# 3. Bootstrap

## 2.1 Introduction to Bootstrap

Bootstrap is highly opinionated CSS styling framework. It basically provides bunch of read to use components like buttons, cards etc. which we can directly use in our website.

## *2.1.1 Bootstrap 5*

It is latest version of bootstrap, which supports all the major browsers except Internet explorer 11 and down word are not supported.

What new it brings to the table compare to Bootstrap 4 is, this version switched from ‘JQuery’ to ‘JavaScript’ and because of that,

* It has better performance, as now the size of JavaScript code needed to load is reduced.
* Better DOM interaction due to use of native JavaScript methods which are more efficient compare to JQuery.
* Added new components and utility classes.

## 2.2 Ways to use Bootstrap

### 2.2.1 include from ‘CDN (Content Delivery Network)

CDN are very group of servers located all around the world. Whenever we try to load data from CDN it first finds nearest server location and then loads data from that server.

For CSS:

<link

      href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"

      rel="stylesheet"

      integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"

      crossorigin="anonymous"

/>

For JavaScript:

<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/js/bootstrap.bundle.min.js"></script>

### 2.2.1 Download from Bootstrap website

### **2.2.1 Different types of CSS**

There are 3 ways for adding CSS into the HTML webpage. Those are following:

1. Inline CSS:

* Here, we can add CSS in element directly using ‘style’ attribute provided by html. For an example:

<header style="height: 11vh">

* This method has highest priority in then other methods.

1. Internal CSS

* In this method, we write CSS in ‘style’ of the HTML. For an example:

<style>

      /\* internal css \*/

      \* {

        padding: 0;

        margin: 0;

        box-sizing: border-box;

      }

</style>

* Usually, it is written above the body tag, but after HTML 5.1 technically we can write ‘style’ tag anywhere we want in entire document.

1. External CSS

* Here, we write CSS in separate document and then we attached that file into our webpage with the help of ‘link’ tag in head element. For an example:

<link rel="stylesheet" href="style.css" />

* Rel: this attribute indicates that linked document is stylesheet
* Href: it provides path to import css file.
* It is very important to note the extension of the document which is ‘.css’

When it comes to the priority order between internal and external CSS then it totally depends on the order in which they are imported and the specificity of the selector. If in both methods selector have same specificity then whichever defined later in the document will override other.

### 2.2.2 ‘!Important’ keyword

This keyword refers that field which it indicates that have ‘very high priority’ for an example:

P {

        font-size: 2.5rem !important;

        margin: 0;

        box-sizing: border-box;

  }

Here, font-size of element ‘p’ will be set to ‘2.5rem’ and it will not be override by any order of import or selector type (which means specificity). But only inline CSS can alter this behaviour because that have ‘highest priority’. See below:

<p style="font-size: 1rem;">Navneet</p>

Now, font-size of p will be set to ‘1rem’

### 2.2.3 priority order

Let’s see the priority order in detail:

1. Inline CSS: (highest priority)

* Irrespective of anything, inline CSS will always have highest priority compare to any other method or special keyword available.

1. !important keyword: (very high priority)

* Using this keyword, we can set priority of any property to very high. But Inline CSS can rule it out.
* Generally, it is recommended to not overused this keyword.

1. Specificity:

* Higher the specificity => higher the priority.
* To increase specificity, select the element more specifically like:

Specificity: 103

#formContainer form > div > div{

    display: flex;

    flex-direction: column;

}

Specificity: 001

div{

    width: 45%;

    height: 5vh;

}

1. Order of declaration

* If both selectors are having the same specificity and neither of them are using ‘!important’ keyword nor defined in inline CSS then their order of the occurrence in the document will define their priority.
* Lower the declaration higher the priority.

My Rule 2,3,4 together describes the priority order between internal and external CSS in detail here.

## 2.3 CSS Syntax

CSS is setoff properties and values which we can assign to any element by selecting it carefully. That means CSS have three important aspect.

1. Selector: To select the element from entire HTML page.
2. Property: Which defines the visual appearance of element like font-size, background-color etc.
3. Value: It refers to the values available corresponding to each element.

div{

    width: 45%;

    height: 5vh;

}

Here, div is the selector. Width and height are the properties. Whereas 45% and 5vh are their values corresponding to the properties.