

BOOTSTRAP - 4

ANECO ACADEMY

Introduction to bootstrap

- ▣ It contains mobile first styles throughout the entire library, instead of using them in the separate files.
- ▣ With just the knowledge of HTML and CSS anyone can get started with Bootstrap. Also the Bootstrap official site has a good documentation.
- ▣ It is supported by all popular browsers and its responsive CSS adjusts to Desktops, Tablets and Mobiles.
- ▣ Provides a clean and uniform solution for building an interface for developers.
- ▣ It contains beautiful and functional built-in components which are easy to customize.
- ▣ It is an open source and provides web based customization.

Installation setup

CSS

Copy-paste the stylesheet `<link>` into your `<head>` before all other stylesheets to load our CSS.

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" : 
```

JS

Many of our components require the use of JavaScript to function. Specifically, they require [jQuery](#), [Popper.js](#), and our own JavaScript plugins. Place the following `<script>`s near the end of your pages, right before the closing `</body>` tag, to enable them. jQuery must come first, then Popper.js, and then our JavaScript plugins.

We use [jQuery's slim build](#), but the full version is also supported.

`<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UEZmM.">`
`<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-Aj" type="text/javascript">`
`<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js" integrity="sha384-1" type="text/javascript">`

Difference between container and container-fluid

The `.container` class is used to wrap the page content with fixed width and content can be placed in the center easily by using the `.container` class as shown below.

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


Difference between container and container-fluid

Fixed Width Container

This is a simple web page with fixed width container by using
`.container` class.

Difference between container and container-fluid

You can create a full width container by using the *.container-fluid* class as shown below.

```
<div class = "container-fluid">  
    ...  
</div>
```

Difference between container and container-fluid

Full Width Container

This is a simple web page with full width container by using `.container-fluid` class.

GRID 12 SYSTEM

- Rows must be placed within a *.container* class for proper alignment and padding.
- For responsive width use *.container* class and for fixed width across all viewport, use the *.container-fluid* class.
- Use rows to create horizontal groups of columns.
- Content should be placed within the columns, and only columns may be the immediate children of rows.
- Columns contain padding for controlling the space between them.
- If you place more than 12 columns in a row, then the columns will be placed in a new line.
- Columns create gaps between column content via padding. Therefore, you can remove the margin from rows and padding from columns with *.no-gutters* class on the row.
- You can make grid system responsive by using five grid breakpoints such as extra small, small, medium, large, and extra large.
- Predefined grid classes like *.col-4* are available for quickly making grid layouts. LESS mixins can also be used for more semantic layouts.

GRID 12 SYSTEM

	Extra small devices (<576px)	Small devices (≥576px)	Medium devices (≥768px)	Large devices (≥992px)	Extra Large devices (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints	Collapsed to start, horizontal above breakpoints
Max container width	None (auto)	540px	720px	960px	1140px
Class classes	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-
# of columns	12	12	12	12	12
Gutter width	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)	30px (15px on each side of a column)
Nestable	Yes	Yes	Yes	Yes	Yes
Column ordering	Yes	Yes	Yes	Yes	Yes

GRID 12 SYSTEM

```
<div class = "container">
  <div class = "row">
    <div class = "col-*-*"></div>
    <div class = "col-*-*"></div>
  </div>

  <div class = "row">
    <div class = "col-*-*"></div>
    <div class = "col-*-*"></div>
    <div class = "col-*-*"></div>
  </div>

  <div class = "row">...</div>
</div>
```

TEXT - COLOR

```
<p class="text-muted">This text is muted.</p>
<p class="text-primary">This text is important.</p>
<p class="text-success">This text indicates success.</p>
<p class="text-info">This text represents some information.</p>
<p class="text-warning">This text represents a warning.</p>
<p class="text-danger">This text represents danger.</p>
<p class="text-secondary">Secondary text.</p>
<p class="text-dark">This text is dark grey.</p>
<p class="text-body">Default body color (often black).</p>
<p class="text-light">This text is light grey (on white background).</p>
<p class="text-white">This text is white (on white background).</p>
...
```

TEXT-COLOR

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.

This text is dark grey.

Default body color (often black).

BUTTON-COLOR

```
<button type="button" class="btn">Basic</button>  
<button type="button" class="btn btn-primary">Primary</button>  
<button type="button" class="btn btn-secondary">Secondary</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-warning">Warning</button>  
<button type="button" class="btn btn-danger">Danger</button>  
<button type="button" class="btn btn-dark">Dark</button>  
<button type="button" class="btn btn-light">Light</button>  
<button type="button" class="btn btn-link">Link</button>  
...
```

BUTTON-COLOR



TABLE DESIGN

```
<table class="table table-bordered">
  <thead>
    <tr>
      <th>Firstname</th>
      <th>Lastname</th>
      <th>Email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>John</td>
      <td>Doe</td>
      <td>john@example.com</td>
    </tr>
    <tr>
      <td>Mary</td>
      <td>Moe</td>
      <td>mary@example.com</td>
    </tr>
  </tbody>
</table>
```

TABLE DESIGN

Firstname	Lastname	Email
John	Doe	john@example.com
Mary	Moe	mary@example.com
July	Dooley	july@example.com

FORM DESIGN

```
<div class="container">
  <h2>Stacked form</h2>
  <form action="/action_page.php">
    <div class="form-group">
      <label for="email">Email:</label>
      <input type="email" class="form-control" id="email" placeholder="Enter email" name="email">
    </div>
    <div class="form-group">
      <label for="pwd">Password:</label>
      <input type="password" class="form-control" id="pwd" placeholder="Enter password" name="pswd">
    </div>
    <div class="form-group form-check">
      <label class="form-check-label">
        <input class="form-check-input" type="checkbox" name="remember"> Remember me
      </label>
    </div>
    <button type="submit" class="btn btn-primary">Submit</button>
  </form>
</div>
```

FORM-DESIGN

Stacked form

Email:

Password:

☐ Remember me

MODAL

Modal Example

Open modal

Modal Heading

X

Modal body..

Close

NAVBAR DESIGN

Active Link Link Disabled

Active Link Link Disabled

Active Link Link Disabled

Logo Link 1 Link 2 Dropdown link ▾

Search

Search

Logo Link Link Disabled

Search

Search