**1. What is Bootstrap?**

- Bootstrap is a front-end framework. That means that it’s used on the client-side, not the server.

- It includes bits of code and HTML, CSS and also JS

- Used for building responsive, mobile-first web apps and websites

**2. Why use Bootstrap?**

- The one reason is that Increase development speed, not just a front-end framework, but any kind of framework PHP. With bootstrap, all your helper classed for alignments and style and positioning

- Responsiveness is another huge advantage instead of using custom media queries, and creating a responsive site from scratch you can simply use the grid system and you’ll have a mobile-friendly website

- Consistency: with class names style markup keeps you’re your website and your theme very consistent

- Customizable: you can add your own classes and your own design to any Boostrap theme and it allows you to create custom interfaces

- Support: It has documentation.

**3. HTML/ CSS Components and Helpers**

- Navbar and Dropdowns

- Jumbotron

- Alerts and progress bars

- Form and Input

- Buttons and Links

- Labels and Badges

- Icons

- List groups

- Wells and Panels

- Pagination

- Thumbnails

- Tables

- Grid System

- Responsive Utilities

**4. JavaScript Widgets and Helpers**

- Transitions

- Models

- Dropdowns

- Tabs

- Carousel / Slider

- Alerts

- Popovers

- Tooltips

- Collapsable Elements

**5. Creating Your First Web Page with Bootstrap**

We recommend adding Bootstrap in your project via CDN (Content Delivery Network) because CDN offers performance benefit by reducing the loading time , since they are hosting the files on multiple servers spread across the globe so that when a user requests the file it will be served from the server nearest to them. We're also using the CDN links in our examples:

**6. :root in Bootstrap**

:root {

--blue: #007bff;

--indigo: #6610f2;

--purple: #6f42c1;

--pink: #e83e8c;

--red: #dc3545;

--orange: #fd7e14;

--yellow: #ffc107;

--green: #28a745;

--teal: #20c997;

--cyan: #17a2b8;

--white: #fff;

--gray: #6c757d;

--gray-dark: #343a40;

--primary: #007bff;

--secondary: #6c757d;

--success: #28a745;

--info: #17a2b8;

--warning: #ffc107;

--danger: #dc3545;

--light: #f8f9fa;

--dark: #343a40;

--breakpoint-xs: 0;

**--breakpoint-sm: 576px;**

**--breakpoint-md: 768px;**

**--breakpoint-lg: 992px;**

**--breakpoint-xl: 1200px;**

--font-family-sans-serif: -apple-system,BlinkMacSystemFont,"Segoe UI",Roboto,"Helvetica Neue",Arial,"Noto Sans",sans-serif,"Apple Color Emoji","Segoe UI Emoji","Segoe UI Symbol","Noto Color Emoji";

--font-family-monospace: SFMono-Regular,Menlo,Monaco,Consolas,"Liberation Mono","Courier New",monospace;

}

**7. body in Bootstrap**

body {

margin: 0;

font-family: -apple-system,BlinkMacSystemFont,"Segoe UI",Roboto,"Helvetica Neue",Arial,"Noto Sans",sans-serif,"Apple Color Emoji","Segoe UI Emoji","Segoe UI Symbol","Noto Color Emoji";

font-size: 1rem;

font-weight: 400;

line-height: 1.5;

color: #212529;

text-align: left;

background-color: #fff;

}

**Lesson 0: Analyse**

**Step 1 : Analyse**

1- card-top = {

        img-top + {overlay = icon-heart }

      }

2 -{ card-body = {

                  card-title,

                  icon-clock, icon-users

                  card-text,

                  a.btn,

                  card-footer = {

                    small-time,

                    icon-views, icon-comment

                  }

                }

              }

**Step 2: give a name of each element**

**Step 3: coding**

**Lesson 2:**

**Bootstrap Grid System**

## 1. What is Bootstrap Grid System

- Bootstrap grid system provides the quick and convenient way to create responsive website lauouts.

- The latest Bootstrap 4 version introduces the new mobile –first **flexbox grid system** that appropriately scales up to **12 columns** as the device or viewport size increases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Features**  **Bootstrap 4 Grid System** | **Extra small**  **<576px** | **Small**  **≥576px** | **Medium**  **≥768px** | **Large**  **≥992px** | **Extra large**  **≥1200px** |
| Max container width | None (auto)  = 100%width | 540px | 720px | 960px | 1140px |
| Ideal for | Mobile (Portrait) | Mobile (Landscape) | Tablets | Laptops | Laptops & Desktops |
| Class prefix | .col- | .col-sm- | .col-md- | .col-lg- | .col-xl- |
| Number of columns | 12 | | | | |
| Gutter width | 30px (15px on each side of a column) =  Left/right **padding** between columns | | | | |
| Nestable | Yes | | | | |
| Column ordering | Yes | | | | |

- col-sm-\* class to an element will not only affect its styling on small devices, but also on medium, large and extra large devices having screen width greater than or equal to 540px, if a .col-md-\*, .col-lg-\* or .col-xl-\* class is not present.

- Similarly, the .col-md-\* class will not only affect the styling of elements on medium devices, but also on large and extra large devices if a .col-lg-\* or .col-xl- class is not present.

2. How to create rows and columns using this 12 column responsive grid system

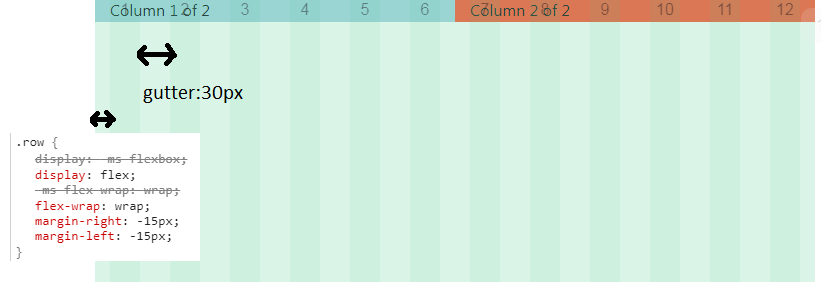
- In .container class

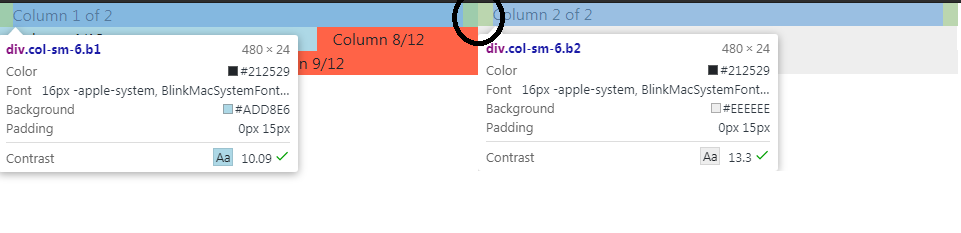
+ Step 1: creat **.row**  inside the container

+ Step 2: to create columns inside any row:

You can use the .col-\*, .col-sm-\*, .col-md-\*, .col-lg-\* and .col-xl-\* classes

3. Creating 2 column layouts





Example: vd 3/12 and 6/12

=> 

- col-sm-6

=> place on right size : 6 /12 - width = 6/12 = ½ = 50%

=> and the remaining is blank

<div class=”container”>

<div class="row">

       <div class="col-sm-4 b1">

          Column 4/12

        </div>

        <div class="col-sm-6 b2">

          Column 6/12

        </div>

      </div>

b/ 3 /12 => width = 1/4 =25% -->

        <div class="col-sm-3 b1">Column left</div>

4.  <!--

        If in mobile phone < 576px

        The columns will automatically become horizontal

       -->

5. If more than 12 grid columns within a single row:

- then each group of extra columns, as a whole, will wrap onto a new line. See [column wrapping behavior](https://www.tutorialrepublic.com/twitter-bootstrap-tutorial/bootstrap-grid-system.php#column-wrapping).

<div class="row">

        <div class="col-sm-3 b1">

          Column 3/12

        </div>

        <div class="col-sm-12 b2">

          Column 9/12

        </div>

      </div>

Result:



6. Creating variable width columns: using “col-{breakpoint}-auto”

<!-- creating variable width columns -->

    <!--

      Using "col-{breakpoint}-auto" =  fit-content

     -->

    <h3>8 - Creating variable width columns using "col-{breakpoint}-auto" classes</h3>

    <div class="container">

      <div class="row justify-content-md-center">

        <!-- 3 /12 =>width = 1/4 =25% -->

        <div class="col-sm-3 b1">Column left</div>

        <div class="col-sm-auto b2">variable width column</div>

        <div class="col-sm-3 b1">Column right</div>

      </div>

    </div>

**Lesson 3:**

**Form**

**1/ Label**



**2/ input is nested into label**

        => we only need to click Football (text)

        => checkbox is also clicked



<label class="d-block">

        <input type="checkbox" class="mr-1" name="sports">

        Tennis

      </label>

**3/ { for } attribute of label**

<label for=”A”>

<input id=”A”>

=> This is not required in Bootstrap

=> This semantially make sense

          <label for="inputEmail">Email</label>

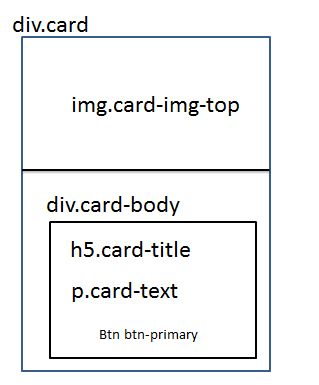
          <input type="email" class="form-control" name="email"

            id="inputEmail" placeholder="Email" required/>

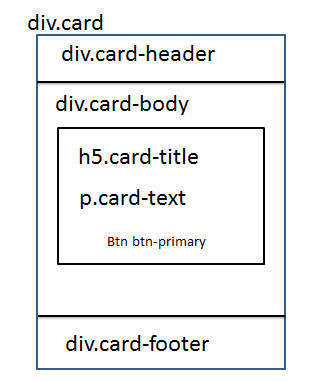
**Lesson 10:**

**Cards (Panel in B3)**

**1/ Basic Cards**



**2/ Header and footer**



**3. Click to whole card:**

- Using { .stretched-link } in a link <a> tag

 <!--

      Adding { .stretched-link } to a link <a> inside the card

      to make the whole card clickable

      .stretched-link::after {

        position: absolute;

        top: 0;

        right: 0;

        bottom: 0;

        left: 0;

        z-index: 1;

        pointer-events: auto;

        content: "";

        background-color: rgba(0,0,0,0);

      }

    -->

    <div class="container">

      <div class="card" style="width: 18rem">

        <img src="https://www.tutorialrepublic.com/examples/images/thumbnail.svg"

          class="card-img-top" alt="img">

        <div class="card-body text-center">

          <h5 class="card-title">Alice Wonder</h5>

          <p class="card-text">

            Lorem ipsum dolor sit amet consectetur adipisicing elit.

            Cum praesentium vero laboriosam facere sed nihil ratione

            corporis. Quisquam ab quas eius sunt,

            totam ea eaque dolor, harum, ratione natus nisi.

          </p>

          <a href="#" class="btn btn-primary stretched-link">

            View Profile

          </a>

        </div>

      </div>

    </div>

**4. disable the first option in <select> tag**

**=>** so we don’t want to choose that default (first) value

4-1

<select>

                    <option disabled selected>Color</option>

                    <option value="1">Green</option>

                    <option value="2">Blue</option>

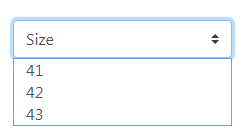
                    <option value="3">Red</option>

                  </select>

4-2: add hidden

<option selected hidden>Size</option>

=> <option> Size will be hidden on the list of select when we click on select



  .custom-select {

    display: inline-block;

    width: 100%;

    height: calc(1.5em + .75rem + 2px);

    padding: .375rem 1.75rem .375rem .75rem;

    font-size: 1rem;

    font-weight: 400;

    line-height: 1.5;

    color: #495057;

    vertical-align: middle;

    background: url(data:image/svg+xml,%3csvg xmlns='http://www.w3.org/2000/svg' viewBox='0 0 4 5'%3e%3cpath fill='%23343a40' d='M2 0L0 2h4zm0 5L0 3h4z'/%3e%3c/svg%3e) no-repeat right .75rem center/8px 10px;

    background-color: #fff;

    border: 1px solid #ced4da;

    border-radius: .25rem;

    -webkit-appearance: none;

    -moz-appearance: none;

    appearance: none;

}

|  |  |
| --- | --- |
| p | font-size: 1rem = 16px    line-height: 1.5              p {                margin-top: 0;                margin-bottom: 1rem;              } |
| h1 | Heading Text = Size x Scale  .h1, h1 {          font-size: 2.5rem;        }        .h1, .h2, .h3, .h4, .h5, .h6, h1, h2, h3, h4, h5, h6 {        margin-bottom: .5rem;        font-weight: 500;        line-height: 1.2;      }      h1, h2, h3, h4, h5, h6 {        margin-top: 0;        margin-bottom: .5rem;      } |
| .container | .container class is the important class          It set <h1> to be responsive when browser bigger or smaller          @media (min-width: 1200px)            .container {              max-width: 1140px;            }          @media (min-width: 992px)            .container {            max-width: 960px;          }            @media (min-width: 768px)            .container {              max-width: 720px;          }            @media (min-width: 576px)            .container {             max-width: 540px;            }          .container {            width: 100%;            padding-right: 15px;            padding-left: 15px;            margin-right: auto;            margin-left: auto;          }  \*, ::after, ::before {        box-sizing: border-box;      }  - Use **.container** for a "responsive pixel width"   - Or **.container-fluid** for width:  100% across all viewport and device sizes.  Note: .container ko co margin-bottom |
| .col | 1 - all .col on the same .row, hav same width  <div class="row">          <div class="col order-2 b1">            First, but ordered at last          </div>          <div class="col order-1 b2">            Last, but ordered at first          </div>        </div> |
| 2-  .col- {number} | 2 – for all device, from small mobile -> extra large desktop |
| d-\* | d-block: display block  d-flex: display flex  d-inline: display inline level box |
| .mr- \* | Set the right margin of an element.  Number \* 0.25rem = margin  Ex:  mr-1 {  Margin-right: 0.25rem;  } |
| .ml- \* | Set the left margin of an element. |
| .row | .row {              display: -ms-flexbox;              display: flex;              -ms-flex-wrap: wrap;              flex-wrap: wrap;              margin-right: -15px;              margin-left: -15px;            }  => gutter = 15\*2 = 30px     .form-row {              display: -ms-flexbox;              display: flex;              -ms-flex-wrap: wrap;              flex-wrap: wrap;              margin-right: -5px;              margin-left: -5px;            }  => gutter = 5\*2 = 10px |
| .col-form-label | <!--              Adding .col-form-label to make label to be              vertically centered with form-control (<input> text, email, password)                .col-form-label {                padding-top: calc(.375rem + 1px);                padding-bottom: calc(.375rem + 1px);                margin-bottom: 0;                font-size: inherit;                line-height: 1.5;              }  <div class="form-group row">            <label for="inputPassword" class="col-sm-2 col-form-label">              Password            </label>            <div class="col-sm-10">              <input type="password" class="form-control"                id="inputPassword" placeholder="Password">            </div>          </div> |
| .form-group  .form-check | .form-check {   1. position: relative; 2. display: block; 3. padding-left: 1.25rem;   }  \_forms.scss:171  .form-group {   1. margin-bottom: 1rem;   }  .form-check-input {            position: absolute;            margin-top: .3rem;            margin-left: -1.25rem;          }          .form-check-label {            margin-bottom: 0;          } |
| .bg-\* | Background   |  |  | | --- | --- | | .bg-primary | Apply blue background-color (#007bff) on an element. | |