

19/4/21

Introduction to CSS...

→ CSS → Cascading Style sheet

In order to style our pages we use CSS,
so that it looks beautiful.

- * Types of CSS → (1) Inline CSS
(2) Internal CSS
(3) External CSS.

(1) Inline CSS → This means that the style
can be applied inside the
particular html tag in order to reflect
the changes.

for ex →

```
<hi style="background-color: #red;  
           color: # white">  
Background </hi>.
```

→ The only problem we face in inline CSS is that it leads to repetition of code and makes it complex.

So, in order to replace this we use Internal CSS.

(2) Internal CSS. →

In this what happens is that we put all the styling in the head section by applying the properties to the selector we wish to see the changes on.

for ex →

<head>

[<style>

h1 {

background-color: # yellow;

color: # gray;

} margin-left: 20px;

</style>]

</head>

<body>

<h1> Background </h1>

</body>

NOTE → The Priority of inline CSS is the highest when compared to others.

for ex → If we have both inline and internal CSS applied to a single html tag, then the priority will be given to the inline CSS.

* User-agent stylesheet → This is something that is done by the browser itself for a particular element, and can be seen when we go to inspect element on the browser itself.

(3) External CSS → This is a separate page that can be used for multiple pages to apply the styling, in order to save time and efforts.

→ If we don't want our code to be too much messy and all the html and CSS code in a single page, then we use external CSS.

→ With an extension [.css]

→ In order to access a CSS page, we must link it with the HTML page in the head section.

for ex →

index.html

```
<head>
<style>
<link rel="stylesheet"
      href="index.css">
</head>
<body>
<h1> Background </h1>
</body>
```

index.css

```
h1 {
  background-color: #white;
  color: #black;
}

p {
```

* Margin And Padding :→

→ Margin is a CSS property that basically defines the space around an HTML element.

These are 4 types → margin-top: 30px;
margin-bottom: 40px;
margin-left: 50px;
margin-right: 30px;

So, do it more efficiently we can write it like →

```
[margin: top left bottom right;]
```

Page No. _____
Date. _____

margin: 50px 20px; .

For top & bottom For left & right.

Ex → margin: 30px 50px 20px 30px;

margin: 50px 20px 60px;

* Margin-Collapsing → top left and right bottom

Ex → h1 {

margin: 50px 30px 40px 30px;

p {

margin: 50px 20px 50px 20px;

}

From above what we understand is that, every element will be provided with this margin and every element will be shown on the browser with same.

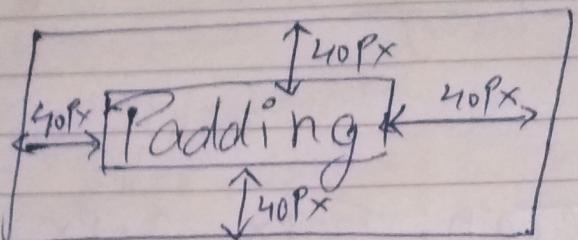
But what actually happens is →

The margins of the elements collapse with each other and reflects the margin which is bigger.

Suppose, margin-bottom of h1 = 40 and margin-top of p = 50 which should be equal to 90. but what it does it collapses and choose the margin as 50px as it is a greater no.

* Padding → Padding adds space between the boundaries of a container and the content present inside it.

Ex →



h1 {

padding: 40px;

We have,

padding - top,
padding - left,
padding - bottom,
padding - right.

* Short → [padding: top left bottom right]

— x —