



# BOOTSTRAP – BY ANGELA YU

SPECIAL THANKS TO [NINAD DHULAP](#)

---

**Responsive site:** if you're looking at the same web site on mobile or on iPad or on desktop, it should have a different layout to be able to best take advantage of the size of the screen real estate. Website changes its design depending on where it's being displayed (adaptive site). (it respond to screen size).

**Bootstrap:** if you add a particular class name, then it will automatically style it in the way that has been specified by the bootstrap code.

**CDN: Content Delivery Network.** Instead of hosting your web site just in a single location you actually have a whole bunch of points where that website can be accessed.

Therefore your website start searching for shortest way to server, so this file –  
`<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-1BmE4kWBq78iYhFIdvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3" crossorigin="anonymous">`  
file can download fastly.

Always put bootstrap link above your CSS link to overwrite style.

**Wireframes:** wireframes are a low fidelity representation of your design.

before touching any code, any CSS, or even creating your website the first thing that you need to do is to wireframe.

**Mock-up:** mock-up is meant to be a high fidelity representation of your app or your web site's design.

if you are getting somebody else to create your web site then it's nice to have a mock-up to show. (like screenshot of website)(use photoshop/illustrator/)  
or also animated video.

Ideas – [dribbble.com](https://dribbble.com) , [UI-Pattern.com](https://UI-Pattern.com).

Wireframes – [Sneakpeak.com](https://Sneakpeak.com) , [Balsamiq.com](https://Balsamiq.com)

[.navbar](#): for navigation bar.

[.navbar-toggler](#) to add expand-collapse button.

[Layout > grid](#) :

Class = 'col-1 to col-12' means 12 boxes along the width. If we want column in 50 % then write col-6.

```
1 * <body>
2 *   <div class="row">
3 *     <div class="col" style="background-color: red; border: 1px solid;">
4 *       col
5 *     </div>
6 *     <div class="col" style="background-color: red; border: 1px solid;">
7 *       col
8 *     </div>
9 *   </div>
10 *  <div class="row">
11 *    <div class="col-6" style="background-color: green; border: 1px solid;">
12 *      col-6;
13 *    </div>
14 *    <div class="col-3" style="background-color: green; border: 1px solid;">
15 *      col-6
16 *    </div>
17 *  </div>
18 </body>
```



But this columns aren't responsive with different devices. To make it responsive code is below ↓

```
<div class="row">
  <div class="col-md-6" style="background-color: yellow; border: 1px solid;">
    col-md-6
  </div>
  <div class="col-md-6" style="background-color: yellow; border: 1px solid;">
    col-md-6
  </div>
</div>
```



↑ Laptop view.

← Mobile view. When size of screen is less than medium it takes down 2<sup>nd</sup> col to bottom.

col-md-6 means is that we should have a 6 unit column on any size from medium size or tablet size upwards, but anything smaller than the medium they will take up the full width.(ex. Mobile)

responsive code -

```
<div class="row">
  <div class="col-lg-3 col-md-4 col-sm-6" style="background-color: yellow; border: 1px solid;">
    col-lg-3
  </div>
  <div class="col-lg-3 col-md-4 col-sm-6" style="background-color: yellow; border: 1px solid;">
    col-lg-3
  </div>
  <div class="col-lg-3 col-md-4 col-sm-6" style="background-color: yellow; border: 1px solid;">
    col-lg-3
  </div>
  <div class="col-lg-3 col-md-4 col-sm-6" style="background-color: yellow; border: 1px solid;">
    col-lg-3
  </div>
</div>
```

Container: when we use container & we change size of display it is responsive to all but characters takes jump that looks bad and also not takes 100% width.

Container-fluid: It is same as container but takes 100% of width and when we change display sizes then it grab 100% width with no jumps (like fluid)

To move contents in the navbar to opp. Side of brand logo see section [Utilities > Spacing](#).

.navbar-expand (-sm, -md, -lg, -xl, -xxl) : this is used to create navbar horizontal instead of vertical.

col	col
col-6;	col-6
col-md-6	
col-md-6	

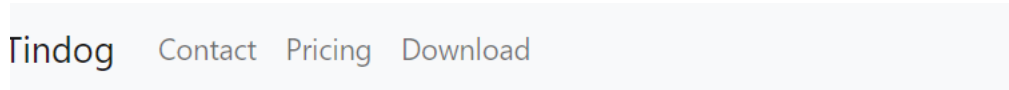
Ex. – if we use navbar-expand-lg then in laptop size screen navbar is horizontal but in tablets/mobiles it

is vertical navbar.

.bg-light – background of navbar. [Click here to know more →](#)

.navbar-brand – used to show name of company In the navbar it can be used with anchor tag.

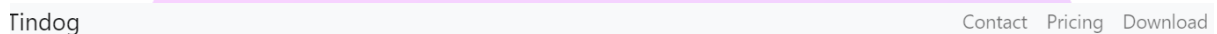
### Margin -



Tindog   Contact   Pricing   Download

To add space between brand logo and unordered list content

Margin-start-auto (ms-auto) to add space.



Tindog   Contact   Pricing   Download

Toggler : To add hamberger menu.

Carousel : To add slide show of images.

the carousel will avoid sliding when the webpage is not visible to the user.

The .active class needs to be added to one of the slides otherwise the carousel will not be visible.

.w-100 - set the width of an element to 100%

.d-block – sets display property to block.

data-bs-slide accepts the keywords prev or next, which alters the slide position relative to its current position. Alternatively, use data-bs-slide-to to pass a raw slide index to the carousel data-bs-slide-to="2", which shifts the slide position to a particular index beginning with 0.

The data-bs-ride="carousel" attribute is used to mark a carousel as animating starting at page load.

Aria-hidden= "true" by this screen readers skip particular part from reading

Ex. We use this in buttons so screen readers not read button left, button right..... (for visually impered persons)

Instead use <span class= "sr-only">Previous</span> to read "previous";

### Card:

.card-body - Use it whenever you need a padded section within a card.

Blockquote: Block quote in Bootstrap is used to present the another resource content or another website content or third party content in your website or portal. Block quote in bootstrap gives vertical line at the right side of the content. In bootstrap this block quote is declared within blockquote tag. We can also apply this block quote with buttons, paragraphs, images, headers, footers etc.

`<blockquote class="blockquote mb-0">` .....mb = 0 is margin-bottom = 0;

### Z-index:

There is x & y position on screen so we can move img up/down/left/right but there is also a z index by this we can move img towards you / away from you.

Child always sit on top of parent.

Things that come first in HTML will be right at the back and then, as we go down the HTML file, every other element becomes stacked on top of the previous elements.

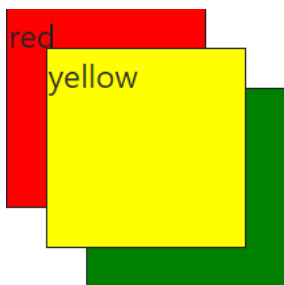
```

<div class="red">
  red
</div>

<div class="green">
  green
</div>

<div class="yellow">
  yellow
</div>

```



```

.red {
  background-color: red;
  border: 1px solid;
  height: 100px;
  width: 100px;
  position: absolute;
}

.yellow {
  background-color: yellow;
  border: 1px solid;
  height: 100px;
  width: 100px;
  position: absolute;
  left: 20px;
  top: 20px;
}

.green {
  background-color: green;
  border: 1px solid;
  height: 100px;
  width: 100px;
  position: absolute;
  left: 40px;
  top: 40px;
}

```

default z index is 0; z = +ve num above & z = -ve num is below;

remember that the z-index only works if your elements are positioned, so it has a positioning. (absolute, relative, fixed)

[Angela's Z-index structure PDF →](#)

Set object's position = absolute and other's position = relative

It allows us to maintain other's position and only change..

CSS Media Queries: for responsive website

Is your web page mobile friendly? [Check it out →](#) (official site by google)

If your website isn't optimized for mobile then your google search ranking decreases.

To make mobile site, either make separate site for mobile so user redirect to mobile site (Ex: m.facebook.com)

Or make it responsive

Media query breakpoints:

Structure-- @media <type> <feature>

```
h1 {  
  font size: 30px;  
}  
  
@media (min-width: 900px) {  
  h1 {  
    font-size: 60px;  
  }  
}
```

```
h1 {  
  font size: 30px;  
}  
  
@media (min-width: 900px) and (max-width: 1000px) {  
  h1 {  
    font-size: 60px;  
  }  
}
```

Why you've to use id?

```
<li class="nav-item">
  <a class="nav-link" href="#footer">Contact</a>
</li>
<li>
  <a class="nav-link" href="#pricing">Pricing</a>
</li>
<li>
  <a class="nav-link" href="#cta">Download</a>
```

When user clicks on navbar item it takes user to appropriate section

It works in url also.

```
OOTSTRAP/TINDOG/index.html#pricing|
```

Refactor your code:

Always remember -      Do not repeat yourself (Dry code)  
                                 & not "We Enjoy Typing" (wet code)

Make code as other person know what's going on.

We have to refactor our code on a regular basis in order to keep it performant and easy

If we refactor our code then it can easily modify/read by another member of our team.

- 1) Readability – any team member can easily gets what's going on.
- 2) Modularity - You would be able to narrow down on the exact section of code or code file that's responsible for the problems that's occurring.
- 3) Efficiency
- 4) Length

Priority → 1 > 2 > 3 > 4



if you're purely just reducing the length of your code and actually making your code less readable, less comprehensible, then that's bad.

If you write css and it is targeting entire elements & if you want to change one of these element property then you can able to write conditions

Ex class= "container-fluid"      id= "title"

```
.container-fluid {  
    padding: 10% 10%;  
}
```

```
#title .container-fluid {  
    padding: 7% 15%;  
}
```

It only applicable for container fluid class inside id title. & for others it is 10%.

Combining selectors:

Multiple selectors – ex

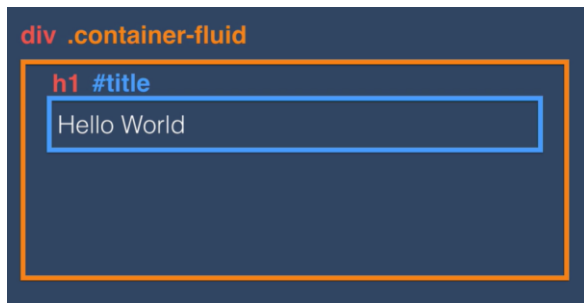
```
h1, h2, p {  
    background-color: grey;  
}
```

Hierarchical selectors –

```
Selector1 selector2 {  
  
}
```

Selector1 is from parent and selector2 is from child

Ex -



If write,

`div h1 {color: red;}` then it will color all h1 inside div's

Read from right to left. (all of the h1s that are contained inside a div.)

Instead if we write,

`.container-fluid #title {color: red;}` (element that has an id of title that is contained inside a parent) then it only changes color of #id's h1

`.container-fluid h1 {color: red;}` this is best.

In this color is change only for container fluid's h1.



Something has to be contained in something else.

Combined selectors : no space between two selectors

```
selector1#selector2 {
}
```

When there is no space you're reading from left to right

`div.container-fluid {color: red;}`

div that has a class of container-fluid should have the text color red.



```
div#title {color: red;}
```

this isn't work.

Priorities;

Html element < external css (classes < id's) < inline css

Use id's very sparingly .

Use it only for sections & very imp places

Don't use inline css