

# Web Programming notes- 14th Oct

## CSS (Cascading Style Sheets)

### Class Selector:-

Used when few elements should have common CSS.

**<p class="red">Hello World</p>**

```
<html>
<head>
<style>
.red-text{
color:red;
}
h1{
color: blue;
}

</style>

</head>
<body>
<p class="red-text"><span>Hello</span> World</p>
<p class="blue-text">This is CSS example</p>
<p class="Orange-text">Lorem ipsum....</p>
<h1>Heading1</h1>
<h2>Heading2</h2>
<a href="https://drive.google.com/drive/my-drive">Visit Google Drive</a>
<a href=" " class="red-text">VisitAmazon</a>
</body>
</html>
```

### Id:

Used to identify the element uniquely.

## div:

div refers to partition (division) in a web page. It is considered as a container of HTML elements.

By default div tag has a width and height of its own, therefore div will occupy the width and height, same as the content.(i.e) the width and height of the div depends on the content.

The developer can define width and height by writing CSS.

```
<html>
<head>
<style>
    div{
        width=400px;
        height=400px;
        background-color: red;
        padding=40px;
        margin:30px;
        text-align=center;

    }
    white-text{
        color-white;
    }

</style>
<body>
    <div>
        <h1> This is a heading.</h1>
        <p>This is a dummy Paragraph</p>
    </div>
</body>
</html>
```

## Padding:

Will be used for applying spaces inside of any element.

Padding can be applied on 4 sides

1. Left- Spacing from the leftmost edge.

```
div{
    width=400px;
    height=400px;
```

```
background-color: red;
padding-left=30px;
}
```

2. Top- Spacing from the topmost edge.

```
div{
width=400px;
height=400px;
background-color: red;
padding-top=40px;
}
```

3. Right-Spacing from the right edge.

```
div{
width=400px;
height=400px;
background-color: red;
padding-right=30px;
}
```

4. Bottom - Spacing from the bottom most edge.

```
div{
width=400px;
height=400px;
background-color: red;
padding-bottom=50px;
}
```

### Shorthand property for padding:

padding=40px;

### Margin:

It is used to provide spacing outside of any element.

### Shorthand property for margin:

margin:30px => The margin of 30px will be applied **uniformly in all directions**.

Margin: 30px 50px => 30px will be applied to top and bottom while 50 px will be applied to right and left side. The values will be read in **CLOCKWISE** direction.

Margin: 30px 50px 40px 18px => 30px will be applied from the top, 50px will be applied from right, 40 px will be applied from the bottom and 18px will be applied from the left.

margin: 30px auto => Automatic adjustment of margin by the browser at the right and left.

## Login Page using HTML and CSS

```
<html>
<head>
<title>Welcome to Login</title>
<style>
    #container{
        width=500px;
        height=500px;
        margin=90px auto;
        text-align:center;
        border-radius=30px;
    }
    Body
    {
        //Background-color:linear-gradient(90 deg,red,purple);
        background: url('path');
    }
    h1{
        padding-top: 40px;
    }
    input{
        Width :70%;
        margin-bottom=20px;
        padding-top:12px;
        padding-bottom=12px;
    }
    #btn{
        width:40%;
    }

</head>
<body>
    <div id="container">
        <h1>Login</h1>
        <form>
            <input type="text" placeholder="Enter Username">
```

```

        <input type="password" placeholder="Enter Password">
        <input type="Submit" value="Login" id="btn">
    </form>
    <p Need Account?<a href="#">SignUp</a></p>
</div>
</body>
</html>

```

## Styling of background

As of now, a single color is applied on the background. But adding a mixture of colors on the background is also possible.

### Syntax:

**background: linear-gradient(shade1,shade2);**

## Types of CSS:

### 1.Internal CSS

- CSS written inside the **style tag of HTML**.
- Ex:
- <html>
- <head>
- <title>Welcome to Login</title>
- <style>
- #container{
- width=500px;
- height=500px;
- margin=90px auto;
- text-align:center;
- border-radius=30px;
- }
- Body
- {
- //Background-color:linear-gradient(90 deg,red,purple);
- background: url('path');
- }
- h1{
- padding-top: 40px;
- }
- input{
- Width :70%;

- margin-bottom=20px;
- padding-top:12px;
- padding-bottom=12px;
- }
- #btn{
- width:40%;
- }
- </style>
- </head>
- <body>
- <div id="container">
- <h1>Login</h1>
- <form>
- <input type="text" placeholder="Enter Username">
- <input type="password" placeholder="Enter Password">
- <input type="Submit" value="Login" id="btn">
- </form>
- <p Need Account?<a href="#">SignUp</a></p>
- </div>
- </body>
- </html>
- 

## 2. External CSS:

- CSS written **outside an HTML file**. Saved with **.css extension**.
- Linked to HTML file by using the link-  
**<link rel="stylesheet" type="text/css" href="style.css">**  
 Ex:

- #container{
- width=500px;
- height=500px;
- margin=90px auto;
- text-align:center;
- border-radius=30px;
- }
- Body
- {
- //Background-color:linear-gradient(90 deg,red,purple);
- background: url('path');
- }
- h1{
- padding-top: 40px;

- }
- input{
- Width :70%;
- margin-bottom=20px;
- padding-top:12px;
- padding-bottom=12px;
- }
- #btn{
- width:40%;
- }
- 

### 3. Inline CSS.

-CSS written **directly inside the specific tag of HTML file.**

Ex:

`<p style= "color: red";font-size: "20px;">Need an account?<a href="#">Sign Up</a></p>`

Inline CSS is not considered as a good practice for designing large web applications, since it makes the code difficult to read. It is best preferred when the requirement of styling is applied to a particular font or a small piece of text and paragraph.

### Preference for types of CSS:

**External CSS > Internal CSS > Inline CSS**

### Box Model :

**Entire Height= padding top + border-width + height + height of the content.**

**Entire Width= padding left + width + padding right.**

### Meta data:

Data about data is called metadata. It is **character encoded**.

Ex: **Search keyword, Character encoding, initial size.**

**<meta>** is used to provide metadata for any web page.

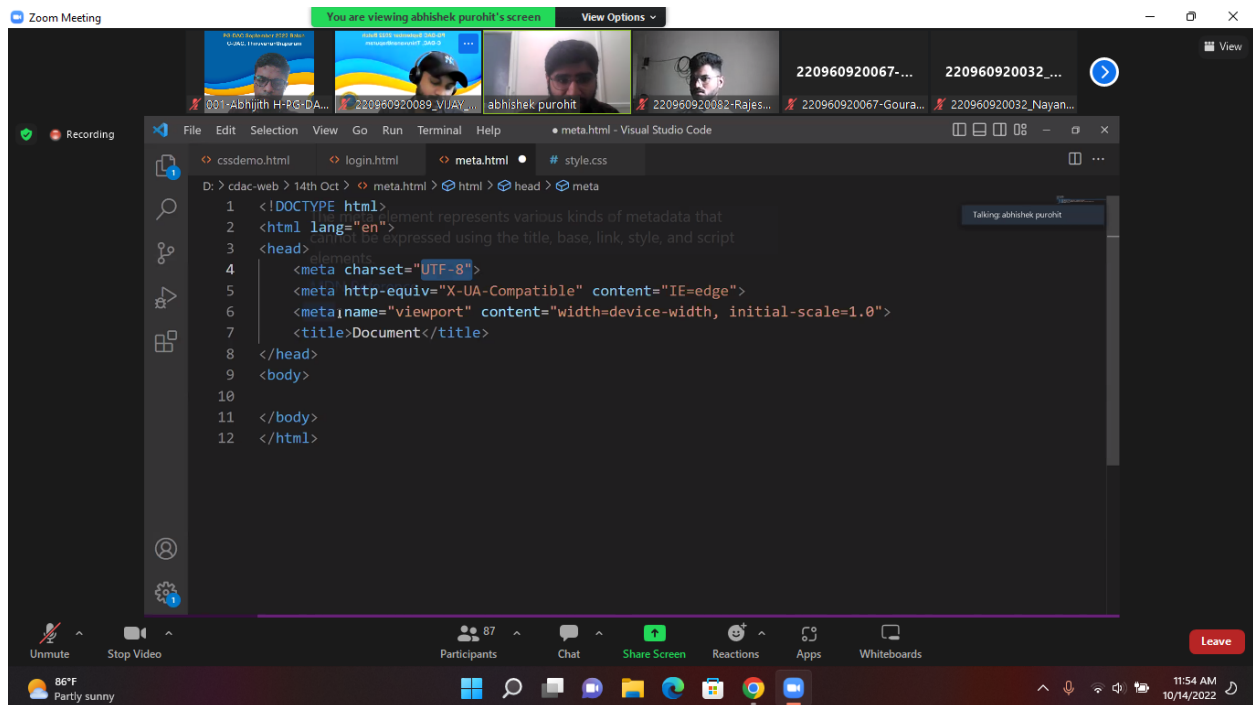
<meta> is given automatically by the IDE

It is always recommended to have the following meta tag on every webpage.

**<meta name="viewport" content="width=device-width , initial-scale=1.0">**

## Viewport:

It is the area which makes the browser's area available to display the content.



## Microdata:

It is a specification used to nest metadata within existing content on web pages. We want our search engines to not just look into meta tags only. We want our search engines to look into other html tags to search metadata. But preference is given to **metadata** over microdata.



Zoom Meeting

Recording

File Edit Selection View Go Run Terminal Help • meta.html - Visual Studio Code

cssdemo.html login.html meta.html style.css

D:\> cdac-web > 14th Oct > meta.html > html > body > p

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta keywords="best school, oldest">
8   <title>Document</title>
9 </head>
10 <body>
11   <h1>Best school in mumbai</h1>
12   <p>This is oldest school in mumbai which started right in early 90's</p>
13 </body>
14 </html>
```

Talking: abhishek purohit

abhishek purohit

Restricted Mode 0 0 0 This site is screen sharing. Stop Share 8 Spaces: 4 UTF-8 CRLF HTML

30°C Sunny

86°F Partly sunny

12:15 PM 10/14/2022

Zoom Meeting

You are viewing abhishek purohit's screen View Options

Recording

File Edit Selection View Go Run Terminal Help • meta.html - Visual Studio Code

cssdemo.html login.html meta.html style.css

D:\> cdac-web > 14th Oct > meta.html > html > body > h1

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Document</title>
8 </head>
9 <body>
10   <h1>Best school in mumbai</h1>
11 </body>
12 </html>
```

Talking: abhishek purohit

abhishek purohit

Unmute Stop Video Participants Chat Share Screen Reactions Apps Whiteboards Leave

85

86°F Partly sunny

12:05 PM 10/14/2022

Zoom Meeting

Recording

tutorialspoint

example

- To create an item, the **itemscope** attribute is used.
- To add a property to an item, the **itemprop** attribute is used on one of the item's descendants.

Here there are two items, each of which has the property "name" -

```
<html>
<body>

<div itemscope>
  <span name is <span itemprop = "name">Zara</span></div>

<div itemscope>
  <span name is <span itemprop = "name">Nihar</span></div>

</body>
</html>
```

It will produce the following result -

My name is Zara.

86°F Partly sunny

12:17 PM 10/14/2022

Zoom Meeting

Recording

meta.html - Visual Studio Code

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta keywords="best school, oldest">
8   <title>Document</title>
9 </head>
10 <body>
11   <div itemscope>
12     <h1 itemprop = "search keyword">Best school in mumbai</h1>
13     <p>XYZ school is oldest school in mumbai which started right in early 90's</p>
14   </div>
15
16 </body>
17 </html>
```

86°F Partly sunny

12:18 PM 10/14/2022

Zoom Meeting

Recording

Visual Studio Code - meta.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <meta keywords="best school, oldest">
8   <title>Document</title>
9 </head>
10 <body>
11   <div itemscope>
12     <h1 itemprop="search keyword">Best school in mumbai</h1>
13     <p><span itemprop="school">XYZ school</span> is oldest school in mumbai which starte
14   </div>
15
16 </body>
17 </html>
```

86°F Partly sunny

12:20 PM 10/14/2022

abhishek purohit