

20-9-20

Bootstrap

what is Bootstrap?

- It is a front end (or) UI framework, sometimes called CSS framework.
- It is developed for faster web development / UI development.
- It gives us tons of pre-made HTML / CSS templates & classes to create websites / Application UI's.
- we get all type of utility classes from margin / padding, Typography as well as component classes and it also has pre-made templates for things like large progressive bars etc.
- we also get Javascript widgets & plugins.
- Mobile first, responsive layout.
- To do the same with HTML / CSS, it takes a lot of time & knowledge.

(1) Browser compatibility - small Modern Browsers.

(2) faster development with pre-made classes, styling and Responsive Grid.

(3) Good for dev's that lack design / UI skills.

(4) Easy to learn.

we don't need to write Media Queries, simply we can use grids for responsiveness.

ways to Install / use Bootstrap.

- Include the CDN (Content Delivery Network). It is fast because files are served from high capacity Infrastructure. It doesn't bog up our server, improves performance & saves money.

→ unless, we are compiling our own files from SaaS, the CDN is the way to go and that's what we will be doing aside from Project where we compile our SaaS.

(2)

→ we can also download the files & include files from Bootstrap.com.

But in this, It is not suggested over CDN unless we are running the files locally without an Internet connection.

(3)

we can use Package Managers. Like
• Node Package Manager
• Yarn.

Complicated, but this need to be done from command Line.

UI Utilities

(1) Grid System → which uses flexbox and it is completely responsive.

(2) Test & Typography classes → for alignment & font Sizing

(3) Positioning classes → we have classes to display as

- Blocks:
- Inline:
- Inline Blocks-

(4) We have Spacing classes for Margin & Padding

(5) we have sizing classes for widths & heights

- (6) we have Alignment classes → for text and other elements.
- (7) we have colors for Backgrounds & text & Buttons
- (8) we have Images classes for making Images fit into that container (or) Making Images circular.
- (9) we have Responsive utilities i.e., if we want to show something on large Screen (or) only on Small Screen.
- (10) we have Shadow classes (or) Border classes.
- (11) forms & Input Groups
- (12) Tables
- (13) List Groups.
- (14) Cards. (displays content in many different ways, if we want a background, border with some padding)
- (15) Progress Bars (change colors, create animations, change width & Height).
- (16) Alerts (Suppose if we want say that we have been logged in (or) we have been logged out and for error message).
- (17) Navbars & Menus
- (18) Buttons & Button Groups.
- (19) Pagination.
- (20) Media objects for newsites & Blogs.
- (21) Jumbotron → for big showcase area that will have headings & Buttons.
- (22) Badges → Go really well with the List Groups and if we have a list of categories & when we want to

Put a number of Posts in that category.

JavaScript widgets

(1) Carousel Slider

where we have indicators, we have Arrows and set options like we can go faster (or) slower things like that, we can actually control that with JQuery.

(2) Collapsible Accordion

If we want a basic button to make another piece of content collapse (or) slide down.

(3) Modals

They are like Pop-ups which can fade in nicely, and we can put wherever we want them like Images, forms, text etc.

(4) ToolTips

If we wanna hover over something and have a tooltip

Showup.

(5) Popovers

(6) Scrollspy

It helps us to scroll through our content and have that piece of content, highlight that area in the Nav menu.

(7) Also we are going to implement custom, smooth Scrolling in our themes.

Extensions

(1) Bootstrap

(2) Live Server

By default, Bootstrap adds a bunch of styling.

Bootstrap can be implemented in our project by importing Bootstrap's CSS link from Bootstrap.com.

Dev tools → F12

If font-size: 2.5 rem



Indicates that they are multipliers.

By default, the core html tag default size is 16px.

So one rem will be 16px.

em

It is also similar to rem i.e., it goes by the size of parent element like if the size of parent element = 16px then the child "em" would be 16px for 1em.

Heading

h1 up to h6

DISPLAY classes They are helpful to make headings more prominent.

They are useful for Showcase Areas & Blocks.

`<h1 class="display-1"> Display 1</h1>`

↳ Increases the font size of the heading.

Otherwise, we can include upto "display-4"

Paragraphs

<P> </P>

we can include class = "lead" to make the Paragraph more prominent because it will just increase the font-size^(1.25 rem)

font-weight (300)

By default, Paragraphs have the default Margin-bottom: 1rem

Morespace Paragraph

<P class = "text-monospace"> My monospace paragraph </P>

<!-- style classes -->

<P class = "font-weight-bold"> Bold Text </P>

<P class = "font-weight-normal"> Normal Text </P>

<P class = "font-weight-italic"> Italic Text </P>

<P class = "font-style-italic"> Oblique Text </P>

Text Transforms

<P class = "text-lowercase"> MAKE LOWERCASE </P>

<P class = "text-uppercase"> MAKE UPPERCASE </P>

<P class = "text-capitalized"> make capitalized </P>

<P class = "text-sentence-case"> Makes every first letter as "uppercase".

Blockquotes

<blockquote class = "blockquote"> <P> This is a blockquote </P>

<P> This is a blockquote </P>

</blockquote>

The above command increases the font size and font weight.

blockquote Right Aligned

```
<blockquote class="blockquote text-right">  
    <p> This is a blockquote aligned right </p>  
</blockquote>
```

blockquote with footer

```
<blockquote class="blockquote text-right">  
    <p> </p>  
    <footer class="blockquote-footer"> Someone famous  
        <cite title="SourceTitle"> Source Title </cite>  
    </footer>  
</blockquote>
```

Text Truncate

```
<div class="text-truncate">  
</div>  
→ truncates the text into one line.
```

By default, lists will have some padding & bullet points.

- List 1
- List 2
- List 3
- List 4

To remove that, we need to follow the command as follows:

```
<ul class="list-unstyled">  
</ul>
```

Inline list → Makes the list-items to align horizontally.

```
<ul class="list-inline">  
  <li class="list-inline-item"> Lorem ipsum </li>  
  "  
  "  
</ul>  
</div>
```

To justify the para i.e., to align it evenly on both sides.

→ class = "text-justify"

Alignment

<p class="text-left"> Text aligned left </p>

<p class="text-center"> Text aligned center </p>

<p class="text-right"> Text aligned right </p>

Responsive Alignment

Sometimes we may want to align based on the screen size.

In Bootstrap, we have four different screen sizes.

- Small - sm
- Medium - md
- Large - lg
- Extra-large - xl

Responsive Right Alignment

<p class="text-sm-right"> Right aligned on small (or) larger </p>

<p class="text-md-right"> Right aligned on medium (or) larger </p>

< p class = "text-lg-right" > Right aligned on large (or) longer </p>.

< p class = "text-xl-right" > Right aligned on xl (or) longer </p>.

Similar things can be done for both left-align & center.

vertical Alignment

< span class = "align-baseline" > baseline

< span class = "align-top" > top

< span class = "align-bottom" > bottom

< span class = "align-middle" > middle

< span class = "align-text-top" > text-top

< span class = "align-text-bottom" > text-bottom

< span class = "align-text-middle" > text-middle

→ Block level elements are the ones which are div's, Paragraphs,

headings ↓.

Means it takes up the whole line area.

h1 element

→ span is an inline element by default.

Turn Block to Inline

< h1 class = "d-inline bg-success" > Hello </h1>

< h1 class = "d-inline bg-success" > Goodbye </h1>

↓.

will convert Block to inline and we look at dev tools, it will be like

• d inline

{ display: inline !important ;
y.

Turn Inline to Block

`` Block
converts Inline to Block.

Inline Block

`<div class="d-inline-block bg-warning">`

`<h3>Hello</h3>`

This is inline

`</div>`

`<div class="bg-warning">`

`<h3>Hello</h3>`

This is inline

`</div>`

`</div>`

Input

Hello

This is inline.

Hello

This is inline.

Hello

This is inline.

Hello

This is inline.

Floats & Fixed Positioning

```
<div class="float-left"> float left </div>  
<br>  
<div class="float-right"> float right </div>  
<br>  
<div class="float-none"> Float None </div>
```

Responsive Floating

```
<div class="float-sm-right"> float right on small (or) wider </div>
```

```
<br> float right on medium (or)  
<div class="float-md-right"> float right on medium (or) wider </div>  
<br> float right on large (or)  
<div class="float-lg-right"> float right on large (or) wider </div>
```

```
<br> float right on extra large (or)  
<div class="float-xl-right"> float right on extra large (or) wider </div>  
<br>
```

If we want to float responsively to the left side, then
we have to specify `class="float-sm|md|lg|xl-left"`

similarly, if we want to float none then:

```
<div class="float-sm-none"> </div>
```

★ clearfix.

```
<div> class = "bg-success" >
```

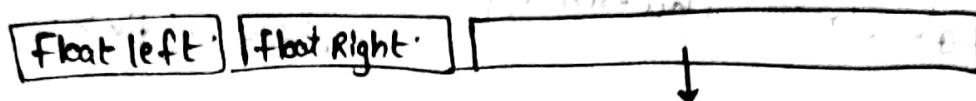
```
  <button> float left </button>
```

```
  <button> float Right </button>
```

```
</div>
```

```
<h3> fixed Top </h3>
```

Now on the screen it will be like.

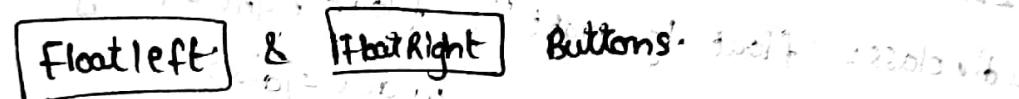


fixed Top

Green color background.

If we add a class to the Buttons like 'float-right' &

"float-left". Then, the fixed Top will go between the



O/p:

float left fixed Top

float right

But actually it's not the correct display.

for this we can add clearfix to the Parent.

```
<div class = "bg-success clearfix">
```

```
  <button class = "float-left"> float left </button>
```

```
  <button class = "float-right"> float right </button>
```

```
</div>
```

Now the O/p will be.

float left

float Right



bg color (green)

and the "fixed Top" text will be printed on the next line.

Fixed Positioning

Sometimes, we want something to the top and kinda stay there. for such purpose, we have to add the class of "fixed-top"

`<h3 class="fixed-top">fixed Top </h3>`

sticky Positioning

`<h3 class="sticky-top">sticky </h3>`

suppose the text "Sticky" is in page 2 then as soon as we arrive at page 2 then the "text" remains fixed at the position.

we can also fix things to the bottom

for this, we have to add the class "fixed-bottom"

`<h3 class="fixed-bottom"> Fixed Bottom </h3>`

Anything we want the user to always see on the screen, like the "Contact Page", "Email Subscription" & the "focus points".

Color & Background

There are certain classes which we can use on any element to change the color of text & background.

Text colors:

<p class = "text-primary"> Text Primary color </p>

<p class = "text-secondary"> Text Secondary color </p>

<p class = "text-success"> Text Success color </p>

↓
This command will change the text to green and it will be much useful like "when the login is successful".

for errors, we will represent the message in "Red color" and for that, we use

<p class = "text-danger"> Text Danger Color </p>

↓
Displays the text in "Red" color.

If we want half the darkness of pure black, then

<p class = "text-black-50"> Text Black 50 </p>

In dev tools, if we analyze, we can see that.

```
{ color: rgba(0,0,0,0.5) !important ;
```

}

same can be done with white-color,

<p class="text-white"> Text white </p>

Link colors

using Bootstrap, we can also change the color of the links like

```
<p>
  <a href="#" class="text-success"> Success link </a>
</p>
```

By default, the link colors will be "blue".

Background Colors

<div class="bg-primary text-white"> Background Primary
color </div>

bg-success —> changes the background to green.

we also have a class for Background Transparent:

i.e., <div class="bg-transparent"> Background
Transparent </div>

Invisible

```
<p class="invisible"> Hello </p>
```

If we analyze using chrome tools, then

we have

```
.invisible
{ visibility: hidden !important;
```

y

There is a difference between display: none and

visibility: hidden

↓
will be present but
can't see it.

↓
It takes out of DOM.

Margin, Padding, spacing

Bootstrap has really nice Margin & Padding classes in order to add spacing. Because we don't really want to add spacing.

- (i) Add line breaks.
- (ii) use Markups to add Spacing.
- (iii) Go to custom CSS and add custom Margin & Padding classes.

MB MARGIN BOTTOM

`<h1 class = "bg-success" > Margin Bottom 0 </h1>`

`<h1 class = "bg-success" > Margin Bottom 1 </h1>`

O/P:

MARGIN BOTTOM 0

MARGIN BOTTOM 1

when we add the class "mb-0" to the h1 class then when we save and execute it then the o/p will be.

MARGIN BOTTOM 0

MARGIN BOTTOM 1

Now, the space has been removed

Margin

→ It is the Spacing outside of the element

Spacing

Padding

→ inside of the element

Suppose there is a command like,

<h1 class = "mb-2 bg-success"> Margin Bottom 2 </h1>
↓
default Margin

and mb-3 → Increases the Margin, but by the element
mb-4 → Much Increases the Margin down.
mb-5 → Much More Increases the Margin.

mt - for classes that set the Margin-top (or) Padding-Top

ml - for classes that set the Margin-left (or) Padding-left

mr - for classes that set margin-right (or) padding-right

mx - for classes that set both *-left and *-right

my - for classes that set both *-top and *-bottom

m → Represents 'blank' for classes that set a margin (or) padding on all 4 sides of the element

mt →

<h1 class = "mt-0 bg-success"> Margin Top 0 </h1>

<h1 class = "mt-3 bg-success"> Margin Top 3 </h1>

Then:

Margin Top 0

Space

w)

Margin Top 3

Pushed downwards

ML & MR -->

```
<p>
<span class="mr-5 bg-success">Margin Right 5 </span>
    <span class="text-decoration-line-through">Margin Right 5</span>

```

is the King.
 Margin left 3
 Margin left 3
is the Queen </p>

op: Mr. Luther thought one

Margin Rights ↑ is the King ↓ Margin left 3 is the
Queen (M1-3) Makes Margin left 3 pushed by 3 Spaces.

M X MARGIN LEFT RIGHT. (HORIZONTAL)

<P> Gorish is
 Bad Guy
 Bad Guy
He is the human <P>.

0/p: $m\chi^{-5}$ $m\chi^{-5}$ BadGuy 1 He is the Human
Harrish is 1

MY MARGIN TOP BOTTOM (VERTICAL)

<P> Hari <|P>

`<p class="my-5 bg-success">Margin Top & Bottom </p>`

$\langle p \rangle$ Scunge $\langle |p| \rangle$

o/p

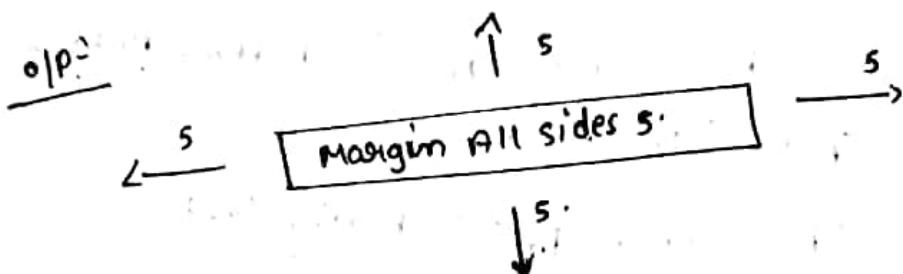
Hari

Margin Top & Bottom

Sangsu

<! -- BLANK - ALL SIDES -->

<p class = "m-5 bg-success" > Margin All sides 5 </p>



Padding

Space inside the Block.

Parent

Inside Space

<! -- PB PADDING BOTTOM -->

<h1 class = "pb-0 bg-warning" > Padding Bottom 0 </h1>

Now the output will be:

padding Bottom

Space will be added.

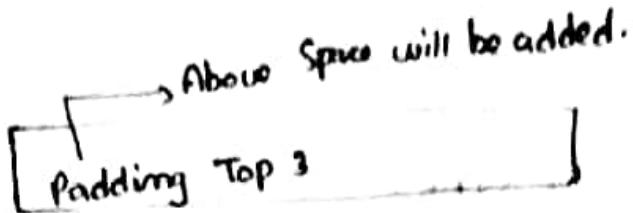
Likewise, pb-0 to pb-5

<! -- pr PADDING TOP -->

<hi class="ft-0 bg-warning"> Padding Top 0 </hi>

for class: "ft-3"

the op is



<!-- pl & PR -->

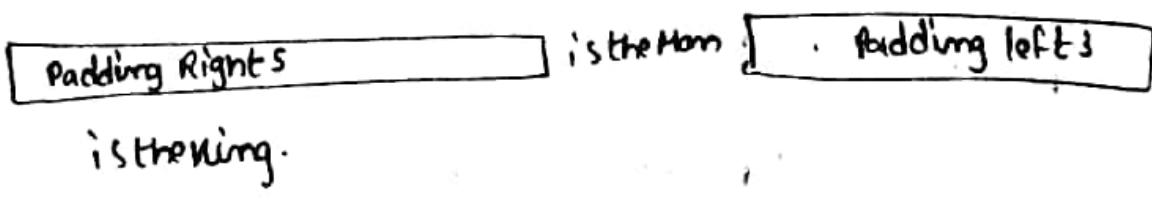
<p> Padding Right 5

 is the Man.

 Padding left 3

 is the King.

op:



<!-- px PADDING LEFT RIGHT -->

<p> Marish is

 padding left &

Right 5 is the Queen </p>

op:

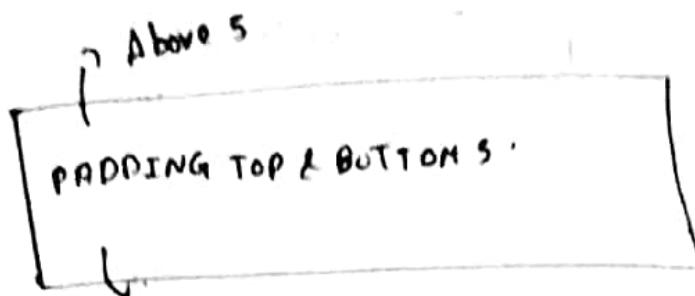
Marish is padding 12r5 is the Queen

If we want to give Padding to Top & Bottom then,

<! -- py padding top bottom -->

<p class = "py-5 bg-warning" > Padding Top & Bottom 5 </p>

o/p:



Now the o/p will be .

<! -- All sides -->

<p class = "p-3 bg-warning" > Padding All sides 5 </p>

Now the o/p will be .

PADDING All sides 5

<! -- Horizontal centering -->

<div class = "bg-info mx-auto" style = "width: 200px;" >

Horizontal centering

</div>

o/p:

HORIZONTAL CENTERING

It's just like setting

margin-left & margin-right as "0"

Sizing & Border classes

width classes

<div class="bg-success p-3 w-25"> width 25% </div>

↓
Means 25% of the container width.

I/P.

width 100%.

O/P

width 25%.

Likewise,

w-50

w-75

w-100

w-auto. → stays the same like w-100.

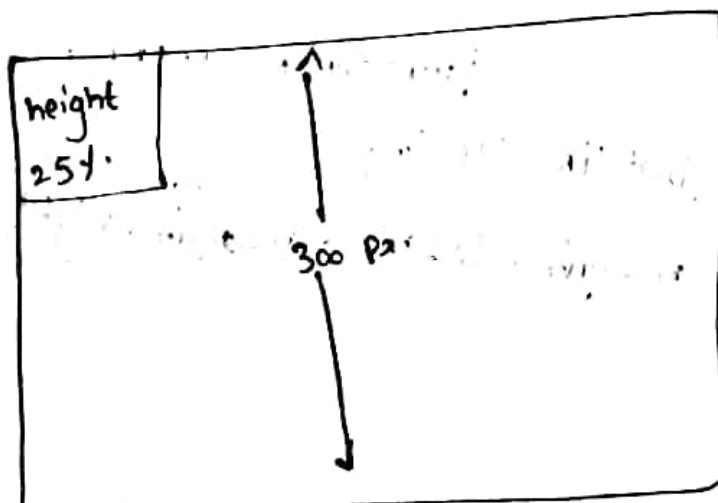
Height classes

<div style="height: 300px; border: 1px solid #333;">

<div class="bg-primary d-inline-block h-25">

Height 25% </div>

</div>



Likewise

`h=50, h=75, h=100, h=auto.`

`<!-- BORDERS -->`

`<div class="p-3 mb-2 bg-light border"> Regular </div>`

`<div class="p-3 mb-2 bg-light border-top"> Border Top </div>`

Likewise, we can add borders → Bottom, Left, Right

By mentioning,

`border-bottom,`

`border-left`

`border-right`

`<!-- BORDER COLORS -->`

`<div class="p-3 mb-2 bg-light border border-primary"> Primary </div>`

`<div class="p-3 mb-2 bg-light border border-secondary"> Secondary </div>`

Likewise, we have many border colors.

BORDER RADIUS

`<div class="p-3 mb-2 bg-light border rounded border-success"> ROUNDED </div>`

`<div class="p-3 mb-2 bg-light border rounded-top p"> ROUNDED TOP </div>`

Likewise

`rounded-bottom,`

`rounded-right`

`rounded-left`

Rounded circle

```
<div class="p-3 mb-2 bg-light w-25 border rounded-circle">  
    Rounded circle </div>
```



likewise, if something is rounded by default and we want to take away them,

```
<div class="p-3 mb-2 bg-light border rounded-0">Dark </div>
```

CSS BREAKPOINTS

when we want to create Projects and use Bootstrap, we may want to add our own Media Queries because we may want to have custom CSS that only happens at certain screen widths (or) certain devices.

we are going to look at different Media Queries for different sizes.

we have to write Media Queries at `<style>` tag.

```
<style>  
    @media (min-width: 576px) {
```

```
        body {
```

```
            background: red;
```

```
        }
```

```
    }
```

12.11.2023

/* Medium devices (tablets) */

@media (min-width: 768px)

{ body {

background: blue;

y

y

/* Large devices (desktops) */

@media (min-width: 992px)

{ body {

background: green;

y

y

/* Extra large devices (large desktops) */

@media (min-width: 1200px)

{ body {

background: yellow;

y

y

In addition to min-width, we can also use max-width

/* The given screen size is smaller +/

/* Extra small devices (portrait phones) */

@media (max-width: 575px) {

body { background: coral;

y

y

/* Small devices (landscape phones) */

@media (max-width : 767px)

}

y

/* Medium devices (tablets) */

@media (max-width : 991px) ?

y

/* Large devices (desktops) */

@media (max-width : 1191px) ?

y

Section 3

Components

Button & Button Groups.

Buttons

<button class = "btn" type = "button" > Primary </button>

↓
This is the core button class.

when we inspect "Primary" using dev-tools, it adds a lot of stuff so "btn" is important for any button for the purpose of "styling".

syntax to add color to the button:

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

<button class = "btn btn-secondary" type = "button" >
Secondary </button>

↳ Adds certain color

to the button.

If we want to make a button to appear as a link then

the syntax is as follows:

<button class = "btn btn-link" type = "button" >
Link </button>

↳ Link.

We can use button classes on different types of elements.

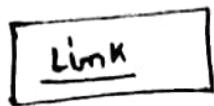
If we want to format the link in the form of button,

then

<!-- BUTTON TAGS / TYPES -->

Link

or:



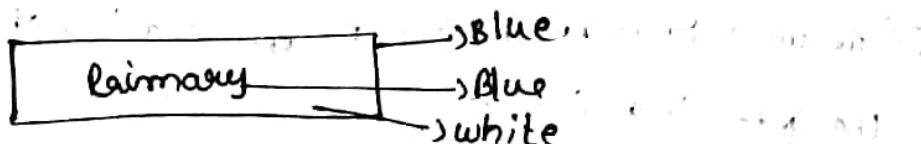
<button class="btn btn-success" type="submit">Button </button>

<input class="btn btn-light" type="button" value="Input">
<input class="btn btn-warning" type="submit" value="Submit">
<input class="btn btn-info" type="reset" value="Reset">

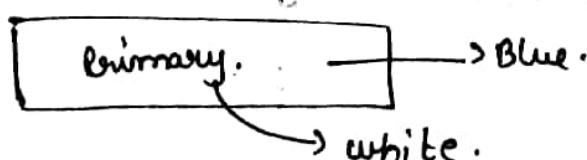
<!-- OUTLINE BUTTONS -->

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

It means when we hover over the button, it will look like normal primary button otherwise,



when hovered over, then



Likewise, there are OUTLINE BUTTON CLASSES for

- (1) btn-outline-secondary
- (2) btn-outline-success
- (3) btn-outline-info
- (4) btn-outline-warning
- (5) btn-outline-danger
- (6) btn-outline-light
- (7) btn-outline-dark

Button sizes; class = "btn btn-primary btn-lg"

{ button type = "button" >
 < i class = "fas fa-user" > + /i > Large button </button>
 < button type = "button" > Small button </button>
This will include
an icon beside the
button and also
makes the button
to appear large.

Block level Button

To make our button go all the way across.

<button class = "btn btn-dark btn-block" type = "button" >
 Block level button </button>

O/P:

Block level button

we also have different states

<button class="btn btn-primary" type="button">
Regular Button </button>

<button class="btn btn-primary active" type="button">
Active Button </button>

This makes the button look some what darker.

<button class="btn btn-primary disabled" type="button">
>Disabled Button </button>

This will make the button to look lighter and there won't be any cursor icon marker when hovered over it.

<button class="btn btn-primary" type="button"
data-toggle="button">

Toggle State

</button>

O/P:

Toggle State

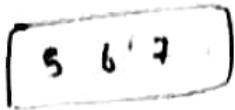
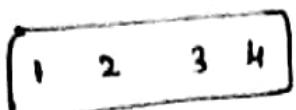
→
when we hover over it and click it then it will become darker and then it become lighter.

```
<!-- BUTTON GROUPS -->  
<div class="btn-group">  
  <button class="btn btn-primary"  
    type="button">Left </button>  
  <button class="btn btn-secondary"  
    type="button">Middle </button>  
  <button class="btn btn-primary"  
    type="button">Right </button>  
</div>
```

Button ToolBar.

```
<div class="btn-toolbar">  
  <div class="btn-group mr-2">  
    <button class="btn btn-primary"  
      type="button">1 </button>  
    <br/> * 4  
  </div>  
  <div class="btn-group mr-2">  
    <button class="btn btn-primary"  
      type="button">5 </button>  
    <br/> * 1  
  </div>  
  <div class="btn-group">  
    <button class="btn btn-primary"  
      type="button">6 </button>  
  </div>  
</div>
```

Op:



for vertical Variant

```
<div class="btn-group-vertical">
```

```
</div>
```

Op:



Button Dropdown

In order to make a button to show list of items.

```
<div class="dropdown">
```

```
  <button class="btn btn-primary dropdown-toggle"
```

```
    type="button" data-toggle="dropdown">
```

My Dropdown

```
</button>
```

```
<div class="dropdown-menu">
```

```
  <a class="dropdown-item" href="#"> Link One
```

```
</a>
```

```
  <a class="dropdown-item" href="#"> Link Two
```

```
</a>
```

```
  <a class="dropdown-item" href="#"> Link Three
```

```
</a>
```

```
</div>
```

< |div>

o/p:

My Dropdown ▾

Link One

Link Two

Link Three.

we can also implement SPLIT DROP Downs.

```
<div class = "btn-group">
  <button class = "btn btn-primary" type = "button">
    My Button </button>
  <button class = "btn btn-secondary dropdown-toggle"
    type = "button" data-toggle = "dropdown">
    <span> Toggle Dropdown </span>
  </button>

  <div class = "dropdown-menu">
    <a class = "dropdown-item" href = "#">
      Link One </a>
    <a class = "dropdown-item" href = "#">
      Link Two </a>
    <a class = "dropdown-item" href = "#">
      Link Three </a>
  </div>
</div>
```

o/p:

My Button ▾ Toggle Dropdown ▾

Link One

Link Two

Link Three.

If we wish to put a divider then,

<div class="dropdown-divider"></div>

Must be placed between two links in the dropdown.

NAVbar & NAVs

- A container will basically move everything to the middle.
It means there will be spacing on the left and right side.
It means there will be spacing on the left and right side.
It means there will be spacing on the left and right side.
- A container basically adds margin: auto and gives it max-width.
By default it will be around 1140px.
- To keep any content inside the containers the following syntax needs to be followed.

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

SIMPLE NAVBAR

```
<nav>  
    <div>  
        <a href="#">Navbar </a>  
        <ul>  
            <li>  
                <a href="#">Home </a>  
            </li>  
            <li>  
                <a href="#">About </a>  
            </li>
```

```

<li>
    <a href="#"> Ab Services </a>
</li>

<li>
    <a href="#"> Contact </a>
</li>

</ul>

</div>

</nav>

```

Now By adding Bootstrap classes

lets where the Responsiveness will kick in.

```

<nav class="navbar navbar-expand-sm navbar-light bg-light">

```

```

<div class="container">
    <a class="navbar-brand" href="#">Navbar </a>
    <ul class="navbar-nav">
        <li class="nav-item">
            <a class="nav-link" href="#">Home </a>
        <li>
            <ul>
                <li> and the name here
                    <li> about
                    <li> services
                    <li> contact
            </ul>
        </li>
    </ul>

```

The code for above gives the o/p:

Navbar	Home About Services Contact
--------	-----------------------------

when we expand the screen the items also gets expanded but remain within the container.

we can add a Hamburger menu (or) collapse menu.

new class : "navbar navbar-expand-sm navbar-light bg-light",

<div class="container">

 Navbar

<button class="navbar-toggler" data-toggle="collapse" data-target="#1h3"> </button>

↓
Now we have to take this id and put it on the div around

the "ul":

<div id="#1h3" class="collapse navbar-collapse">

<ul class="navbar-nav">

<li class="nav-item">

 HOME

→ 3 and the text are

About

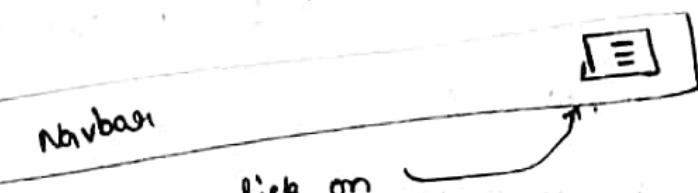
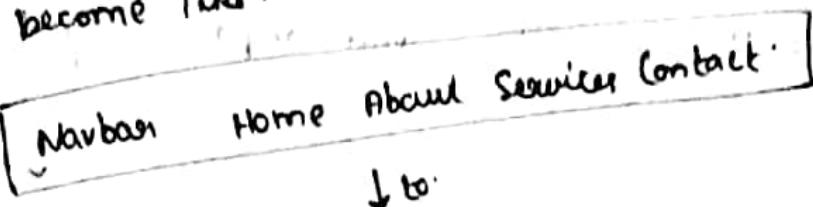
Services

Contact

</div>

</div>

O/P:
once the screen becomes small, the navigation bar will become like:



then the dropdown of four items like

HOME

About

Services

Contact

will be displayed.

In order to align
the form to the
right.

NAVBAR - FORM

complete previous code + under the last, .

<form class="form-inline ml-auto">

<input type="text" class="form-control mr-2" placeholder="Search">

<button class="btn btn-outline-success">

Search </button>

</form>

2! -- NAVBAR WITH DROPODOWN-->

```
<li class="nav-item dropdown">
```

```
<a href="#" class="new-link dropdown-toggle">before
```

۲۱۸

data-toggle = "dropdown"

bottom to dropdown, To add the menu items,

```
<div class="dropdown-menu">
```

```
class="dropdown-item">Link1</a>  
a href="#" class="dropdown-item">Link2
```

7

11

<|div>

2 | 8:17

2.1 dropdown
we have to include the scripts of Bootstrap, JQuery and
JavaScript at the bottom in order to make the Dropdown
menu to work properly.

NAVBAR COLORS

NAVBAR colors: `<nav class="navbar navbar-expand-sm navbar-dark bg-dark">`

```
<nav class="navbar navbar-expand-sm bg-dark">
```

०१८:

O/P:

lighter teal

(Navbar)

Home About Services Contact

dark.

If we add bg-blur, we might need our Navbar to look like last
like below

for that, we have to keep navbar-dark

<!-- fixed-top -->

<nav class="navbar fixed-top navbar-dark bg-dark">

Navbar

fixed

Top

</nav>

O/P:

Navbar Fixed Top

Navbar will be sticked to Top without any movement even if
we are scrolling

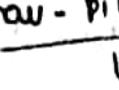
for fixing bottom

simply add the class "fixed-bottom" instead of "fixed-top".

sticky Top

once we move through the content of "sticky-top"
it will remain fixed in that position and doesn't
move.

L! -- NAVS -->

ul class: "nav nav-pills" 
, put everything inline

} <li class="nav-item">
 Link1
 , class="nav-link"

* 3

O/P:

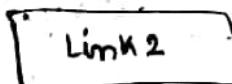
Link1 Link2 Link3 Link4

To keep a link as active
we have to keep "active" in

 Link2

Then

O/P:



If we want to put the links in the middle, then we have to
add the class "justify-content-center".

O/P:

Link1 Link2 Link3 Link4

If we want to Right align the things then, Put
justify-right x

justify-content-end

for vertical alignment of links.

we have to put the class "flex-column"

o/p:

Link 1

Link 2

Link 3

Link 4

for fill & justify.

we can put the class = "nav-fill" in the ul.

Syntax

<ul class="nav nav-fill nav-pills">

o/p:

Link 1

Link 2

Link 3

Link 4.

It means the links are spreaded all the way across.

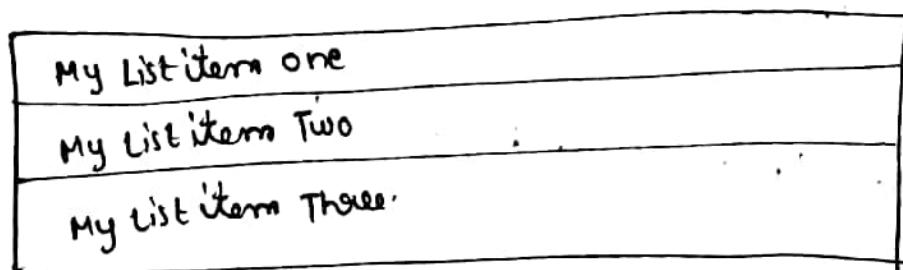
List Groups & Badges

(1) List Groups are basically like fancy "unordered list", we can use this for menu (or) just for listing out contents

Syntax

```
<ul class="list-group">  
    <li class="list-group-item"> My List Item One </li>  
        " " " Two "  
        "  
        "  
</ul>
```

Output



So,

<!-- LIST GROUP WITH LINKS -->

```
<ul class="list-group mnb-5">
```

```
    <a class="list-group-item" href="#"> My List One  
    </a>
```

```
    <a class="list-group-item" href="#"> My List
```

Item Two

"

```
    </a>
```

```
</ul>
```

Q/P:

My List One
My List Item Two.

<!-- contextual classes -->

<ul class="list-group">

<li class="list-group-item"> Regular List Item

<li class="list-group-item list-group-item-binary">
 > Binary List Item

Similarly we have

list-group-item - secondary

" " - success

" " - info

" " - warning

" " - danger

" " - light

" " - dark

Q/P:

This gives color to the list-group-items

<!-- FLUSH LIST GROUP -->

```
<ul class="list-group list-group-flush mb-5">  
  <li class="list-group-item"> My list Item one </li>  
  " " "  
  " " "  
  </li>  
</ul>
```

o/p:

My list Item one.

My list Item Two

Badges.

They are another things we can use this lists and lists items.
May be suppose if we have a Post categories ,Blog Posts and
we want to show that how many posts we have in each
category.

```
<span class="badge badge-primary">30</span>
```

```
<ul class="list-group">
```

```
  <li class="list-group-item"> My list Item one </li>
```

d-flex justify-content-between align-items-center

o/p:

My list Item one.

30

Breadcrumb

- They are quite useful for inner Pages if we want to show kind of maps of our links of our Pages.
- Breadcrumbs are made of ordered List

Syntax and understanding.

<ol class = "breadcrumb">

<li class = "breadcrumb-item active"> Home

<ol class = "breadcrumb">

<li class = "breadcrumb-item">

 Home

 class = "breadcrumb-item active"

 Users

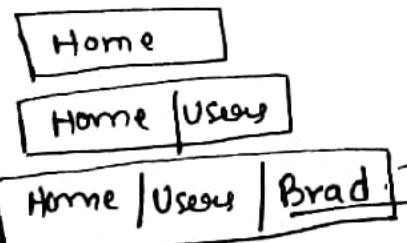
<ol class = "breadcrumb">

<li class = "breadcrumb-item">

 Users

<li class = "breadcrumb-item active"> Brad

O/P:



It's lighter and Home, Users will be blue.
since active

Forms & Input

<form>

<div class="form-group">

<label for="name"> Name </label>

<input class="form-control" type="text"

id="name" placeholder="Enter name">

</div>

Output:

Name:

Enter name

<div class="form-group">

<label for="email"> Email address </label>

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

id="email" placeholder="Enter email">

</div>

Here one thing to note is that we are keeping "form-group" class in the "div" and "form-control" class in the input. So to increase the font-size of the "input" add form-control-lg beside "form-control".

Suppose if we want it to be smaller, then we have to write "form-control-sm"

To give a message like "Your email will not be shared" underneath our email we have to simply mention that

<small class = "form-text text-muted"> Your email will not ever be shared </small>

Note:

If we mention "readonly" in the input field then we cannot be able to type anything just we can be able to read it.

o/p:

password

password

can't be typed anything.

Now the form-group can be applied to select class too

<div class = "form-group">

<label for = "gender"> Gender </label>

<select class = "form-control" id = "gender">

<option> Male </option>

<option> Female </option>

</select>

</div>

o/p:

Gender

Male



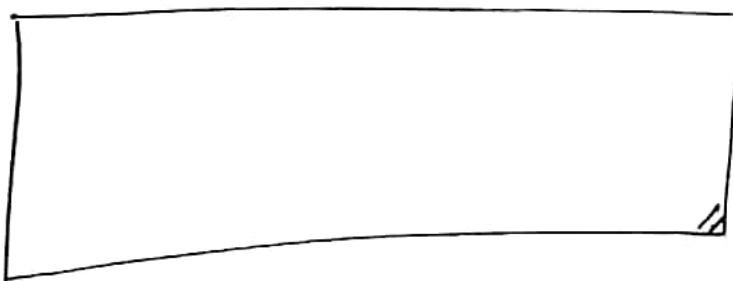
Something happens with the text area

```
<div class="form-group">  
  <label for="message"> Message </label>  
  <textarea class="form-control" id="message" rows="3"></textarea>
```

</div>

o/p

Message



Now we have file input too

```
<div class="form-group">  
  <label for="file"> File Input </label>  
  <input type="file" id="file" class="form-control-file" />  
  <small class="form-text text-muted" id="fileHelp"> Max 3mb size </small>  
</div>
```

</div>

o/p:

file input

[choose file] No file chosen.

Max 3mb size

custom file Input.

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



we also have a "RANGE" input.

```
<div class="form-group">  
  <label for="membership"> Membership Level </label>  
  <input class="custom-range" type="range">  
</div>
```

membership Level.

Now, the input class of type "range" has attributes

If suppose we gave min="1" max="5" step="1"

value="1" then the o/p will look like the

following.

Membership Level.

when we click on the above line the point will be moved forward with a step size of 1.

every form may generally have a "submit" button so,
to use a submit button the following syntax will be
applicable:

```
<button class="btn btn-primary btn-block" type="submit">  
    Submit </button>
```

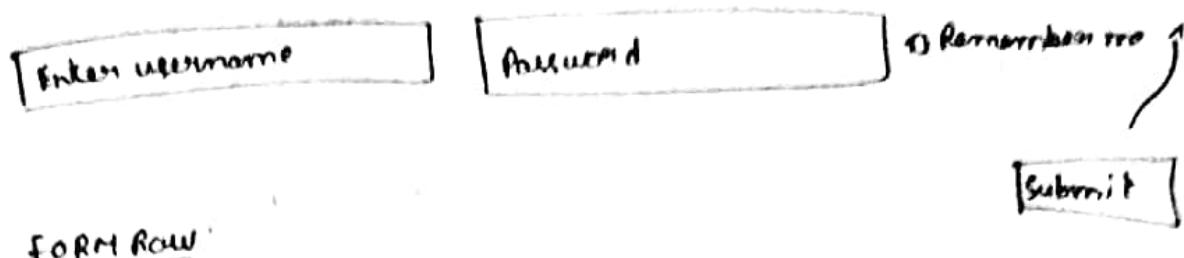
O/P:

Submit

we can also have inline forms for the purpose of "login" stuff
like that.

```
<form class="form-inline">  
    <input class="form-control mr-2" type="text"  
        id="username" placeholder="Enter username">  
    <input class="form-control mr-2" type="text"  
        id="password" placeholder="Password">  
    <div class="form-check">  
        <label class="form-check-label mr-2">  
            <input type="checkbox"  
                class="form-check-input"> Remember me  
        </label>  
    </div>  
    <button class="btn btn-light"  
        type="submit">Submit </button>  
</form>
```

O/P:



Form Row:

```
<form>
  <div class="form-row">
    <div class="col">
      <input type="text" class="form-control"
            placeholder="First name">
    </div>
    <div class="col">
      <input type="text" class="form-control"
            placeholder="Last name">
    </div>
  </div>
</form>
```

</form>

O/P:



validation:

we can do form validation using "frontend Javascript"
and the backend too.

Bootstrap provides us classes to show the red border
when the input is wrong

around the
form.

```
<div class="form-group">
  <label for="username"> Username </label>
  <input class="form-control" type="text"
        id="username" /> is-valid
</div>
```

↓
It gives us a green box.

```
<div class="form-group">
  <label for="password"> Password </label>
  <input class="form-control" is-invalid type="text"
        id="password" />
  <div class="invalid-feedback">
    Password not strong enough
  </div>
</div>
```

This gives a red box.

```
<div class="form-group">
  <label for="password2"> Confirm Password </label>
  <input class="form-control" is-invalid type="text"
        id="password2" />
  <div class="invalid-feedback">
    Password does not match
  </div>
</div>
```

OP:

Username

→ It will be in green color.

Password

Password not strong enough

Both will be displayed
as "red-color"
Since it is
is-invalid.

confirm Password

Password does not match.

Both will be red.

Input Groups & Addons

Just we have form Group class, we also InputGroup one which will group together the fields and whatever else we want whether it's a symbol (or) text (or) anything like that.

Basic Input Groups

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

o/p:

@	Username
---	----------

Now Below we are doing the same as above but, we are attending something.

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

o/p:

username	@Something.com
----------	----------------

for custom-URL

```
<label for="basic-url">custom URL </label>
<div class="input-group mb-3">
  <div class="input-group-prepend">
    <span
      class="input-group-text">https://something.com/
```

o/p:

custom URL

https://something.com/member/

we can apply both Bootstrap & append at the same time,

Ex:

```
<div class="input-group mb-3">  
  <div class="input-group-prepend">  
    <span class="input-group-text">$</span>  
  </div>  
  <div class="input-group-append">  
    <span class="input-group-text">.00</span>  
  </div>  
</div>
```

O/P:

\$.00
----	--	-----

we can also prepend "checkboxes" & "radio boxes".

```
<div class="input-group mb-3">  
  <div><div class="input-group-prepend">  
    <div><div class="input-group-text">  
      <input type="checkbox"/>  
    </div>  
  </div>  
  <input type="text" class="form-control"/>  
</div>
```

O/P:

<input type="checkbox"/>	
--------------------------	--

same can be done with radio too, simply replace the:

input type = "radio".

→ O/P:

<input type="radio"/>	
-----------------------	--

we can also append multiple inputs.

```
<div class="input-group">  
  <div class="input-group-append">  
    <span class="input-group-text" id="">  
      Name & Email </span>  
  
  </div>  
  <input class="form-control" type="text"  
    placeholder="full Name">  
  <input class="form-control" type="email"  
    placeholder="Email">  
  
</div>
```

O/P:

NAME & EMAIL	FULL NAME	EMAIL
--------------	-----------	-------

we can also do Button ADDONS

```
<div class="input-group mb-3">  
  <input class="form-control" type="text"  
    placeholder="find By Name ...">  
  <div class="input-group-append">  
    <button class="btn btn-outline-secondary"  
      type="button">Search </button>  
  
</div>  
</div>
```

O/P:

findBy Name	Search
-------------	--------

Alerts and Progress Bars

- Alerts are helpful when we are building applications with JavaScript framework like React.
- Alerts are used for dynamic purposes i.e., for the messages saying that we are logged in (or) logged out or

Blog Posts:

```
<div class = "alert alert-primary">  
    <strong> Primary </strong> Blog Post added.  
</div>
```

O/P:

Primary Blog Post added

Light Blue Background.

Likewise, we can use:

alert-secondary → leads to light grey background.

alert-success → Mostly Popular because it is for successful messages.

alert-danger → Mostly Popular because it is for error messages.

alert-info → Basically used for messages.

alert-warning → used for warnings and quite errors.

alert-light → very very light grey.

alert-dark →

These alerts stays up until the page reloads and if we are using some front end framework and when we want to do something away using DOM.

we can also make DISMISSABLE ALERTS i.e., we can click and make them go away without us having to do any JavaScript (or) anything.

<!-- DISMISSABLE ALERT -->

```
<div class="alert alert-success alert-dismissible">
  <button class="close" type="button"
    data-dismiss="alert">
    <span>×</span> </span>
  </button> → for getting "X" icon.
  <strong> DISMISSABLE </strong> Blog Post added
</div>
```

O/P:

DISMISABLE Blog Post added

Now when we click 'X' then the message will be disappeared.

Alerts for Extra content:-

```
<div class="alert alert-success">
  <h4 class="alert-heading">Congrats! </h4>
  <p> Harish is the winner </p>
  <br>
  <p class="mb-0"> Lorem ipsum </p>
</div>
```

O/P:

Congrats!

Hanish is the winner

Lorem ipsum

→ green color - alert-Success

Progress Bars

It means the amount of task (.) completed in the overall amount

Syntax & usage

```
<div class="Progress">  
  <div class="Progress-bar" style="width: 35%;"></div>  
  </div>  
  
<div class="Progress">  
  <div class="Progress-bar" style="width: 35%;">35%  
  </div>  
</div>
```

O/P

If we want to change the color of "Progress-bar" then,

```
<div class="Progress">  
  <div class="Progress-bar bg-info" style="width: 75%;>  
    </div>  
</div>
```

↓
Here we have to specify the color.

If we want to change the height, then we have to mention the class = "Progress" but not in the "Progress-bar" class.

Syntax

```
<div class="Progress" style="height:1px;">  
  <div class="Progress-bar bg-success" style="width:75%;">
```

</div>

</div>

o/p:

75%



Striped Bar

```
<div class="Progress">
```

```
  <div class="Progress-bar bg-danger">
```

```
    Progress-bar-striped" style="width:35%;"></div>
```

</div>

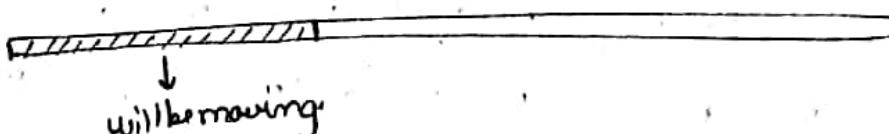
o/p:



Animated

In addition to "striped bar" Syntax we have to add
Progress-bar-animated beside Progress-bar-striped.
So that the line will be continuously moving.

o/p:



If we want to have multiple rows

<div class="progress">

<div></div>

<div></div>

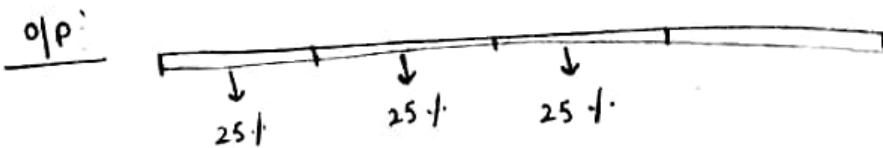
<div></div>

</div>

} In each of them, we will have
some width (or) different
width in the 'style' attribute

So, If everything contains
width: 25%.

then



Tables & Pagination

Here "Pagination" refers to UI but not to the actual Pagination

<table class="table">

<thead>

<tr>

<th> # </th>

<th> first Name </th>

<th> Last Name </th>

<th> Email </th>

</tr>

</thead>

<tbody>

<tr>

<th scope="row"> 1 </th>

<td> John </td>

<td> Doe </td>

<td> jdoe@gmail.com </td>

</tr>

</tbody>

</table>

<!-- Inverse / Dark TABLE -->

we have to add the class of "table-dark"

so that the table will become dark shaded.

<!-- TABLE HEAD INVERSE -->

we can also give inverse to only "table heading area".

Syntax:

<thead class = "thead-dark">.

we can also do a striped table.

Syntax

<table class = "table table-striped">.

what it will do is starting from 1st row, alternate row background will be greyed.

we can do a Bordered table.

<table class = "table table-bordered">.

o/p will be like

#	First Name	Last Name	Email
1	S	O	Hariishi
2			
3			

we can also give a borderless class

<table class = "table table-borderless">

we can make the table hoverable.

<table class = "table table-hover">

what will it do, it makes the background (gray color) to appear when we hover over the rows.

we can also add contextual classes.

<table class = "table">

basically we will specify the contextual classes for the table rows.

Syntax

<tr class = "table-success">

table-danger

table-warning.

and also to decrease the padding.

If we want to decrease our table size, then we can mention

table-sm

<table class = "table table-sm">

basically table-sm is used when we want to fit more entries.

* we can also mix the table properties.

<table class = "table table-sm table-hover table-striped">

<!-- RESPONSIVE TABLE -- scrollbars -->

<div class = "table-responsive">

{ <table class = "table">

} Make the table responsive according to the screensize.

Pagination UI

It is basically the thing with some NAVs, UL's and some URL's, I mean the Links

```
<nav>
  <ul class = "Pagination">
    <li class = "Page-item">
      <a class = "Page-link" href = "#"> Previous </a>
    </li>
    <li class = "Page-item">
      <a href = "#"> 1 </a>
    </li>
    <li class = "Page-item">
      <a href = "#"> 2 </a>
    </li>
    <li class = "Page-item">
      <a href = "#"> 3 </a>
    </li>
    <li class = "Page-item">
      <a href = "#"> Next </a>
    </li>
  </ul>
</nav>
```

Before every href we have to put the class "Page-link"

o/p:

Previous	1	2	3	Next
----------	---	---	---	------

In order to make Previous as "disabled" we have to mention "disabled" in the

```
<li class="page-item disabled">
```

PAGINATION CENTER

For this purpose, we have to write the class namely "justify-content-center" in the ul class: "Pagination".

Syntax

```
<ul class="Pagination justify-content-center">
```

```
<!-- PAGINATION RIGHT -->
```

```
<ul class="Pagination justify-content-end">
```

We can also add large pagination.

By putting "Pagination-lg".

Include "Pagination-lg" in the "ul class".

```
<ul class="Pagination Pagination-lg">
```

Similarly for Pagination small, add "Pagination-sm".

If we want to add arrows instead of Previous & Next?

In the Pagination UI code, Instead of previous add the following

```
<span> &ltquo; </span>
<span class = "sr-only">Previous </span> } o/p [x]
```

for right facing arrow:

```
<span> &gtquo; </span>
<span class = "sr-only">Next </span> } o/p [>]
```

Cards

```
<!-- Block. -->
```

```
<div class = "card">
<div>
    Lorem ipsum dolor sit amet, consectetur
    adipiscing elit. Selectus, voluptate.
</div>
```

```
</div>.
```

Basically the class = "card" gives border to the text.

o/p:

```
Lorem ipsum dolor sit amet consectetur
adipiscing elit. Selectus, voluptate.
```

Now if we add the class = "card-body" inside the div then,
we get the padding.

Syntax:-

```
<div class = "card">
    <div class = "card-body">
        Lorem ipsum dolor sit amet, consectetur
        adipiscing elit. Selectus, voluptate.
    </div>
</div>.
```

Simple card

```
<div class="card" style="width: 200mm;">
  <div class="card-body">
    <h4 class="card-title">Card Title </h4>
    <h6 class="card-subtitle text-muted">Card
      subtitle </h6>
      ↓
      Makes the text look
      lighter.
    <p class="card-text">Lorem ipsum dolor sit amet,
      consectetur adipiscing elit. Illud quae. </p>
    <a class="btn btn-outline-primary" href="#">
      Read More </a>
  </div>
</div>
```

Output:
 Card Title
 Card subtitle
 Content
Read More

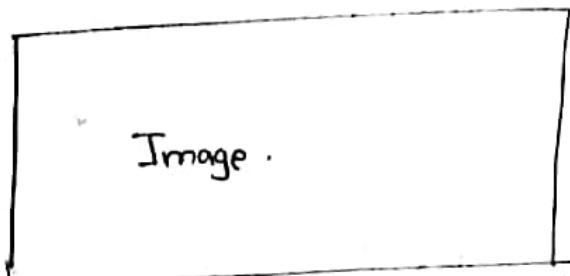
Card with Image

```
<div class="card">
  
  <div class="card-body">
    <h4 class="card-title">Card Title </h4>
    <p class="card-text">Lorem ipsum dolor sit
      amet, consectetur adipiscing
      elit. Illud quae. </p>
```

2 a class = "btn btn-success btn-block" href = "#" > Read More

</div>
</div>

O/P:



Card Title

lorem ipsum dolor sit amet

Read More.

<!-- CARD WITH HEADER -->

<div class = "card">
 <div class = "card-header">
 My Card
 </div>

<div class = "card-body">

 <h4 class = "card-title"> Card Title </h4>

 <p class = "card-text"> Lorem ipsum dolor sit amet,

 consectetur adipiscing elit. Aut , Exquisitiq </p>

class
= "btn
btn-danger"
 Read More
 </div>
</div>

QPR

My Card

Card Title

lorem ipsum dolor sit amet consectetur adipiscing elit. Aut,
permetatis.

Read More

<!-- HEADER, FOOTER, CENTERED -->

same in before code. Give text-center in the first div.

<div class="card text-center">

and then at last for footer,

<div class="card-footer text-muted">

2 Days Ago

</div>

QPR:

My card

Card Title

Loewm

Read More

2 Days Ago

2! -- CARD WITH NAV --> *for this, we may need Javascript*

<div class="card">

<div class="card-header">

<ul class="nav nav-tabs card-header-tabs">

<li class="nav-item">

}

 Active

>*2

Link

disabled

</div>

<div class="card-body">

<h4 class="card-title"> Card Title </h4>

<p class="card-text"> Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Aut, beneficiatis</p>

 Read More

</div>

</div>

: Op:

Active Link Disabled

|

Card Title

 Lorem ipsum dolor sit amet, consectetur adipiscing
 elit. Aut, beneficiatis.

Read More

Image overlay

when we want to add the content inside the image on its background
then we can do it with cards.

<! -- IMAGE OVERLAY --> Text - white makes the image fit within the card.
<div> class = "card" </div>
<div> class = "card-img" </div>
<img src = "https://source.unsplash.com/random/1000x300"
alt = "">
<div> class = "card-img-overlay" </div>
<div>
 <h4 class = "card-title"> Card title </h4>
 <p class = "card-text"> This is a wider card with supporting text below as a natural lead-in to additional content. This content is a little bit longer </p>
 <p class = "card-text">
 <small class = "text-muted"> Last updated 3 mins ago </small>
 </p>
</div>
</div>

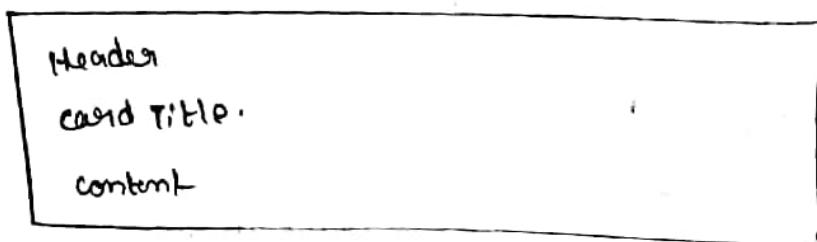
Output:



* !-- Background color -->
 > bg-primary text-white.

```
<div class="card">
    <div class="header"> Header </div>
    <div class="card-body">
        <h4 class="card-title"> Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. </h4>
        <p class="card-text"> Content </p>
    </div>
</div>
```

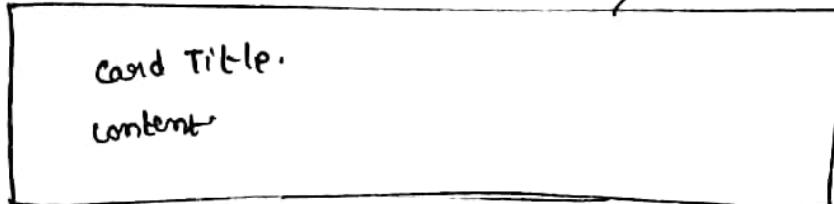
Olp:



Card outline

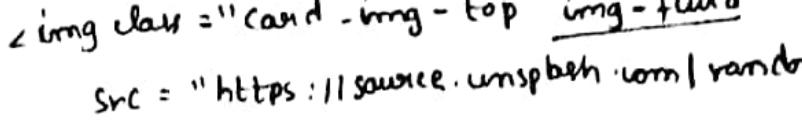
```
<div class="card border-primary mb-3">
    <div class="card-body">
        <h4 class="card-title"> Card Title </h4>
        <p class="card-text"> Content </p>
    </div>
</div>
```

Olp:



<! - CARD COLUMNS - >

<div class = "card-columns">
<div class = "card">

 
 <div class = "card-body">
 <h4 class = "card-title">Card title that
 wraps to a new line</h4>
 <p class = "card-text">This is a longer card
 with supporting text below as a natural lead-in
 to additional content. This content is a little bit
 longer.</p>

</div>

</div>

Next card is going to have blockquote:

<div class = "card">
 <blockquote>
 <p>Lorem ipsum dolor sit amet, consectetur
 adipiscing elit. Integer posuere erat a ante
 </p>

<footer>

<small>

Someone famous in

<cite title = "Source title"> Source
Title</cite>

</small>

</footer>

</blockquote>

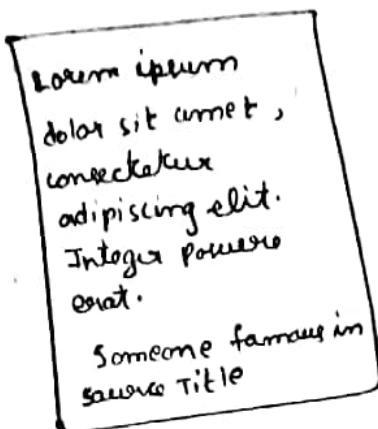
```
</div>
<div class="card">
    
    <div class="card-body">
        <h4 class="card-title">Card title </h4>
        <p class="card-text">This card has supporting text below as a natural lead-in to additional content </p>
        <p class="card-text">
            <small class="text-muted">Last updated 3 mins ago </small>
        </p>
    </div>
</div>
<div class="card card-dark bg-transparent p-3">
    <blockquote class="card-body-quote">
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer posuere erat. </p>
    <small>
        Someone famous in
        <small>Source Title</small> Source Title </small>
    </small>
    </blockquote>
    <small>
        <small>Source Title</small> Source Title </small>
    </small>

```

</div>

</div>

o/p:



<div class="card-text - content">

<div class="card-body" class="card-title">

<h1> Card title </h1>

<p> This card has supporting text below as a
class="card-text" natural lead-in to additional content. </p>

<p> class="card-text">

<small> Last updated 3min ago </small>

</p>

</div>

</div>

media objects

- They are like the "image" floated to a certain position and text and title.
- They are used for newsites, small blogs etc even like in a side bar.

<!-- MEDIA OBJECT --> Media-object turns the display to

<div class = "media"> flex;

<img class = "mr-3" mt-3

src = "https://source.unsplash.com/random/90x90" >

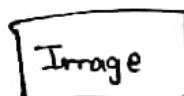
<div class = "media-body"> It will set the flex as "1".

<h5> Media heading </h5> media-body is the content part of component of element of media-object

</div>

</div>

o/p :-



Media heading

content

Nesting media objects

<div class = "media">

<img class = "mr-3"

src = "https://source.unsplash.com/random/90x91" >

<div class = "media-body">

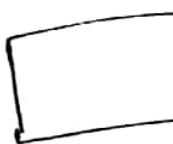
<h5> Media heading </h5>

content

<div> class = "media mt-4"


```
  
<a>  
<div class="media-body">  
<h3>Nested Media </h3>  
content  
</div>  
</div>
```

O/P:



media heading

content



Nested Media

content

we can also mess with the alignment

```
<div class="media">  

```

remember that we are using "media" class which turns into a flex container

```
<h3 align-self-start>Top aligned media</h3>
```

This will align the image to the top

```
<div class="media-body">
```

```
<h3> Top aligned media </h3>
```

```
<p> content
```

```
</p>
```

```
</div>
```

```
</div>
```

O/P:



Top aligned media
content

If we want to center align the Image.

```
<div class = "media">  
  <img class = "mr-3 align-self-center"  
       src = "https://source.unsplash.com/random/90x90",  
  <div class = "media-body">  
    <h5>center aligned media</h5>  
    <p>content</p>  
</div>  
</div>
```

Op:



center aligned media

content

Similarly for bottom,

we have to include the class: "align-self-end".

i.e., we can do this easily without manually aligning. Just with the help of Media object we can do it since they are flex containers.



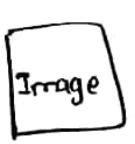
Bottom aligned media

content

we have the Media objects in the list too.

```
<!-- Media List -->  
<ul class="list-unstyled">  
  <li>  
      
    <div> class="media-object"  
      <h5> content body.  
    </div>  
  </li>  
> * 2.  
</ul>
```

o/p:



List based media object

content



List based media object

content



List based media object

content

Jumbotron

It has been for quite a while, since bootstrap was first released.
It's a kind of showcase area / header / hero where we have
some large heading with the text underneath

```
<!-- JUMBOTRON -->
<div class="jumbotron">
    <h1 class="display-4"> Welcome To My website! </h1>
    <p class="lead"> Lorem ipsum, dolor sit amet  
consectetur adipiscing elit. nostrum necessity
    </p>
```

<h2>

```
<p> Lorem ipsum dolor sit amet </p>
```

```
<a href="#" role="button"> Read More </a>
```

</div>

```
> class = "btn btn-primary btn-lg"
```

```
<!-- FLUID JUMBOTRON -->
```

```
<div class="jumbotron jumbotron-fluid">
```

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

```
<h1 class="display-4">
```

fluid Jumbotron </h1>

It will make
the Jumbotron
to expand
through
whole screen.

```
<p class="lead"> Lorem ipsum dolor  
sit amet consectetur
```

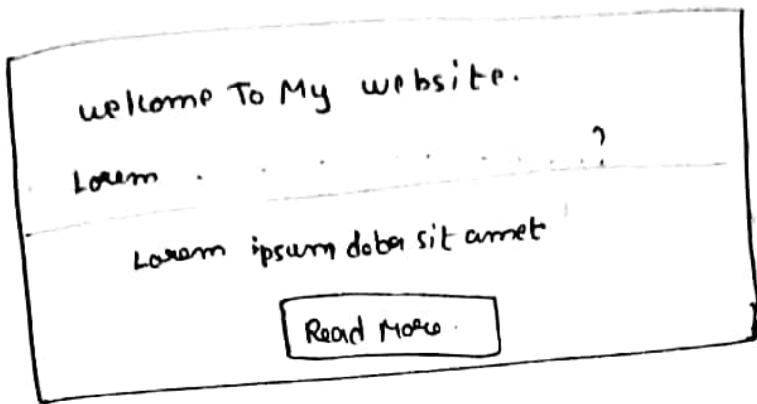
It will make
the background
to expand
and keeps its
content in place.

```
</p>
```

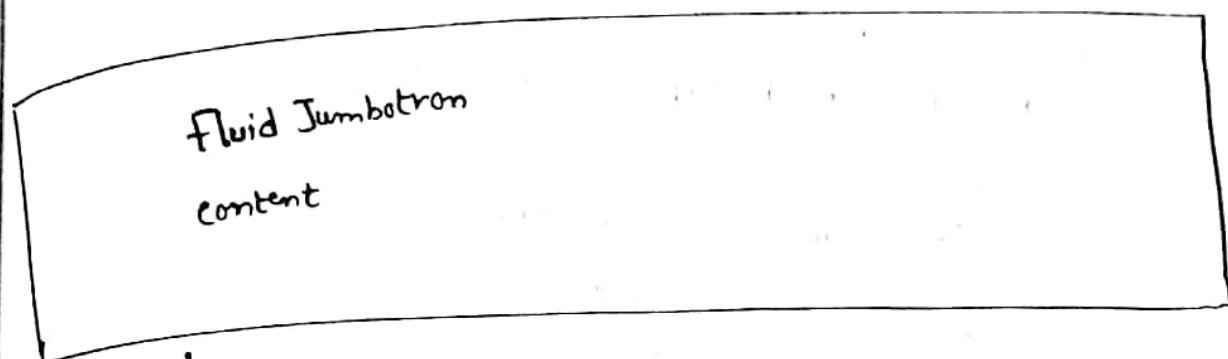
```
</div>
```

```
</div>
```

O/P for jumbotron.



O/P for fluid Jumbotron.



Grey Background will be extended all over the screen but the content will be in the container without being expanded.

Grid System

- Grid System is one of the important parts of Bootstrap framework.
- Grid System allows us to align things easily in our application / website and pretty much automatically responsive
- we can set different column sizes for different screen sizes,
- different screen sizes.

1. -- Responsive Grid -->

Bootstrap 4 grid uses
flexbox.

```
<div class="row">
  <div style="border: 1px solid #333">
    <ul>
      <li> 6 column on small screens </li>
      "   8 column on medium screen   "
      "   9 column on large screen   "
      " 10 column on x-large screen
    </ul>
  </div>
<div style="border: 1px solid #333">
  <ul>
    <li> 6 column on small screens </li>
    <li> 4 column on medium screen </li>
    <li> 3 column on large screen </li>
    <li> 2 column on x-large screen </li>
  </ul>
</div>
</div>
```

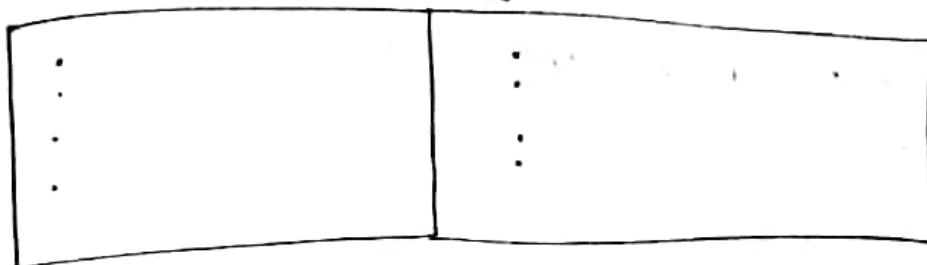
Input:

- 6 column on small screen
- 8 column on medium screen
- 9 column on large screen
- 10 column on x-large screen
- 6 column on small screen
- 11 column on medium screen
- 12 column on large screen
- 13 column on x-large screen



changes to Row:

Because the "Row" does changes it into a flex container.



If suppose, we want 6 columns on small Screen sizes then,

we need to add the class = "col-sm-6" to the second div.

```
<div class="col-sm-6" style="border: 1px solid #333">  
</div> <div> </div> <div> </div> <div> </div> <div> </div>
```

If we specify col-md-8 and col-md-4 then the o/p will be.

• 6 column on small screen	• 6 column on small screen
• 8 column on medium screen	• 4 column on medium screen
• 9 column on large screen	• 3 column on large screen
• 10 column on x-large screen	• 2 column on x-large screen

unless we specify the col size for large and x-large screens
the medium size will be reflected through the screen if even
it drag beyond the medium to large & large then also,
Medium class column size will be repeated.

Generally for large screens, the size will be

col-lg-9

col-lg-3

For extra small the columns in a Row will be stacked.

for xlarge screens,

"col-xl-10"

"col-xl-2"

Generally even without specifying the screen size we can mention the Equal width.

<!-- EQUAL WIDTH -->

<div class="row">

<div class="col" style="border: 1px solid #333"> Equal

width </div>

*3

</div>

Output:

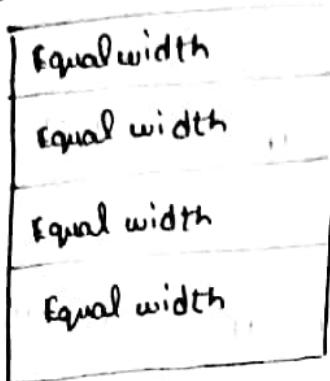
Equal width	Equal width	Equal width	Equal width
-------------	-------------	-------------	-------------

one thing to remember is that if we expand the screen to extra small then the four divs will wrap like below

Equal width	Equal width	Equal width
Equal width		

They won't get stacked on top of each other.

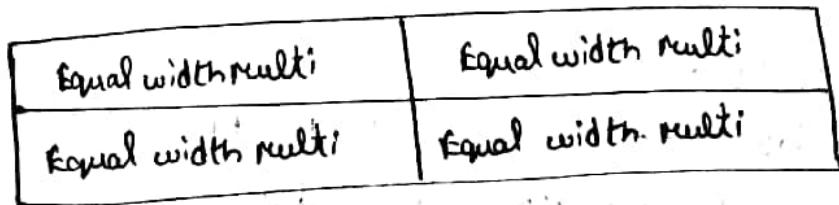
like:



<!-- EQUAL WIDTH MULT 1 Row -->

```
<div class="row">
  <div class="col" style="border:1px solid 0#333"> Equal
    width multi </div>
  <div class="col" style="border:1px solid 0#333">
    equal width multi </div>
  <div style="border:1px solid 0#333"></div>
)*2
</div>
```

O/P:



They are not gonna stack but they will stay at equal width.

<!-- AUTO LAYOUT -->

<div class="row">

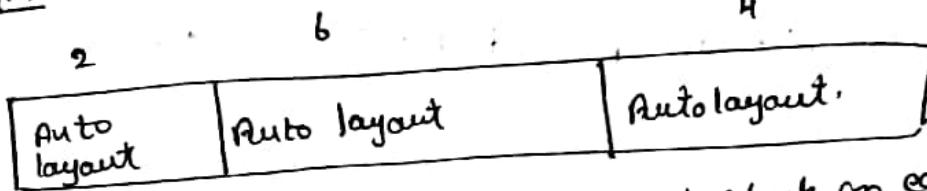
<div class="col" style="border: 1px solid #333">
Auto layout </div>

<div class="col-6" style="border: 1px solid #333">
Auto layout </div>

<div class="col-4" style="border: 1px solid #333">
Auto layout </div>

</div>

Op:



Suppose, if we make it smaller it will not stack on each other.

Suppose, if we want to make Equal width completely stacked them.

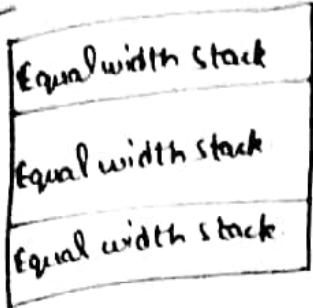
<div class="row">
<div class="col-sm" style="border: 1px solid #333">
Equal width stack </div>

) * 2

</div>

Generally if we specify screen size like col-sm (or)
col-lg (or) col-x1 then the Responsiveness will be
complete.

o/p:

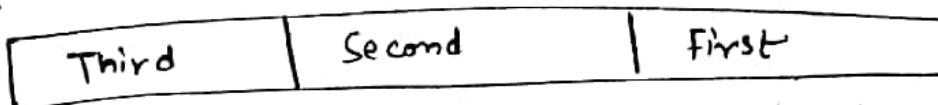


In the extra Small screen.

<!-- ORDERING -->

```
<div class="row">
  <div class="col order-3" style="border: 1px solid #333">
    first
  </div>
  <div class="col order-2" style="border: 1px solid #333">
    Second
    <div>
      <div class="col" style="border: 1px solid #333" order-1>
        Third
      </div>
    </div>
  </div>
```

o/p:

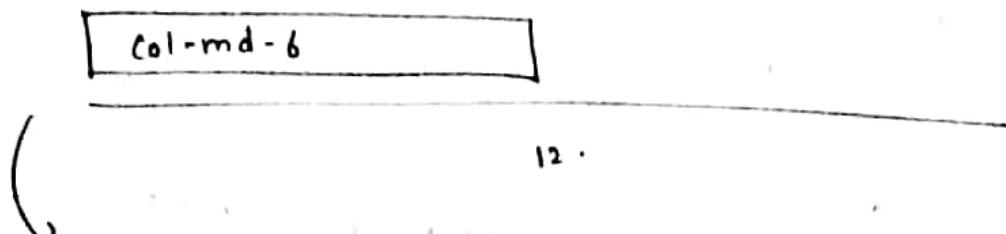


offsetting is really an important concept

Because Generically we have 12 col division.

6	6
8	4
9	3
10	2

Suppose take col-md-6.

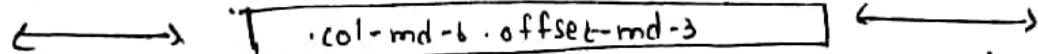


If we want to push it to the middle having Margin of 3 spaces from left & Right then,

```
<div class="col-md-6 offset-md-3" style="border:1px solid 0#333">.col-md-6.offset-md-3</div>  
</div>
```

Opp:

3 Spaces



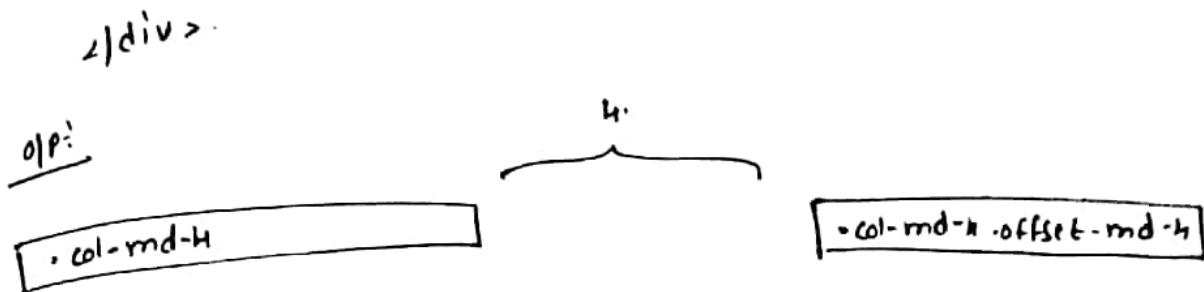
3 Spaces

"space: cols":) in this case.

Now

<!-- 2 4 column divs , the second offset by 4 -->

```
<div class="col-md-4" style="border: 1px solid #333"> .col-  
md-4  
</div>  
<div class="col-md-4 offset-md-4" style="border: 1px  
solid #333"> .col-md-4 .offset-md-4
```

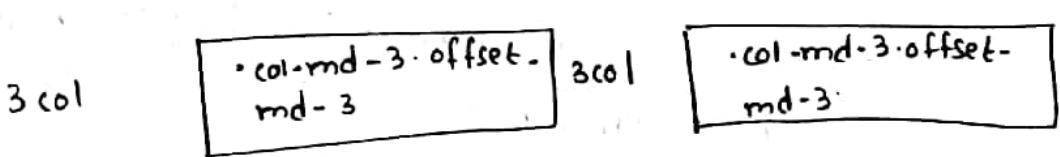


! -- 2 3 column divs, both offset by 3 -->

```
<div class="col-md-3 offset-md-3" style="border: 1px solid  
#333"> .col-md-3 .offset-md-3 </div>  
<div class="col-md-3 offset-md-3" style="border: 1px  
solid #333"> .col-md-3 .offset-md-3 </div>
```

```
</div>
```

Op:



! -- NESTING -->

```
<div class="row">  
    <div class="col-sm-9" style="border: 1px solid #000">  
        level 1: .col-sm-9  
        <div class="row">  
            <div class="col-8 col-sm-6" style="border: 1px solid  
            #000">  
                level 2: .col-8 .col-sm-6
```

```
</div>
```

```
<div class="col-4 col-sm-6" style="border: 1px solid #000;">  
    level 2: .col-4 .col-sm-6
```

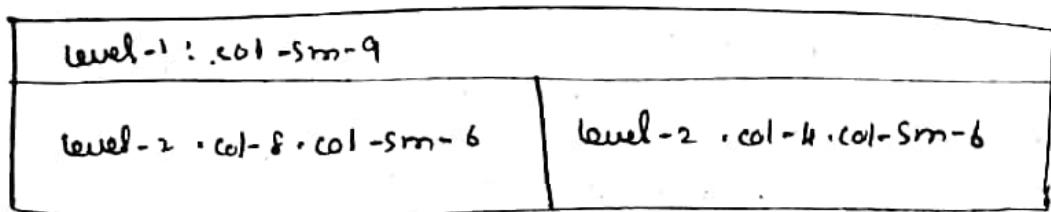
```
</div>
```

```
</div>
```

```
</div>
```

```
</div>
```

o/p:



Grid alignment

It is basically how we align items within the Grid.

!--> vertical Alignment -->

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

```
    <div class="row align-items-start" style="height: 200px;  
        border: 1px solid #333;">
```

```
        <div class="col">
```

Top Aligned Text

```
</div>
```

```
<div class="col">
```

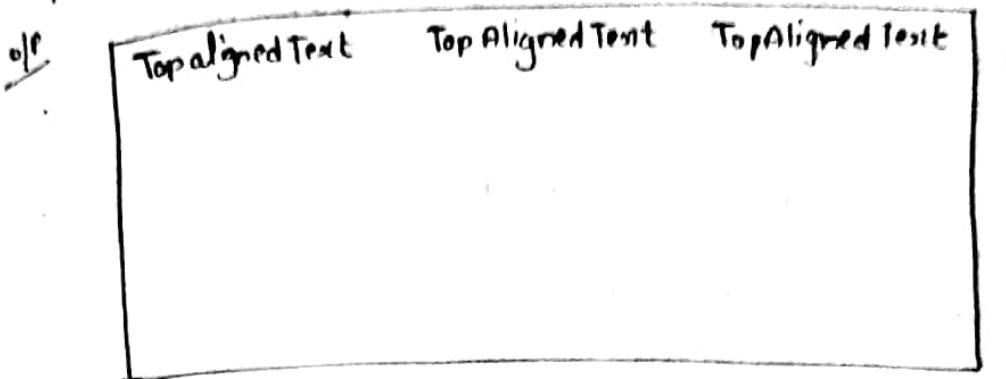
Top Aligned Text

```
</div>
```

```
<div class="col">
```

Top Aligned Text

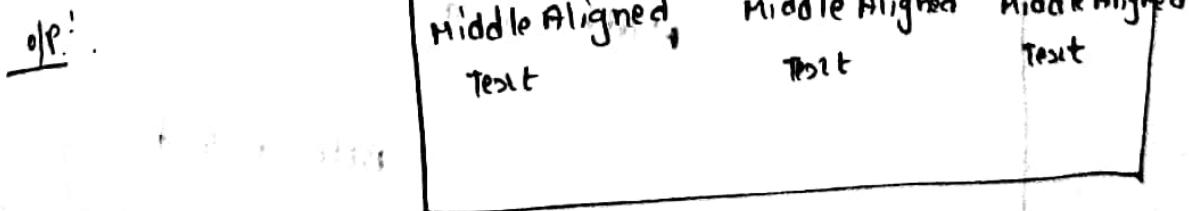
```
</div>  
<div class="col">  
    Top Aligned Text  
</div>
```



If we want to align our content in the middle then,
we have specify "align-items-center"

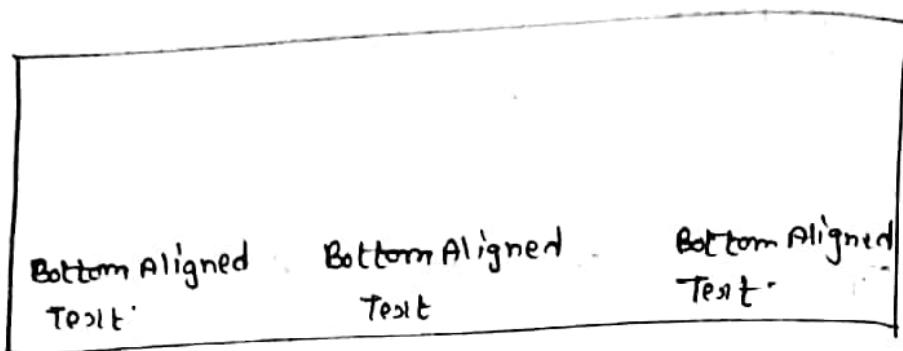
Syntax & usage

```
<div class="row align-items-center"  
      style="height: 200px; border: 1px solid black">  
    <div class="col">  
        Middle Aligned Text  
    <div class="col">  
        Middle Aligned Text  
    <div class="col">  
        Middle Aligned Text  
</div>
```



If we want to align the items to the bottom, then we must specify "align-items-end"

O/P:



If we want to fix the position of individual col then we should specify the self class on the column,

<!-- vertical align individual cols -->

<div class="container">

<div class="row" style="height: 200px; border: 1px 0 #333 solid">

<div class="col align-self-start">

Top Aligned

</div>

<div class="col align-self-center">

Middle Aligned

</div>

<div class="col align-self-end">

Bottom Aligned

</div>

</div>

</div>

O/P:



! -- Horizontal Aligned col's -- !

```
<div class="container">
  <div class="row justify-content-start"
    style="height: 200px; border: 1px solid #333">
    <div class="col-4">
      left aligned Text
    </div>
    <div class="col-4">
      left aligned Text
    </div>
    <div>
```

</div>

</div>

o/p:

left aligned Text right aligned Text

If we specify `justify-content: center;`

then

o/p:

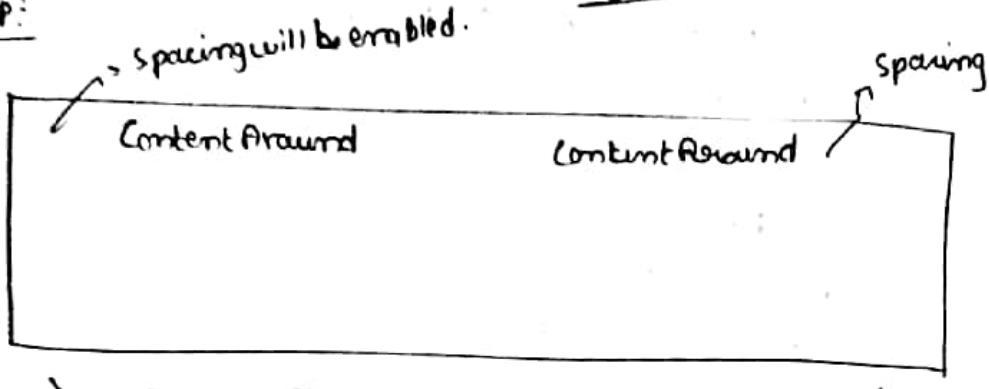
center aligned Text center aligned Text

If we want to align the items to the right then we need to
specify "justify-content: end";

we also have "content-around" what this will do is it will add the spacing around the content

Syntax: justify-content-around

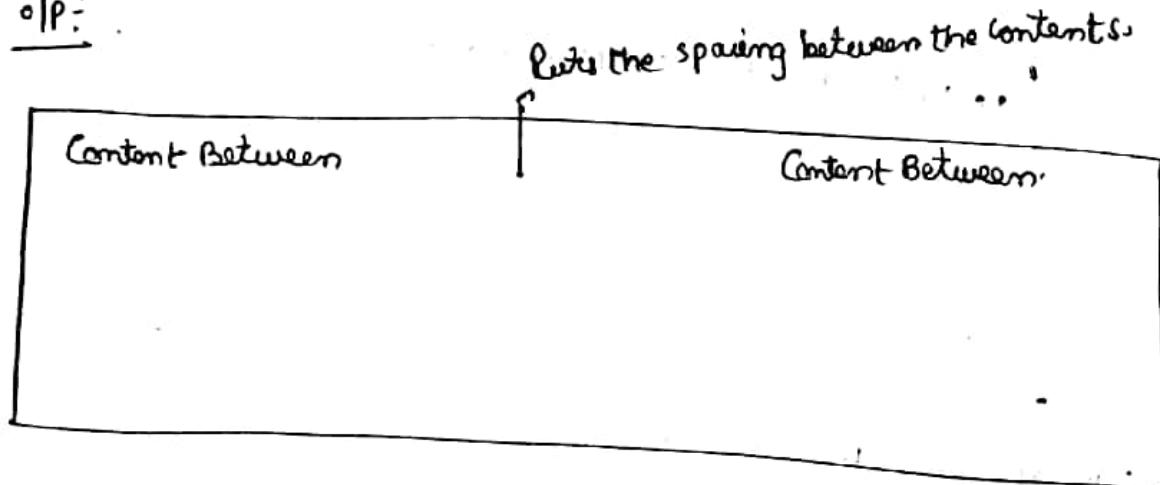
o/p:



It is not centered but some space is added.

we can also put "justify-content-between"

o/p:

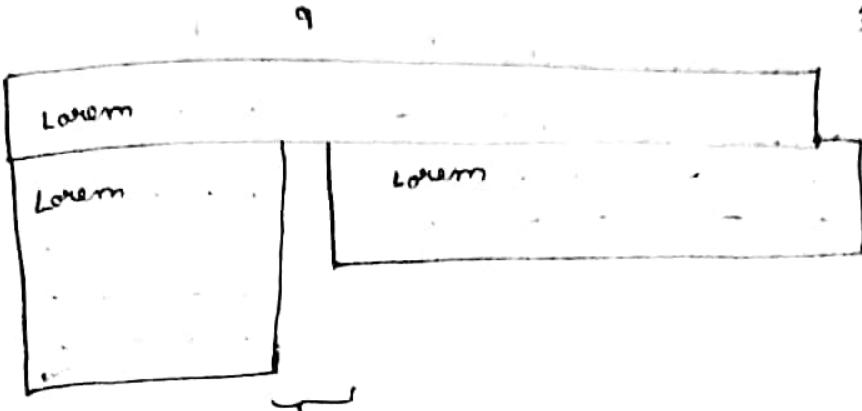


<!-- COLUMN WRAPPING & NO GUTTERS -->

```
<div class="row no-gutters">
  <div class="col-9">
    <div class="card">
      <div class="card-block">
        <p> Lorem . . .
      </div>
    </div>
  </div>
</div>
```

Suppose, if for the most divi there are col-4 and col-6

o/p will be like

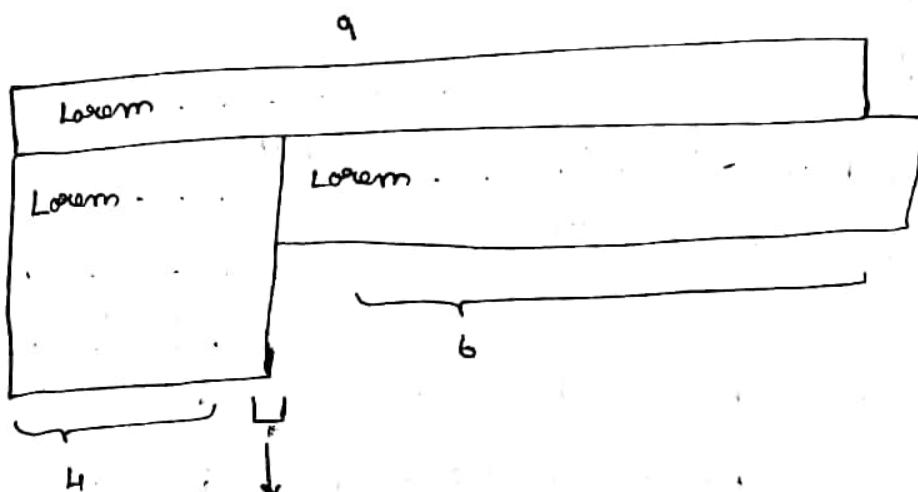


If we want to remove such space then we need to use no-gutters.

<div class="row no-gutters">

then now the o/p will be.

o/p



Now, we can observe that the space has been removed.

Flexbox

we know that the Grid system uses "flexbox" but we also know that we can use "flex" without the grid or separate elements (box) wherever we like.

```
<style>
```

```
.row-hl {  
    background: #f4f4f4;  
    height: 250px;
```

y

```
.item-hl {
```

```
    border: 1px solid #333;
```

y

```
</style>
```

<!-- flex Row & flex Items --> makes rows → cols.

<div class="d-flex flex-row row-hl">

By default

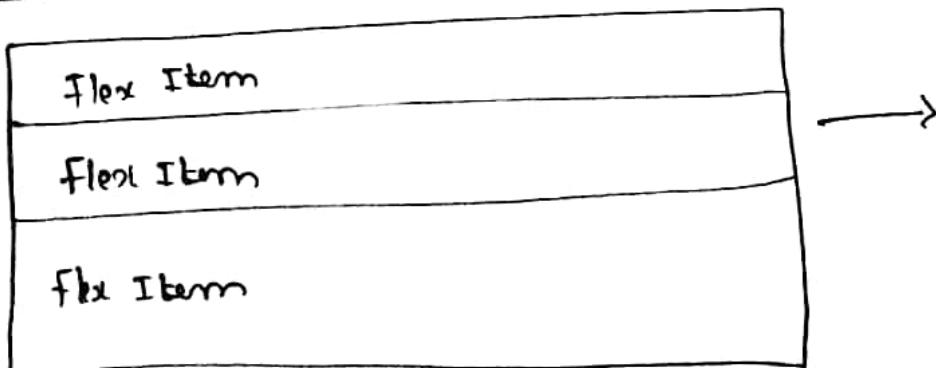
<div class="p-4 item-hl"> Flex Item </div>

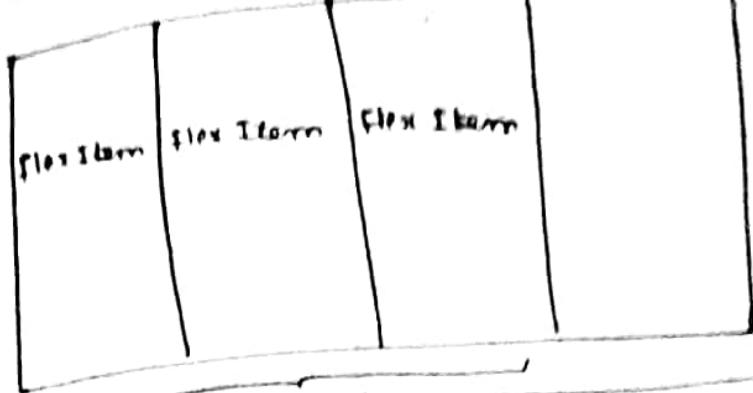
<div class="p-4 item-hl"> Flex Item </div>

<div class="p-4 item-hl"> Flex Item </div>

```
</div>
```

Output:

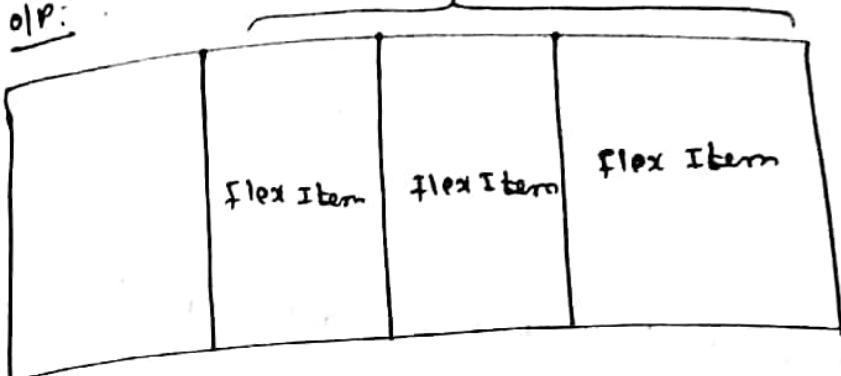




! -- flex Row Reversed -->

`<div class = "d-flex flex-row-reverse row-hl">`

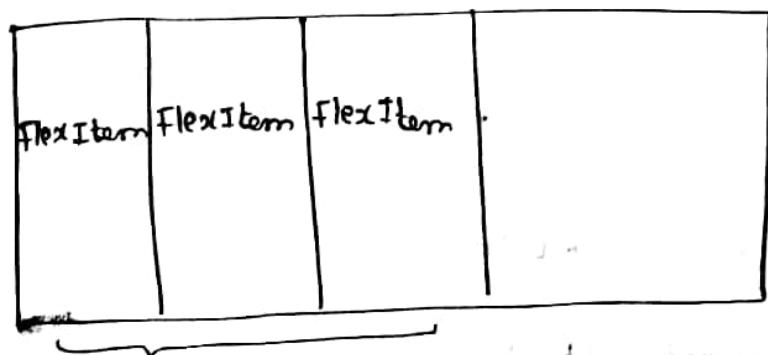
o/p:



! -- Justify Content -->

`<div class = "d-flex justify-content-start row-hl">`

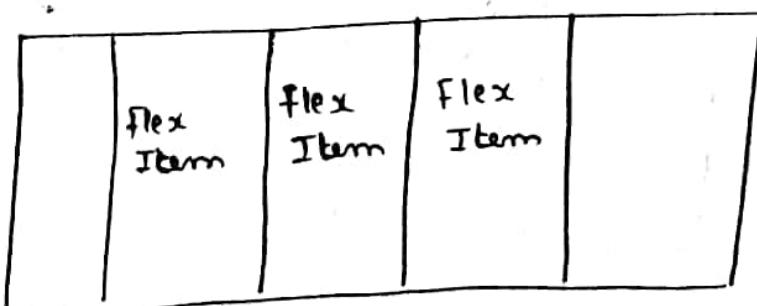
o/p:



Same as default d-flex.

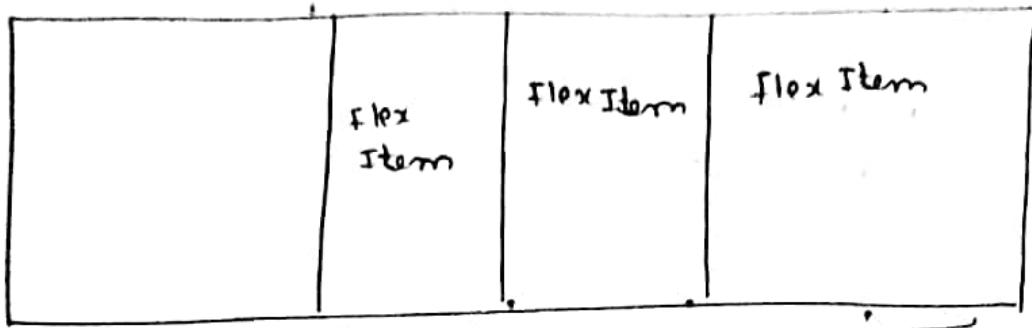
If we specify justify-content-center then

o/p:



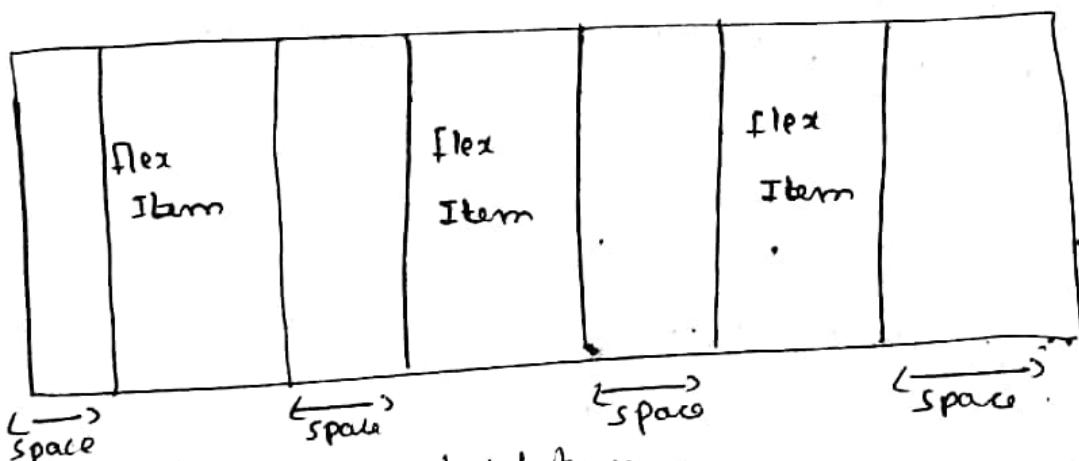
If we want to align the contents to the end,
→ `div class="d-flex justify-content-end row-hl"`

o/p:



... all of equal width aligned to
the right.

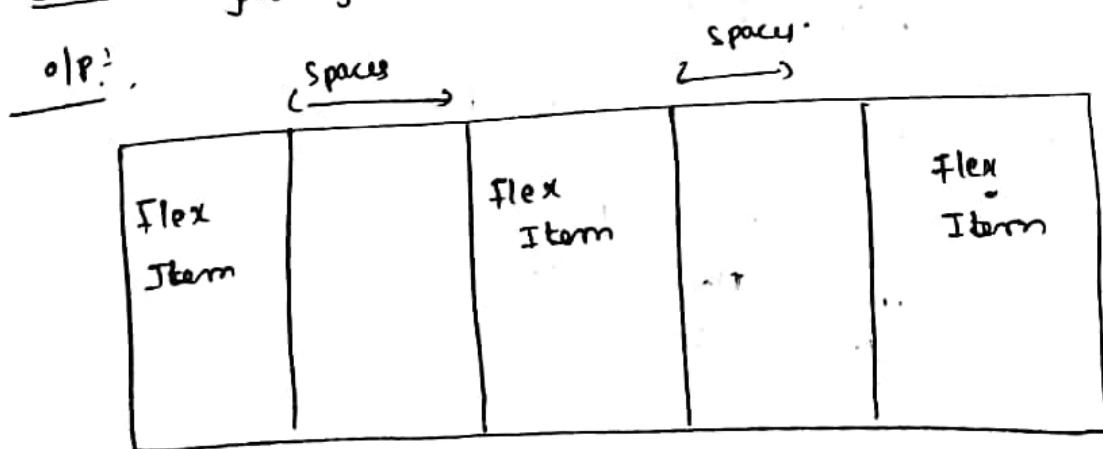
we can also specify "justify-content-around" in order to
provide spaces in between the flex items.



we can also justify the content between.

Syntax: "justify-content-between"

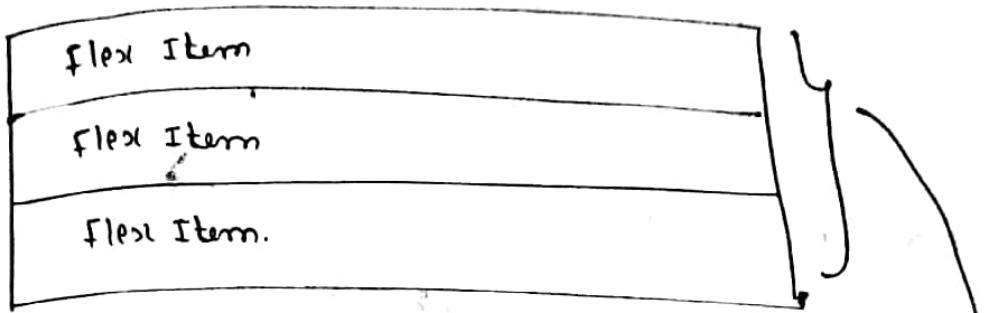
o/p:



The flex items could be anything like Paragraphs, Images etc whatever we like.

1. -- flex column ->
<div class="d-flex flex-column row-hl">

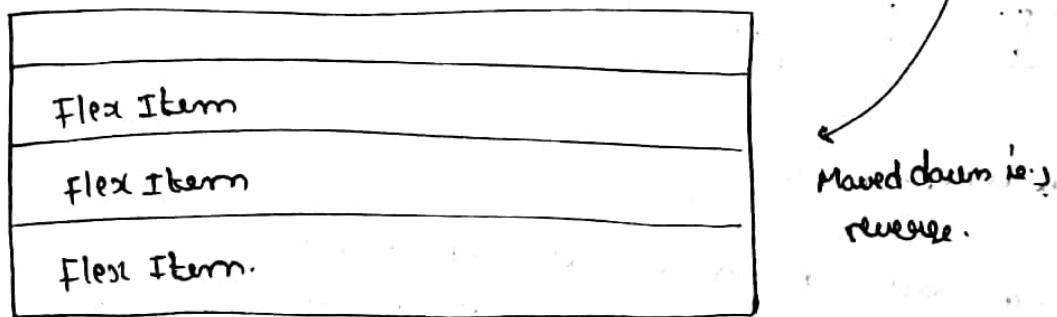
O/P:



Now if we want the reverse of columns then,

<div class="d-flex flex-column-reverse row-hl">

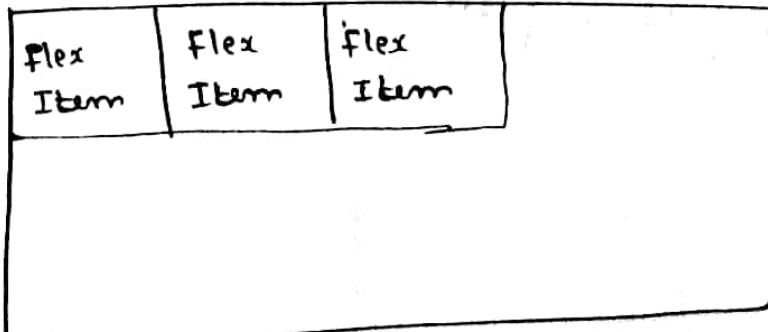
O/P:



2. -- vertical Align Items ->

<div class="d-flex align-items-start row-hl">

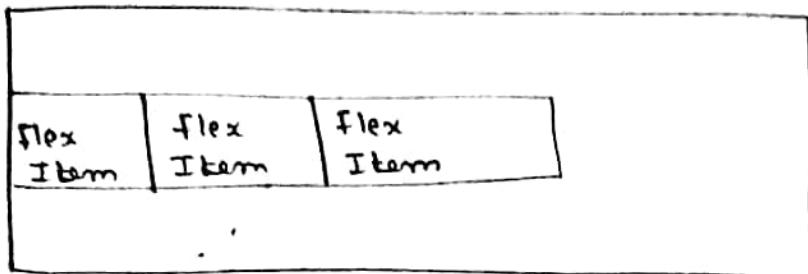
O/P:



If we want to align them to center them,

<div class="d-flex align-items-center row-hl">

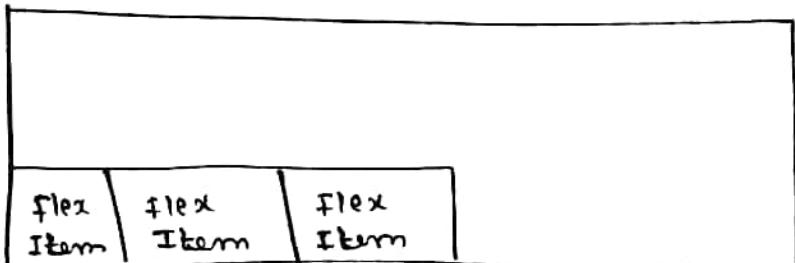
O/P:



vertical align bottom of items

<div class="d-flex align-items-end row-hl">

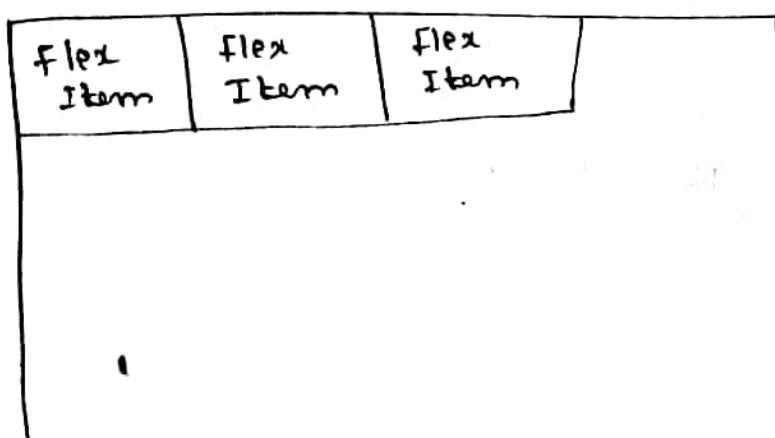
O/P:



vertical align baseline

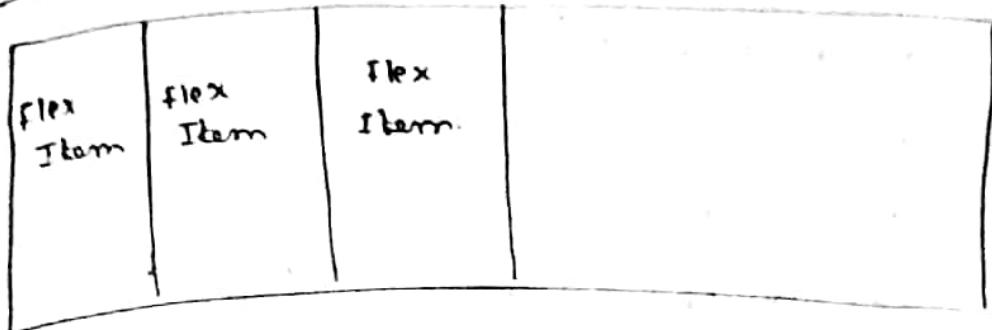
<div class="d-flex align-items-baseline row-hl">

O/P: Somewhat same as "align-items-start"
But the CSS property will
be different.



we can also stretch the items such that they go up and down.
<div class="d-flex align-items-stretch row-hl">

op:



<!-- align self -->

<div class="d-flex row-hl">

<div class="align-self-start p-4 item-hl">flex item</div>

<div class="align-self-center p-4 item-hl">flex item</div>

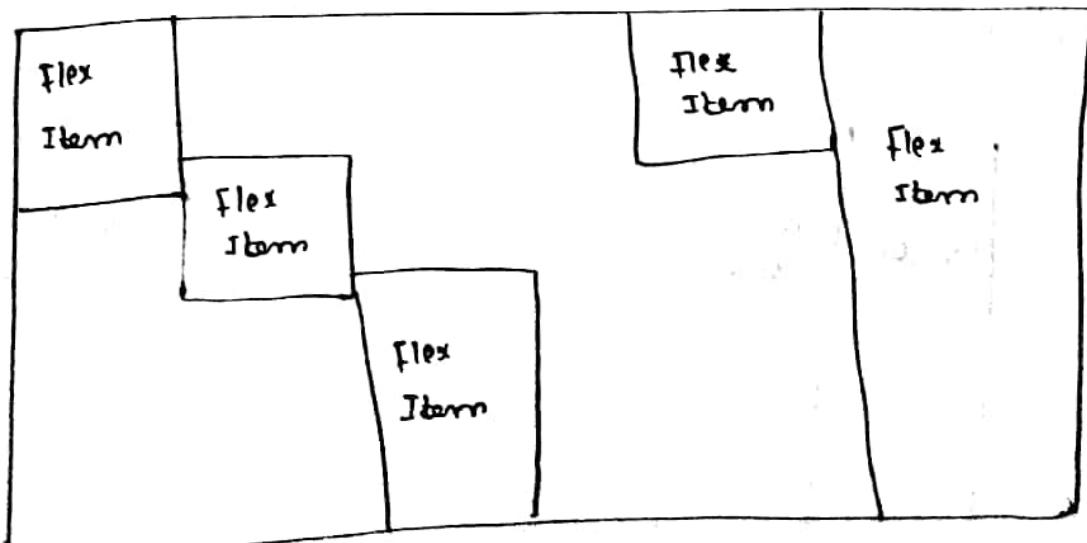
<div class="align-self-end p-4 item-hl">flex item</div>

<div class="align-self-baseline p-4 item-hl">flex item</div>

<div class="align-self-stretch p-4 item-hl">flex item</div>

</div>

op:

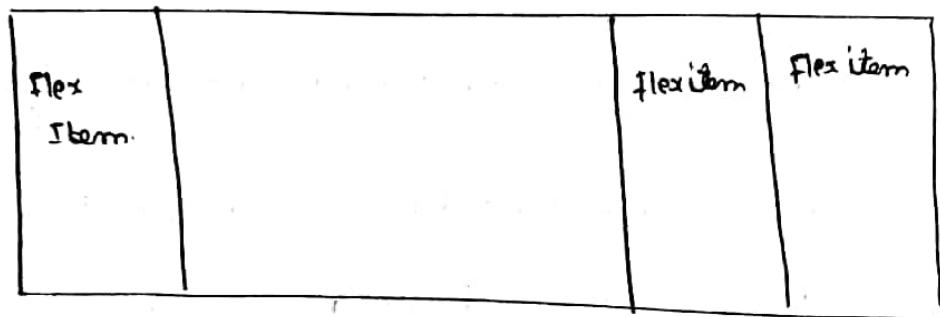


Auto Margins & wrapping.

<! -- MR-AUTO -->

```
<div class="d-flex row-hl">  
  <div class="mr-auto p-2 item-hl"> flex item </div>  
  <div class="p-2 item-hl"> Flex item </div>  
  <div class="p-2 item-hl"> flex item </div>  
</div>
```

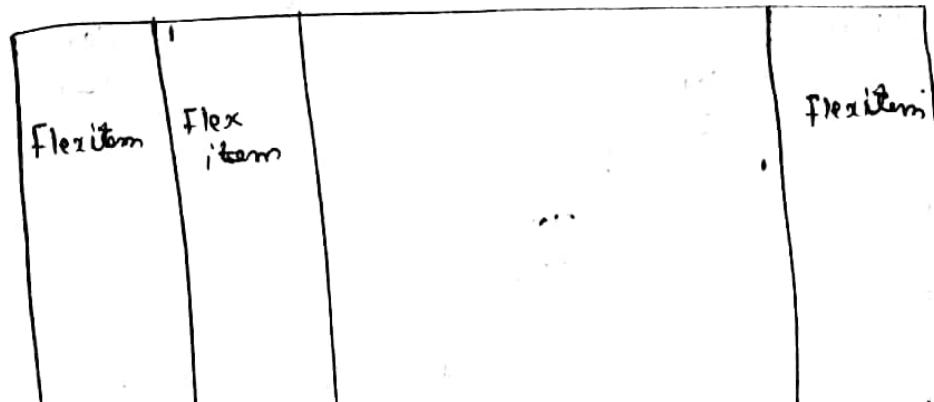
o/p:



<! -- ML-AUTO-->

```
<div class="d-flex row-hl">  
  <div class="p-2 item-hl"> flex item </div>  
  <div class="p-2 item-hl"> flex item </div>  
  <div class="ml-auto p-2 item-hl"> flex item </div>  
</div>
```

o/p:



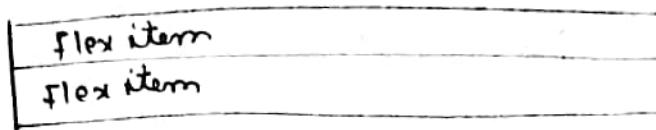
we can also use margin-bottom & margin-top auto

1 - - MB - AUTO - - 3

```
<div class = "d-flex flex-column row-hl">
  <div class = "mb-auto p-2 item-hl"> flex item </div>
  <div class = "p-2 item-hl"> flex item </div>
  <div class = "p-2 item-hl"> flex item </div>

</div>
```

O/P:

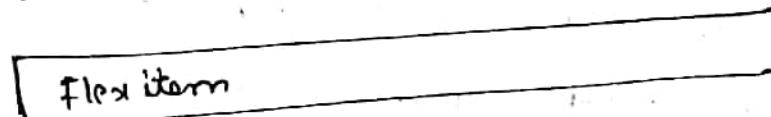


margin-top Auto:

```
<div class = "d-flex flex-column row-hl">
  <div class = "p-2 item-hl"> flex item </div>
  <div class = "p-2 item-hl"> flex item </div>
  <div class = "p-2 item-hl"> flex item </div>
    <br> mt-auto
```

</div>

O/P:



wrap

```
<div class="d-flex flex-wrap row-hl">
```

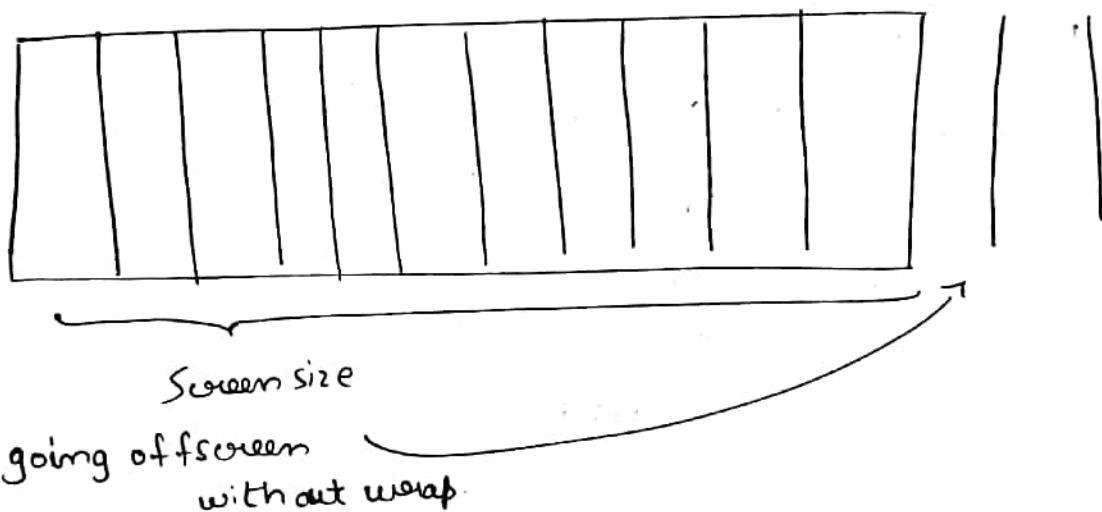
flex.
→ item

o/p:



If by default there flex-no-wrap then,

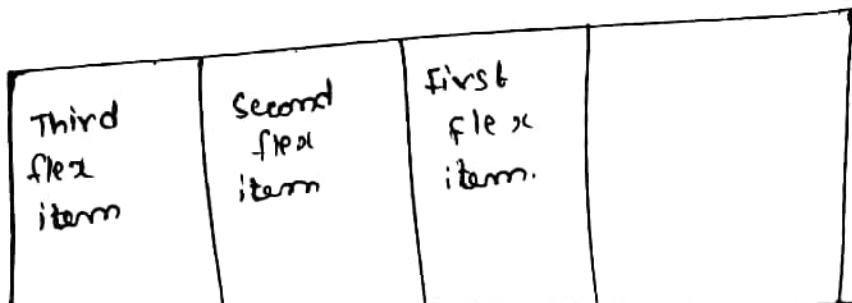
o/p:



-- ORDERING -->

```
<div class="d-flex row-hl">
  <div class="order-3 p-2 item-hl"> first flex item </div>
  <div class="order-2 p-2 item-hl"> Second flex item </div>
  <div class="order-1 p-2 item-hl"> Third flex item </div>
</div>
```

o/p:



JavaScript widgets

If we are doing some kind of Projects out of Sandbox, we need to include scripts of "jQuery", "Popper.js" and "bootstrap"

simple slider

```
<div class="container">  
  <div class="row">  
    <div class="col-sm-8 m-auto">  
      <!-- SIMPLE SLIDER -->
```

To create a slider, we need to give it a class.

```
<div class="carousel slide mb-5" data-ride="carousel">  
  <div id="Slider1">
```

```
    <div class="carousel-inner" > active.
```

```
      <div class="carousel-item" >
```

```
        
```

```
    </div>
```

```
    <div class="carousel-item" >
```

```
      
```

```
  </div>
```

```
  <div class="carousel-item" >
```

```
    
```

```
  </div>
```

```
  </div>
```

```
</div>
```

when we add a class = "d-block img-fluid" it's gonna be its size relative to the screen size.

Also, the screen why we specified "active" is that when the screen was opened then only that image will be visible.

Q.P:



Later after some time second Image and then third Image.

Slider with controls

copy the before code.

After the "carousel-inner" div,

Type "a.carousel-control-prev" then we get,

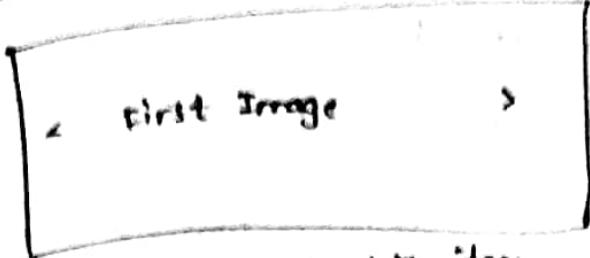
It is the slider to which it points.

<a href="#" class="carousel-control-prev"
data-slide="prev">

But put "prev".
<a href="#" class="carousel-control-next"
data-slide="next">

</div>

O/P:



Now we have arrows on both sides.

We can also add "indicators".

Copy the first code i.e., "simple slider" along with the "controls" code.

Now, Right above the "carousel-inner" put an ordered list "ol.carousel-indicators".

Type

```
<ol class="carousel-indicators">  
    <li class="active">
```

Type

Then

```
    <li class="active" data-target="#Slider3"  
        data-slide-to="0"></li>
```

first image
will be shown active.

Now again type

```
    <li class="active" data-target="#Slider3"  
        data-slide-to="1"></li>
```

```
    <li data-target="#Slider3" data-slide-to="2"></li>
```

No need of
this because
at once both
classes can't
be active.

O/P:



Slider with captions

"In order to add caption

under each image in the previous code, but

```
<div class="carousel-item d-none d-md-block">
  <h3>Slide one </h3>
  <p> lorem10 </p>
</div>
```

This says don't add the caption.

→ 2. { for other two Images
too, add it }.

This says only when the
screen is medium and
the caption otherwise "No".

<!-- controls -->

To edit the properties of
carousel class we need to go
to <script> tag and type the
following.

<script>

```
$( '.carousel' ).carousel(
```

{ interval : 1000 }

keyboard : true

});

If we want to

call the image on

hover, we can do that

Pause : ' hover '

It is 1sec.

If we hover over a particular
image, when it is sliding then
the image will be fixed.

we can also a wrap option
wrap: true } By default

wrap: false → means after going to last image, it will stop without any further going.

```
$( "#Slider" ).on( 'slide.bs.carousel', function() {  
    console.log('SLIDE!');  
});
```

y);

```
$( "#Slider" ).on( 'slide.bs.carousel', function() {  
    console.log('SLIDE!');  
});
```

y);

Collapse & Accordion

```
<button class="btn btn-primary d-block mb-4"  
       data-toggle="collapse" data-target="#collapse-btn-1">→  
       Read More </button>
```

```
<div class="collapse mb-5" id="collapse-btn-1">  
    <div class="card">
```

```
        <div class="card-body">
```

Learn to

</div>

</div>

</div>

Now: open:

Read More.

— now when we click this
the text will
be visible and when again
click the text, disappears

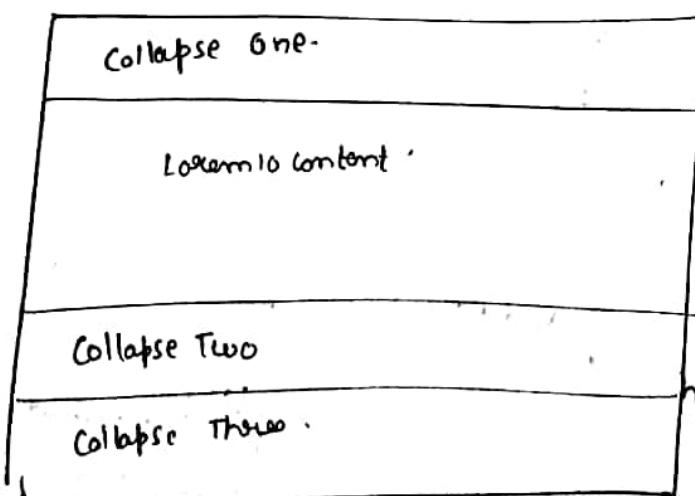
Accordion (multiple collapses).

```
<div id="accordion">  
  <div class="card">  
    <div class="card-header">  
      <h3>  
        <a href="#" data-parent="#accordion" data-toggle="collapse" data-target="#collapseOne">  
          collapse one.  
        </a>  
      </h3>  
      <div>  
        <h3>  
          <div id="collapseOne" class="collapse" data-parent="#accordion" data-toggle="collapse" data-target="#collapseOne">  
            <div class="card-body">  
              Lorem10.  
            </div>  
          </div>  
        </div>  
      </div>  
    </div>  
  </div>
```

→ 2.

when we click collapse one
then it is disappear.

O/P:



ToolTips

we can put tooltip on any element.
 after we hover over an element, it just appears and we can control the positioning.
 Now we are doing "buttons"

! -- Tool Tips -->

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

similarly, we have other three values for
 data-placement like.

→ right

→ bottom

→ left

Q/P:
 we are gonna use "jQuery" to use the tooltip.

In script tag

<script>

//Init tool tips

```
$(' [ data-toggle ] = "tooltip" ').tooltip();
```

</script>

Now, we get tooltip on top, right, bottom and left.

we can also put HTML tags on a tooltip

```
<!-- Tooltips with HTML -->
<button id="hello" class="btn btn-primary" data-toggle="tooltip"
        data-placement="top" title="<h3>Hello</h3>">
    data-html="true"
<p>Hello</p>
> ToolTip
with HTML
</button>
```

we can trigger any HTML using JavaScript

```
<button class="btn btn-secondary" onclick="showToolTip();">
    Show Tool Tip
</button>
<button class="btn btn-secondary" onclick="hideToolTip();">
    Hide Tool Tip
</button>
<button class="btn btn-secondary" onclick="toggleToolTip();">
    Toggle Tool Tip
</button>
```

Go to script tag and create the functions

```
<script>
function showToolTip() {
    $('#hello').tooltip('show');
}

</script>
```

(,)*
y
2

O/P:
for Tool tips with HTML we have got:

Tooltip on with HTML

Now we have three Buttons.

Show Tooltip

Hide Tooltip

Toggle Tooltip

when we click ↴

[Hello
HelloWorld] will be shown as a tooltip

when
we
click
here.

it will be
hidden

when clicked
this then
it will be
toggling.

In <script>:

```
$('#hello').on('show.bs.tooltip', function () {  
    console.log('Tooltip shown');  
})
```

// Tooltip Events:

```
$('#hello').
```

* 4

Instead of show,
shown

hide
hidden

Popovers

They work very similar to tooltips

```

<div class="container">
  <!-- Popover -->
  <div id="hello" data-toggle="popover"
    data-trigger="button" data-placement="top" data-content="Learn to
    Toggle Popover."
  </div>
  <button class="btn btn-primary" data-toggle="popover" data-trigger="click" data-placement="right" data-content="I'm a popover!">Toggle Popover</button>
</div>

```

Like Tooltip, change the Placement to right, left, bottom

when we click over the Button, it doesn't do anything again.
we have to initialize it like we done with the tooltip case
using Jquery.

```
<script>
```

```
// Init Popover
```

```
$('[data-toggle="popover"]').popover()
```

```
</script>
```

For Popover, we have to click.

```
<!-- DISMISSABLE Popover -->
```

copy the same code above and just add a class

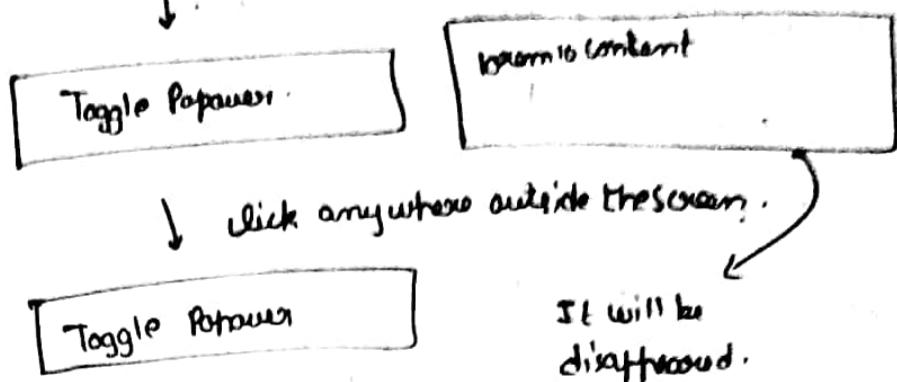
namely data-trigger = "focus"

O/P:

Toggle Popover

when we click on it

toggle *Lo*gout content would be displayed.



!.. CONTROL WITH JS -->

```
<button class="btn btn-secondary" onclick="showPopover()>  
    Show Popover </button>.  
<button class="btn btn-secondary" onclick="hidePopover()>  
    Hide Popover </button>  
<button class="btn btn-secondary" onclick="togglePopover()>  
    Toggle Popover </button>
```

```
</div>
```

```
<script>
```

```
function showPopover() {  
    $('#hello').popover('show');
```

```
y
```

```
function hidePopover() {
```

```
    $('#hello').popover('hide');
```

```
y
```

```
function togglePopover() {
```

```
    $('#hello').popover('toggle');
```

```
y
```

```
</script>
```

// Popover events

```
$( '#hello' ).on( 'show.bs.popover', function() {  
    console.log( 'Popover show' );  
});
```

(y),
(3, 3.

now mention inside.

shown

hide

hidden

Now when I clicked on the

Toggle popover

I will

get Popover shown in the console

Popover shown and when I come out of

Popover hide . Toggle Popover

Popover hidden. I will get Popover hide
Popover hidden, in the console

Modal

A Modal is like a lightbox i.e., it's like a PopUp that fades in and we can put pretty much any kind of content inside of it

<!-- MODAL TRIGGER -->

```
<button class="btn btn-primary" data-toggle="modal"  
data-target="#myModal">Launch Modal </button>
```

J.

Indicates which modal to open.

```
<!-- MODAL -->
<div class="modal" id="myModal">
  <div class="modal-dialog">
    <div class="modal-content">
      <div class="modal-header">
        <h3 class="modal-title">Modal Title</h3>
      <button class="close" data-dismiss="modal">&times;</button>
      <div>
        <div> class="modal-body">
          Lorem 10'
        </div>
      </div>
      <div class="modal-footer">
        <button class="btn btn-secondary" data-dismiss="modal">Close</button>
      </div>
    </div>
  </div>
</div>
```

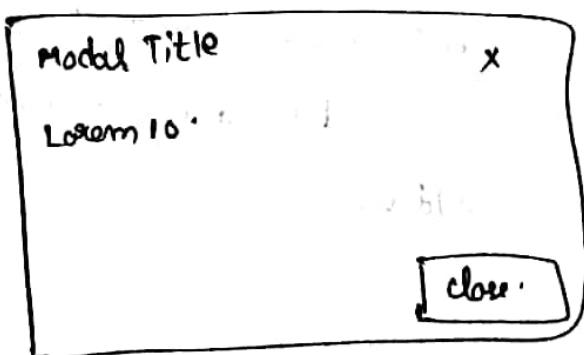
O/P:

Launch Modal

↓ when we click on it

Launch Modal

fades up and



Now, we want a form inside the Modal.
like for eg: Registration form.

-!- LOGIN MODAL TRIGGER - ->

copy the previous button code here.
and change the text to "login".
and also the other main Modal code.

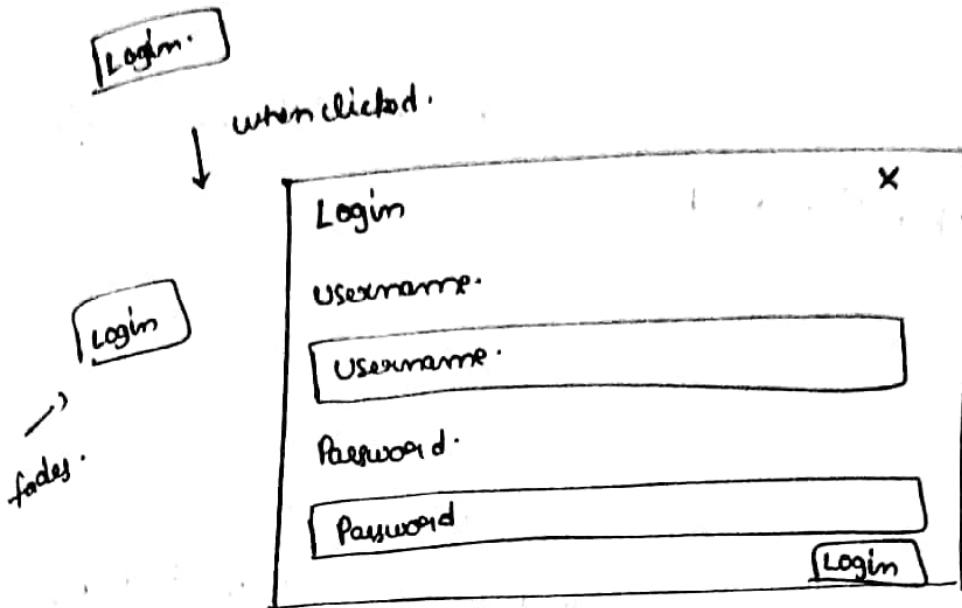
```
<div class="modal-body">  
    <form>  
        <div class="form-group">  
            <label for="username"> Username </label>  
            <input type="text" class="form-control" placeholder="Username" />  
        </div>  
    </form>
```

replace Username with Password,

and type = "password".

Now in the modal-footer.

```
<div class="modal-footer">  
    <button class="btn btn-primary"  
          data-dismiss="modal"> Login </button>  
    </div>
```



Scandal Spy

ScrollSpy: when we have certain sections of content on a page and we have navigation items that pertains to them.

Mike "

foreg:

new bar

<u>welcome</u>	<u>about</u>	<u>Services</u>
w when hit this ↑	w when ↑	w when ↑.
↓	hit	
welcome	about	Services
will be	will be	will be
highlighted	highlighted	highlighted.

when we scroll there should be some sort of smooth scrolling effect and that can be accomplished with the help of "Jquery".

```
<script>
  $('body').scrollspy({
    target: '#main-nav'
  })
</script>
It is for.
```

Navbar.

Paragraph of welcome

Paragraph of About

Paragraph of Services

//Add smooth scrolling.

```
$('#main-nav a').on('click', function(e) {
```

// check for a hash value

```
if (this.hash != '')
```

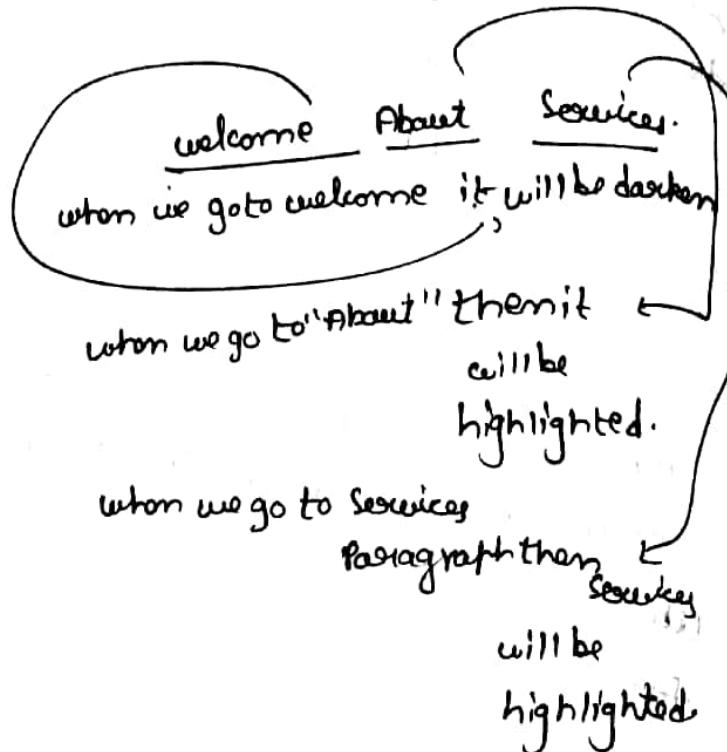
```
{ // Prevent default behavior.
```

```
e.preventDefault();
```

// Store hash

```
const hash = this.hash;
```

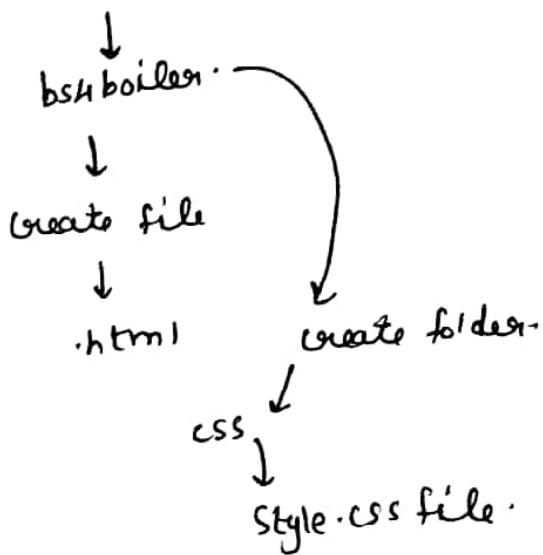
// Animate smooth scroll



```
$('html, body').animate( {  
    scrollTop: $(hash).offset().top  
}, 900, function() {  
    // add hash to URL after scroll  
    window.location.hash = hash;  
});  
}  
y  
y);  
</script>
```

Creating the Boiler

Create new folder.



Create another folder called "img"

! + shift

(Bootstrap.com)

copy the css link and Paste it above the
custom style sheet

copy the js files too and then go to
code.jquery.com -> Gzominified → Put that above

other two JS script tags

<script>

</script>