

## Paper Code: CSCS351    **WEB TECHNOLOGY**

**MODULE – III Introduction to DHTML – Introduction to style sheets – Setting the default style sheet language – Inline style information – External Style sheets – Cascading Style sheets.**

**Cascading Style Sheets**, referred to as **CSS**, is a simple design language intended to simplify the process of making web pages presentable.

**CSS** is a **MUST** for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. The following key advantages present in **CSS**:

- **Create Stunning Web site** - **CSS** handles the look and feel part of a web page. Using **CSS**, you can control the color of the text, the style of fonts, the spacing between paragraphs, how columns are sized and laid out, what background images or colors are used, layout designs, variations in display for different devices and screen sizes as well as a variety of other effects.
- **Become a web designer** - If you want to start a career as a professional web designer, **HTML** and **CSS** designing is a must skill.
- **Control web** - **CSS** is easy to learn and understand but it provides powerful control over the presentation of an **HTML** document. Most commonly, **CSS** is combined with the markup languages **HTML** or **XHTML**.
- **Learn other languages** - Once you understand the basic of **HTML** and **CSS** then other related technologies like javascript, php, or angular are become easier to understand.

Hello World using **CSS**.

```
<!DOCTYPE html>
<html>
  <head>
    <title>This is document title</title>
    <style>
      h1 {
        color: #36CFFF;
      }
    </style>
  </head>
  <body>
    <h1>Hello World!</h1>
  </body>
</html>
```

Applications of **CSS**

As mentioned before, **CSS** is one of the most widely used style language over the web. I'm going to list few of them here:

- **CSS saves time** - You can write **CSS** once and then reuse same sheet in multiple **HTML** pages. You can define a style for each **HTML** element and apply it to as many Web pages as you want.
- **Pages load faster** - If you are using **CSS**, you do not need to write **HTML** tag attributes every time. Just write one **CSS** rule of a tag and apply it to all the occurrences of that tag. So less code means faster download times.

- **Easy maintenance** - To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.
- **Superior styles to HTML** - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.
- **Multiple Device Compatibility** - Style sheets allow content to be optimized for more than one type of device. By using the same HTML document, different versions of a website can be presented for handheld devices such as PDAs and cell phones or for printing.
- **Global web standards** - Now HTML attributes are being deprecated and it is being recommended to use CSS. So its a good idea to start using CSS in all the HTML pages to make them compatible to future browsers.

CSS comprises of style rules that are interpreted by the browser and then applied to the corresponding elements in your document. A style rule is made of three parts –

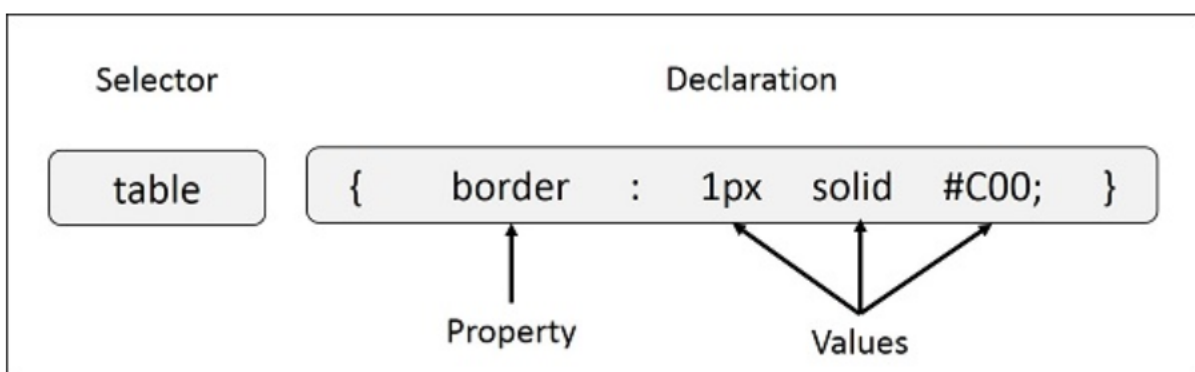
**Selector** – A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.

**Property** – A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border etc.

**Value** – Values are assigned to properties. For example, color property can have value either red or #F1F1F1 etc.

#### CSS Syntax

```
selector { property: value }
```



Example – We can define a table border as follows –

```
table{ border :1px solid #C00; }
```

Here table is a selector and border is a property and given value 1px solid #C00 is the value of that property.

We can define selectors in various simple ways based on your comfort. Let me put these selectors one by one.

## The Type Selectors

This is the same selector we have seen above. Again, one more example to give a color to all level 1 headings –

```
h1 {  
    color: #36CFFF;  
}
```

## The Universal Selectors

Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type –

```
* {  
    color: #000000;  
}
```

This rule renders the content of every element in our document in black.

## The Descendant Selectors

Suppose you want to apply a style rule to a particular element only when it lies inside a particular element. As given in the following example, style rule will apply to <em> element only when it lies inside <ul> tag.

```
ul em {  
    color: #000000;  
}
```

## Setting the default style sheet language

Very often Web creators are using an external style sheet, or a style element to add style information to their html pages. By doing, we specify what is the style language used in the Web page. For example using the `link` element.

```
<link href="cute.css" rel="stylesheet" type="text/css"/>
```

But if your page is using the style attribute, the user agent may not know what is the language used for styling. This is a principle of orthogonality. A company could launch a user agent with a new style language and a new mime-type. It will be difficult to impose and ensure interoperability because of the deployment base, but it is still possible.

If you want to set up a default style sheet language for your Web site, there are a few choices. You can use a meta name in each of your page:

```
<meta http-equiv="Content-Style-Type" content="text/css"/>
```

or you can specify an HTTP header:

Content-Style-Type: text/css

## Inline CSS Information

An inline CSS is used to apply a unique style to a single HTML element.

An inline CSS uses the style attribute of an HTML element.

The following example sets the text color of the <h1> element to blue, and the text color of the <p> element to red:

Example

```
<h1 style="color:blue;">A Blue Heading</h1>
```

```
<p style="color:red;">A red paragraph.</p>
```

## Internal Style Sheet (or) Embedded CSS - The <style> Element

We can put your CSS rules into an HTML document using the <style> element. This tag is placed inside the <head>...</head> tags. Rules defined using this syntax will be applied to all the elements available in the document. Here is the generic syntax –

```
<!DOCTYPE html>
<html>
  <head>
    <style type = "text/css" media = "all">
      body {
        background-color: linen;
      }
      h1 {
        color: maroon;
        margin-left: 40px;
      }
    </style>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
  </body>
</html>
```

It will produce the following result –

**This is a heading**

This is a paragraph.

## External CSS

An external style sheet is used to define the style for many HTML pages.

To use an external style sheet, add a link to it in the <head> section of each HTML page:

### Example

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="style1.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

"style1.css":

```
body {
  background-color: powderblue;
}
h1 {
  color: blue;
}
p {
  color: red;
}
```

### CSS Colors, Fonts and Sizes

Here, we will demonstrate some commonly used CSS properties. You will learn more about them later.

The CSS color property defines the text color to be used.

The CSS font-family property defines the font to be used.

The CSS font-size property defines the text size to be used.

Example

Use of CSS color, font-family and font-size properties:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  color: blue;
  font-family: verdana;
  font-size: 300%;
}
p {
  color: red;
  font-family: courier;
  font-size: 160%;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body></html>
```