </div>
```

elevvo

.flex justify-content

Add this class when you want to justify text on right or left, and when you want that justification to start and end.

```html
<div class="d-flex justify-content-start">
  <div class="p-2">Flex item 1</div>
  <div class="p-2">Flex item 2</div>
  <div class="p-2">Flex item 3</div>
</div>

<div class="d-flex justify-content-end">
  <div class="p-2">Flex item 1</div>
  <div class="p-2">Flex item 2</div>
  <div class="p-2">Flex item 3</div>
</div>
```

.flex align-items

Specify when the alignment begins and when it ends for a list of items.

```html
<div class="d-flex align-items-start">
  <div class="p-2">Flex item 1</div>
  <div class="p-2">Flex item 2</div>
  <div class="p-2">Flex item 3</div>
</div>

<div class="d-flex align-items-end">
  <div class="p-2">Flex item 1</div>
  <div class="p-2">Flex item 2</div>
  <div class="p-2">Flex item 3</div>
</div>
```

.flex align-self-start

You can also override a containers alignment for a selected item within that container

```html
<div class="align-self-start">Aligned flex item</div>
```

elevvo

Alignment

How and where do you want to align your content, buttons, etc. - right, left, center?

.align-baseline

```
<div class="align-self-baseline">Aligned flex item</div>
```

.align-top -middle - bottom: place buttons where you want.

```
<span class="align-baseline">baseline</span>
<span class="align-top">top</span>
<span class="align-middle">middle</span>
<span class="align-bottom">bottom</span>
```

.align-text-top -bottom: place text where you want

```
<span class="align-text-top">text-top</span>
<span class="align-text-bottom">text-bottom</span>
```

Borders

Borders add elegance to a site. Use them for emphasis or aesthetics

.border

```
<span class="border border-dark">Hello World</span>
```

.border-light, -dark -primary, -secondary, -warning, -success, -danger, -info. -white)
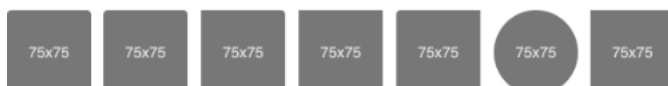
```
<span class="border border-light"></span>
<span class="border border-dark"></span>
<span class="border border-primary"></span>
<span class="border border-secondary"></span>
<span class="border border-warning"></span>
<span class="border border-success"></span>
<span class="border border-danger"></span>
<span class="border border-info"></span>
<span class="border border-white"></span>
```

Border- (top, right, bottom, left,)

```
<span class="p-1 border border-primary border-0">Hello World</span>
<span class="p-1 border border-primary border-top-0">Hello World</span>
<span class="p-1 border border-primary border-right-0">Hello World</span>
<span class="p-1 border border-primary border-bottom-0">Hello World</span>
<span class="p-1 border border-primary border-left-0">Hello World</span>
```

Border Corner Styling Options

```
<img src="..." alt="..." class="rounded">
<img src="..." alt="..." class="rounded-top">
<img src="..." alt="..." class="rounded-right">
<img src="..." alt="..." class="rounded-bottom">
<img src="..." alt="..." class="rounded-left">
<img src="..." alt="..." class="rounded-circle">
<img src="..." alt="..." class="rounded-0">
```

elevvo

Position- configure the position of elements

.sticky-top
Position an element at the top of viewport, going from edge to edge.
```
<div class="sticky-top bg-primary">Sticky top</div>
```

.fixed-top, .fixed-bottom
Place an element at the top or bottom of the viewport, from edge to edge.

```
<div class="fixed-top">Something atop</div>
<div class="fixed-bottom">Something at the bottom</div>
```

Shadows - add decorative contrasts for extra appeal.

.shadow
```
<div class="shadow p-3 mb-5 bg-light rounded">Regular shadow</div>
```

.shadow-none
```
<div class="shadow-none p-3 mb-5 bg-light rounded">No shadow</div>
```

.shadow-sm
```
<div class="shadow-sm p-3 mb-5 bg-light rounded">Small Shadow shadow</div>
```

.shadow-lg
```
<div class="shadow-lg p-3 mb-5 bg-white rounded">Larger shadow</div>
```

Visibility - tune the visibility of any element

Visible
```
<div class="visible">...</div>
```

Invisible
```
<div class="invisible">...</div>
```

elevvo