Module 3: Cascading Style Sheet-CSS

Module Overview

In this module, students will be able to learn about frontend technologies CSS3.

Module Objective

At the end of this module, students should be able to demonstrate appropriate knowledge, and show an understanding of the following:

- Learn CSS role in creating user interfaces (for mobile) and websites.
- Learn the basic CSS concepts: selectors, CSS properties, CSS code structure, CSS declarations, CSS unit types etc.
- Learn how CSS is included in an HTML page.

Introduction to CSS

Introduction to CSS

CSS stands for Cascading Style Sheets. It is a simple design language intended to simplify the process of making web pages presentable. 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, as well as a variety of other effects. CSS works with HTML and other Markup Languages (such as XHTML and XML) to control the way the content is presented. Cascading Style Sheets is a means to separate the appearance of a webpage from the content of a webpage.

Definition

Cascading Style Sheets (CSS) is a simple mechanism used to format the layout of Web Pages and add style (e.g., fonts, colors, spacing) to web documents that previously could only be defined in a page's HTML. CSS describes how HTML elements are to be displayed on the screen, paper, or in other media. It can control the layout of multiple web pages all at once.

What is the "Cascade" part of CSS?

The cascade part of CSS means that more than one style sheet can be attached to a document, and all of them can influence the presentation. For example, a designer can have a global style sheet for the whole site, but a local one for say, controlling the link color and background of a specific page. Or, a user can use their style sheet if s/he has problems seeing the page, or if s/he just prefers a certain look.

Difference between HTML and CSS

HTML and CSS are the basic and important web scripting languages, the primary use of which is to design web pages and web applications. The initial and important difference between the two is that HTML is used for the creation of the webpages and CSS is used to control the styling and layout of web pages.

In HTML, firstly you write words and then add elements or tags to it, which thereafter appear on your page. In this way, the browser gets to know the heading of the page, the beginning and end of the paragraph, and so on.

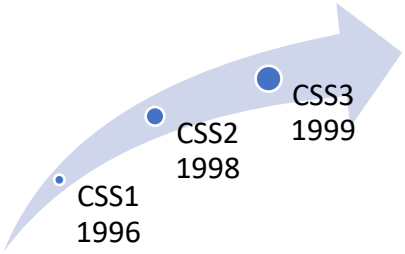
In CSS, rules are used by utilizing CSS properties. CSS properties are generally classified into two broad categories. First is the presentation which specifies the color of the text, font type, font size, background colors, background images, etc. The second is layout defines the position of the different elements on the screen

BASIS FOR COMPARISON	HTML	CSS
Basic	Dictates content and structure of the web pages.	Modifies the design and display of the HTML elements.
Relevance	CSS can be used in HTML files.	HTML cannot be used in CSS style sheets.
Consist of	Tags surrounding content.	Selectors succeeded by a declaration block.
Methods of using	There are no defined methods.	Inline CSS code, Internal and external stylesheets any method could be used for implementing the code.

Key Differences Between HTML and CSS

- ***HTML*** *is the basic markup language that describes the content and structure of the web pages. On the other hand,* ***CSS*** *is the extension to the* ***HTML*** *which modifies the design and display of the web pages.*
- ***HTML*** *file can contain* ***CSS*** *code while* ***CSS*** *stylesheets can never contain* ***HTML*** *code in it.*
- ***HTML*** *comprises tags surrounding content. Whereas* ***CSS*** *comprised of selectors succeeded by a declaration block*.

CSS Timeline



The CSS started in 1994, at a time when electronic publishing was fast catching on. Defining layout and styles with HTML was not always possible hence, the need for a styling language arose. CSS was created by HåkonWium Lie who worked at CERN with Tim Berners Lee - the father of the web After much struggle, it was finally accepted as a W3C standard in 1996 which is named CSS1. From then on CSS has evolved continuously into CSS2 in 1998 and CSS3 in 1999.

CSS Syntax

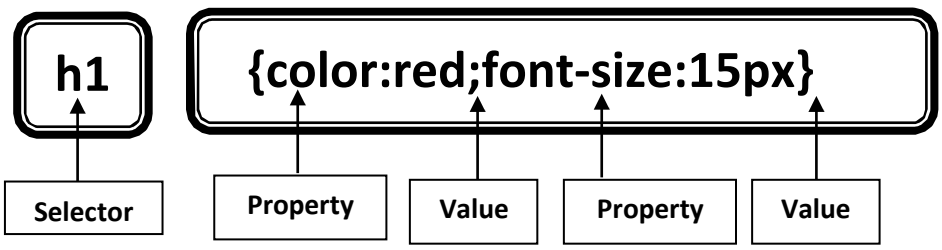
A CSS style rule is made of three parts:

1. ***Selector:*** *A selector is an HTML tag at which a style will be applied. This could be any tag like <h1>, <p> or <table> etc.*
2. ***Property:*** *A property is a type of attribute of an HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be color, border, bgcolor, etc.*
3. ***Value:*** *Values are assigned to properties. For example, the color property can have the value either red or #F1, F1F1, etc.*

The format or syntax of CSS is:

```
Selector {property:
```

Example: You can define a heading as follows:



Here h1 is a selector, color and font size are properties, the given value is red , and 1is the value of that property.

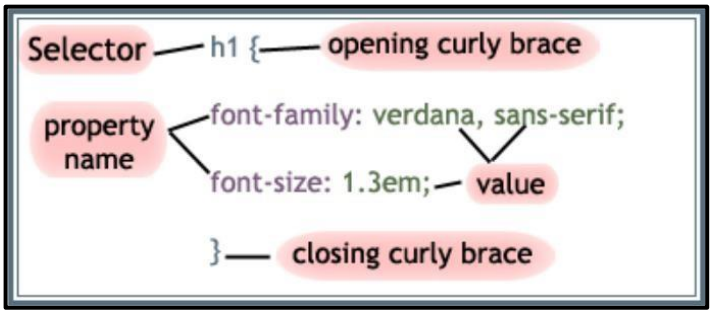
- The selector is normally the HTML element you want to style.
- Each declaration consists of a property and a value.
- The property is the style attribute you want to change. Each property has a value.

Rules/Principles of CSS

1. Every statement must have a selector and a declaration. The declaration comes immediately after the selector and is contained in a pair of curly braces.
2. The declaration is one or more properties separated by semicolons.
3. Each property has a property name followed by a colon and then the value for that property. There are many different types of values, but any given property can only take certain values as set down in the specification.
4. Sometimes a property can take several values, as in the font family. The values in the list should be separated by a comma and a space.
5. Sometimes a value will have a unit as well as the actual value, as in the 1.3em. You must not put a space between the value and its unit.
6. As with HTML, white space can be used to make your style sheet easier to read and write.

Parts of the style sheet

A style sheet consists of one or more rules that describe how to document elements that should be displayed. A rule in CSS has two parts: the selector and the declaration. The declaration also has two parts, the property, and the value. Let's take a look at a rule for a heading 1 style: h1{font-family Verdana , "sans serif"; font-size: 1.3em} This expression is a rule that says every h1 tag will be Verdana



or other sans-serif font and the font size will be 1.3em. Let's take a look at the different parts of this rule.

Selector
{
property1: some value;
property2: some value;
}

The declaration contains the property and value for the selector. The property is the attribute you wish to change and each property can take a value. The property and value are separated by a colon and surrounded by curly braces:

body { background-color: black}

If the value of a property is more than one word, put quotes around that value: body { font-family: "sans serif"; } If you wish to specify more than one property, you must use a semi-colon to separate each property. This rule defines a paragraph that will have blue text that is centered.

p { text-align: center; color: blue }

You can group selectors. Separate each selector with a comma. The example below groups headers 1, 2, and 3 and makes them all yellow. h1, h2, h3 { color: yellow}

The browser or other software that is capable of interpreting HTML, such as word processors. The basic structure of an HTML document

Advantages of CSS

The advantages of CSS are:

- **CSS saves time -** You can write CSS once and then reuse the same sheet in multiple HTML pages.
- **Pages load faster** – Increases Download Speed
- **Easy maintenance** - To make a global change, all the elements in all the web pages will be updated automatically.
- **Superior styles to HTML** – It isa 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.
- **Global web standards** - Now HTML attributes are being deprecated and it is being recommended to use CSS

Exercise

The trainer will initiate a discussion of common questions on CSS as given below:

1. **Write the three parts of CSS syntax?**

A. Selector, Property, Value
B. Frame, Text, Color
C. Head, Body, HTML
D. Font, Color, Style

2. **What are the advantages of CSS?**

CSS Selectors

CSS Selectors

A CSS selector is the initial segment of a CSS Rule. It is an example of components and different terms that let the program knows which HTML components ought to be chosen to have the CSS property estimations inside the standard applied to them. The component or components which are chosen by the selector are alluded to as the subject of the selector.

You can define selectors in various simple ways based on your comfort. Let me put these selectors one by one. Three types of CSS Selectors are,

1. *The Element selectors*
2. *The ID Selectors*
3. *The Class Selectors*

Element Selectors

A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly brackets: For example -

p {color:red;text-align:center;}

To make the CSS more readable, you can put one declaration on each line, like this:

```
p
{
color:red;
text-align:center;
}
```

ID Selector

The id selector is used to specify a style for a single, unique element.The id selector uses the id attribute of the HTML element, and is defined with"#". Example –

Imagine within the body element of our HTML page, we have the following paragraph element

*** Welcome to the 1st CSS Document ***

We can then create a CSS rule with the id selector:

```
#Welcome
{
color:red;
text-align:center;
}
```

Class Selector

The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements. This allows you to set a particular style for many HTML elements with the same class. The class selector uses the HTML class attribute, and is defined with a ".". In the example below, all HTML elements with class="center" will be center-aligned:

Imagine within the body element of our HTML page, we have the following header element

<h2 class=" center">Summary</h2>

We can then create a CSS rule with the class selector:

. center {text-align:center;}

You can also specify that only specific HTML elements should be affected by a class. In the example below, all p elements with class="center" will be center-aligned: example

p.center {text-align:center;}

Universal selector

An asterisk (*) is the universal selector for CSS. It matches a single element of any type. Omitting the asterisk with simple selectors has the same effect. For instance, *. warning and. warning are considered equal. Rather than selecting elements of a specific type, the universal selector quite simply matches the name of any element type: For example-

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

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

Attribute Selector

You can also apply styles to HTML elements with particular attributes. The style rule below will match all the input elements having a type attribute with a value of text:

```
input[type="text"]
{
color: #000000;}
```

The advantage of this method is that the <input type="submit" /> element is unaffected, and the color is applied only to the desired text fields. Thereare following rules applied to attribute selector.

- p[lang] - Selects all paragraph elements with a lang attribute.
- p[lang="fr"] - Selects all paragraph elements whose lang attribute has a value of exactly "fr".
- p[lang~="fr"] - Selects all paragraph elements whose lang attribute contains the word "fr".

- p[lang|="en"] - Selects all paragraph elements whose lang attribute contains values that are exactly "en", or begin with "en-".

Exercise

The trainer will initiate a discussion of common questions on CSS Selector as given below:

1. What are the three types of selectors?
2. What is a universal selector?
3. p[lang] is which type of selector?

Embedding CSS to HTML

Embedding CSS to HTML

If you're building a website, then you'll start with HTML. With this markup language, you can add headings,paragraphs, images, tables, forms, lists, and much more. But you can't control how these elements are presented or laid out on the page.That's where CSS comes in.

CSS describes how a page should look to the browser, which renders it accordingly. CSS can be used for a wide variety of stylistic purposes, including changing text and background color on a page, removing the underline from links, and animating images, text, and other HTML elements.

How to Add CSS to HTML?

There are three ways to add CSS to HTML. You can add inline CSS in a style attribute to style a single HTML element on the page. You can embed an internal stylesheet by adding CSS to the head section of your HTML doc. Or you can link to an external stylesheet that will contain all your CSS separate from your HTML.

If you want greater control over the appearance of your site, then you need to know how to add CSS to your site.

External Style Sheet

How to Add an External CSS File to HTML

External CSS is formatted like internal CSS, but it isn't wrapped in <style> tags or placed in the head section of your HTML file. Instead, it's placed in an external file with the extension ".css." In the head section, you'll just have to add a link to this external stylesheet that looks something like:

<link rel="stylesheet" type="text/css" rel="noopener" target="_blank" href="mystyles.css">

Using external CSS is considered the best practice for a few reasons.

Since you can make changes across your site by changing the CSS in this external file, it's the most time-effective method. It's also the fastest and most SEO-friendly. Storing CSS in another file makes your HTML file easier to read for search engines. It also enables a visitor's browser to cache the CSS file to load your website faster for their next visit.

External CSS Example

Let's use external CSS to style a div in HTML.

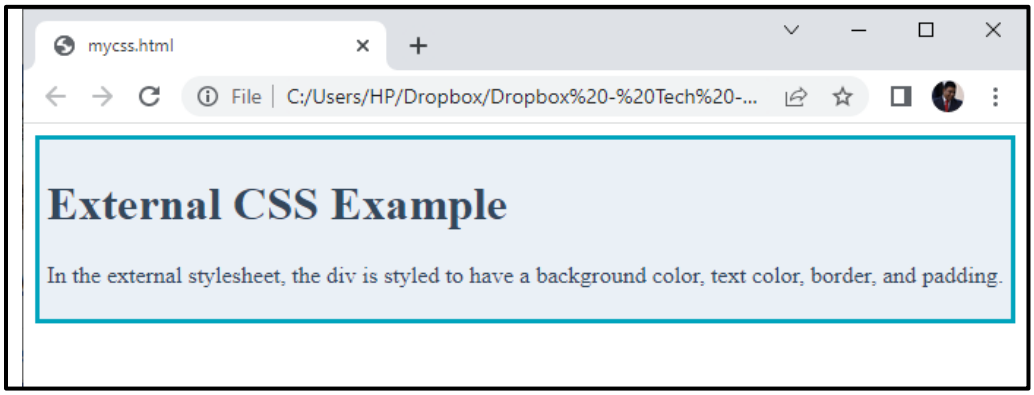
Example

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" rel="noopener" target="_blank" href="mystyle.css">
</head>
<body>
<div>
<h1>External CSS Example</h1>
<p>In the external stylesheet, the div is styled to have a background color, text color, border, and padding.</p>
</div>
</body>
</html>
```

Here's how the mystyle.css file would look:

```
div {
  background-color: #EAF0F6;
  color: #33475B;
  border: 3px solid #00A4BD;
  padding: 5px;
}
```

Output



Internal Style Sheet

How to Add Internal CSS to HTML

Internal CSS looks different from inline CSS. A CSS property and value is still set, but instead of being placed inside a style attribute, it is placed inside brackets and defined by a CSS selector. This rule set is then wrapped in <style></style> tags and found in the head section of the HTML file.

```
<!DOCTYPE html>
<html>
<head>
<style>
selector {
CSS property: value;
}
</style>
</head>
```

Using internal CSS is considered better practice than using inline CSS.
Internal CSS allows you to style groups of elements at once — rather than having to add the same style attributes to elements over and over again.

Also, since it separates the CSS and HTML into different sections but keeps it in the same document, internal CSS is ideal for one-page websites. If you have a multi-page site and would like to make changes across your site, you'll have to open up each HTML file representing those pages and add or change the internal CSS in each head section. (Or you can use external CSS).

Internal CSS Example

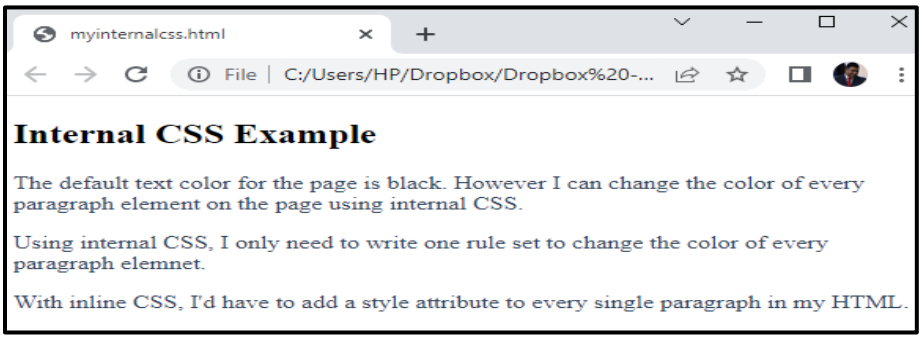
Let's say you want to change the text color of every paragraph element on a web page to a navy blue color. In that case, you'd set the color property to the hex color code for a shade of navy (#33475B), place it within a CSS rule set with the type selector p, and place the whole thing inside the head section of the web page.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  color: #33475B;
}
</style>
</head>
<body>
```

```
<h2>Internal CSS Example</h2>
<p>The default text color for the page is black. However, I can change the color of every paragraph element on the page using internal CSS.</p>
<p>Using internal CSS, I only need to write one rule set to change the color of every paragraph element.</p>
<p>With inline CSS, I'd have to add a style attribute to every single paragraph in my HTML.</p>
</body>
</html>
```

Output:



Inline Style Sheet

Requires the style attribute placed inside an HTML element.

How to Add Inline CSS to HTML

Inline CSS allows you to put CSS "in" HTML. To add inline CSS, you use a style attribute and place it inside the opening tag of an HTML element. Here's the syntax:

<element style="CSS property: value">

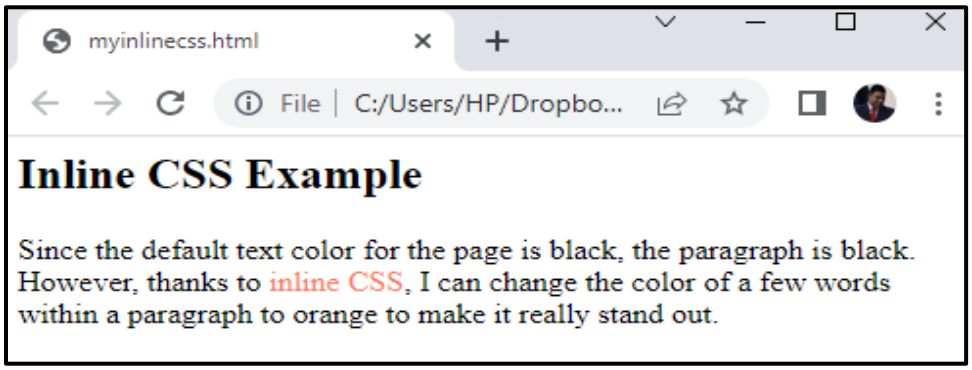
Inline CSS, otherwise known as the “embedded stylesheet,” will override any other CSS targeting the same elements. Since it's the closest to HTML, browsers determine inline CSS declarations are the most relevant to the HTML element and should be applied. For this reason, inline CSS is effective for targeting a single element with unique style properties — but it should be avoided when it's possible to use internal or external CSS.

Inline CSS Example

<h2>Inline CSS Example</h2>

* Since the default text color for the page is black, the paragraph is black. However, thanks to inline CSS, I can change the color of a few words within a paragraph to orange to make it really stand out. *

Let's say you want to change the color of a key term in a paragraph to a bright orange and leave the other text as is. To start, you'd wrap the key term in span tags. Then, set the color property to the hex color code for a shade of orange (#FF7A59), place it within a style attribute, and place the whole thing inside the opening tag. Check it out below.



Background image handling

The background-image property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element. The background image for a page can be set like this: body {background- image:url('paper.gif');}

Example

```
<html>
<head>
<Title>This is my Internal css page</Title>
<style type="text/css"> body
{
background-image:url ("C:/Users/SAI/Desktop/Desktop/100MSDCF/11. jpg");
}
</style>
</head>
<body>
Background Image
</body>
</html>
```

Output:

You can set the following background properties of an element:

- The **background-image property** is used to set the background image of an element.
- The **background-repeat property** is used to control the repetition of an image in the background.
- The **background-position property** is used to control the position of an image in the background.
- The **background-attachment property** is used to control the scrolling of an image in the background.
- The background property is used as shorthand to specify a number of other background properties.

Exercise

The trainer will initiate a discussion of common questions on CSS Selector as given below:

1. **How many ways to build style sheets?**
2. **What are the background properties?**
3. **What are the differences between Internal and inline style sheet**

Formatting Fonts

Formatting Fonts

A font is the combination of typeface and other qualities, such as size, pitch, and spacing. For example, Times Roman is a typeface that defines the shape of each character. Within Times Roman, however, there are many fonts to choose from -- different sizes, italic, bold, and so on. You can set the following font properties of an element:

- The font-family property is used to change the face of a font.
- The font-style property is used to make a font italic or oblique.
- The font-variant property is used to create a small-caps effect.
- The font-weight property is used to increase or decrease how bold or light a font appears.
- The font-size property is used to increase or decrease the size of a font.
- The font property is used as shorthand to specify a number of other font properties.

CSS Fonts

Selecting the right font has a huge impact on how the readers experience a website.The right font can create a strong identity for your content.Using a font that is easy to read is important. The font adds value to your text. It is also important to choose the correct color and text size for the font.

Browsers do not support all the fonts, so you need to use multiple fonts to be on safer side.

CSS font-family defines the priority for the browser to choose the font from multiple fonts.

There are 2 types of font families which you can use –

Font-Family – This is a specific type of font like Arial, Verdana, Tahoma

Generic Font-Family – This is a General Font and almost all browsers support this generic font family. Example: serif, Sans-serif etc

CSS Font Family

It is the name of the font-family such as "Courier", "Arial", "Times", etc.

The font-family property should hold several font names as a "fallback" system, to ensure maximum compatibility between browsers/operating systems. Start with the font you want, and end with a generic family (to let the browser pick a similar font in the generic family, if no other fonts are available). The font names should be separated with comma.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
.p1 {
font-family: "Times New Roman", Times, serif;
}
.p2 {
  font-family: Arial, Helvetica, sans-serif;
}
.p3 {
font-family: "Lucida Console", "Courier New", monospace;
}
</style>
</head>
<body>
<h1>CSS font-family</h1>
<p class="p1">This is a paragraph, shown in the Times New Roman font.</p>
```

```
<p class="p2">This is a paragraph, shown in the Arial font.</p>
<p class="p3">This is a paragraph, shown in the Lucida Console font.</p>
</body>
</html>
```

Output:

CSS Generic Font Families

In CSS there are five generic font families:

- **Serif** fonts have a small stroke at the edges of each letter. They create a sense of formality and elegance.
- **Sans-serif** fonts have clean lines (no small strokes attached). They create a modern and minimalistic look.
- **Monospace fonts** - here all the letters have the same fixed width. They create a mechanical look.
- **Cursive fonts** imitate human handwriting.
- **Fantasy** fonts are decorative/playful fonts.

Difference between Serif and Sans-serif Fonts

Generic Font Family	Examples of Font Names
Serif	Times New Roman Georgia Garamond
Sans-serif	Arial Verdana Helvetica

Monospace	Courier New Lucida Console Monaco
Cursive	Brush Script MT Lucida Handwriting
Fantasy	COPPERPLATE PAPYRUS

CSS Font Style

CSS Font style property defines what type of font you want to display. It may be italic, oblique, or normal.

CSS Syntax

font-style: normal|italic|oblique|initial|inherit;

Property Values

- **normal** The browser displays a normal font style. This is default
- **italic** The browser displays an italic font style
- **oblique** The browser displays an oblique font style
- **initial** Sets this property to its default value. Read about initial
- **inherit** Inherits this property from its parent element. Read about inherit

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.a {
font-style:normal;
}
p.b {
font-style:italic;
}
p.c {
font-style:oblique;
}
</style>
</head>
<body>
<h1>The font-style Property</h1>
<p class="a">This is a paragraph, normal.</p>
<p class="b">This is a paragraph, italic.</p>
<p class="c">This is a paragraph, oblique.</p>
</body>
</html>
```

Output:

CSS Font Weight

The font-weight property sets how thick or thin characters in text should be displayed.

CSS Syntax

font-weight: normal|bold|bolder|lighter|number|initial|inherit;

Property Values

- **normal** Defines normal characters. This is default
- **bold** Defines thick characters
- **bolder** Defines thicker characters
- **lighter** Defines lighter characters
- **100-900** Defines from thin to thick characters. 400 is the same as normal, and 700 is the same as bold
- **initial** Sets this property to its default value. Read about initial
- **inherit** Inherits this property from its parent element. Read about inherit

Example:

Set different font weight for three paragraphs:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {
  font-weight: normal;
}
p.light {
font-weight: lighter;
}
p.thick {
```

```
font-weight: bold;
}
p.thicker {
font-weight: 900;
}
</style>
</head>
<body>
<h1>The font-weight Property</h1>
<p class="normal">This is a paragraph.</p>
<p class="light">This is a paragraph.</p>
<p class="thick">This is a paragraph.</p>
<p class="thicker">This is a paragraph.</p>
</body>
</html>
```

Output

CSS Font Size

This property of font is to set size of nay font, it can be define with pixels, percentages or by size key word (small, medium, large)

CSS Syntax

font-size:medium|xx-small|x-small|small|large|x-large|xx-large|smaller|larger|length|initial|inherit;

Property Values

- **MediumSets** the font-size to a medium size. This is default
- **xx-smallSets** the font-size to an xx-small size
- **x-smallSets** the font-size to an extra small size
- **small Sets** the font-size to a small size
- **large Sets** the font-size to a large size
- **x-large Sets** the font-size to an extra large size
- **xx-large Sets** the font-size to an xx-large size
- **smaller Sets** the font-size to a smaller size than the parent element

- **larger Sets** the font-size to a larger size than the parent element
- **length Sets** the font-size to a fixed size in px, cm, etc. Read about length units
- **% Sets** the font-size to a percent of the parent element's font size
- **Initial Sets** this property to its default value. Read about initial
- **inherit** Inherits this property from its parent element. Read about inherit

Example

Set the font size for different elements:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.a {
font-size: 15px;
}
div.b {
font-size: large;
}
div.c {
font-size: 150%;
}
</style>
</head>
<body>
<h1>The font-size Property</h1>
<div class="a">This is some text.</div>
<div class="b">This is some text.</div>
<div class="c">This is some text.</div>
</body>
</html>
```

Output:

CSS Icons

Icons can be defined as the images or symbols used in any computer interface refer to an element. It is a graphical representation of a file or program that helps the user to identify about the type of file quickly.Using the icon library is the easiest way to add icons to our HTML page. It is possible to format the library icons by using CSS. We can customize the icons according to their color, shadow, size, etc.

There are 3 types of icon libraries available, namely

- *Font Awesome Icons*
- *Google Icons*
- *Bootstrap Icons*

Font Awesome icons Syntax:

To use this library in our HTML page, we need to add the following link within the **<head></head>** section.

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">

Example

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Icons</title>
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/4.7.0/css/font-awesome.min.css">
<style>
body{
text-align:center;
background-color:lightblue;
}
.fa{
color:red;
font-size:50px;
margin:10px;
}
</style>
</head>
<body style="text-align:center">
<h1>Font Awesome Library</h1>
<iclass="fa fa-cloud"></i>
<iclass="fa fa-file"></i>
<iclass="fa fa-heart"></i>
<iclass="fa fa-bars"></i>
<iclass="fa fa-car"</i>
```

```
</body>
</html>
```

Output:

Bootstrap icons Syntax:

As similar to the font awesome library, we can add the bootstrap icons in our HTML page using the link given below in the **<head></head>** section.

<link rel="stylesheet"

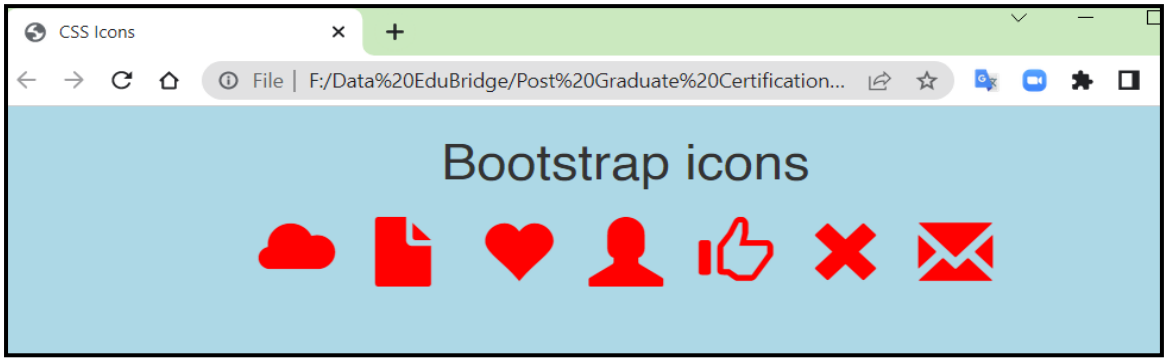
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Icons</title>
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">
<style>
body{
text-align:center;
background-color:lightblue;
}
.glyphicon{
color:red;
font-size:50px;
margin:10px;
}
</style>
</head>
<body style="text-align:center">
```

```
<h1>Bootstrap icons</h1>
<i class="glyphiconglyphicon-cloud"></i>
<i class="glyphiconglyphicon-file"></i>
<i class="glyphiconglyphicon-heart"></i>
<i class="glyphiconglyphicon-user"></i>
<i class="glyphiconglyphicon-thumbs-up"></i>
<i class="glyphiconglyphicon-remove"></i>
<i class="glyphiconglyphicon-envelope"></i>
</body>
</html>
```

Output:



Google icons Syntax:

As similar to the above libraries, we can add it in our HTML page simply by adding the link given below in the <head></head> section.

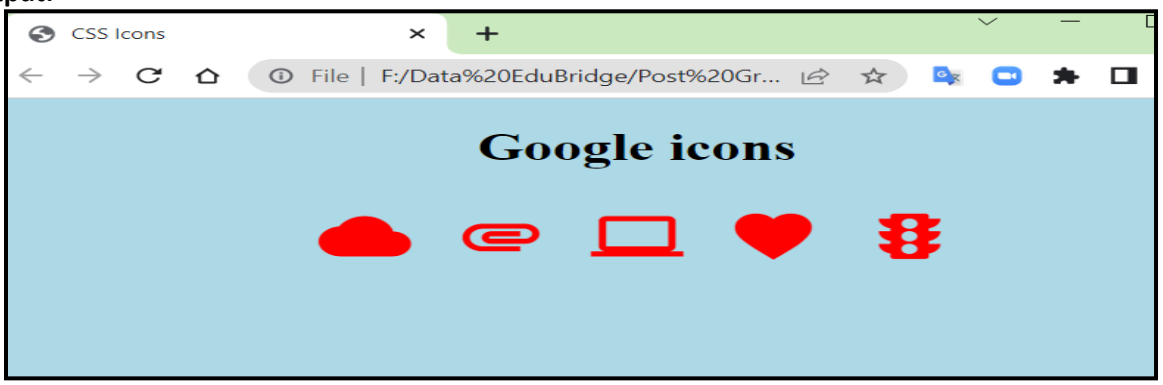
<link** rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons"**>

Example:

```
<!DOCTYPE html>
<html>
<head>
<title>CSS Icons</title>
<link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
<style>
body{
text-align:center;
background-color:lightblue;
}
.material-icons{
color:red;
font-size:50px;
margin:10px;
}
</style>
```

```
</head>
<body style="text-align:center">
<h1>Google icons</h1>
<i class="material-icons">cloud</i>
<i class="material-icons">attachment</i>
<i class="material-icons">computer</i>
<i class="material-icons">favorite</i>
<i class="material-icons">traffic</i>
</body>
</html>
```

Output:



Exercise

Trainer will initiate a discussion of common questions on CSS Selector as given below:

1. **Create a web page heading as Resume, Every sub heading should be Bold, Font size 14 from Font Family, all sub heading will contain bootstrap icons, all information under sub heading will have font size 12 from generic font family.**

Text Color

Text color

The color property is used to set the color of the text. The color is specified by:

- a color name - like "red"
- a HEX value - like "#ff0000"
- an RGB value - like "rgb(255,0,0)"

Look at CSS Color Values for a complete list of possible color values.

The default text color for a page is defined in the body selector.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
color: blue;
}
h1 {
  color: green;
}
</style>
</head>
<body>
<h1>This is heading 1</h1>
<p>This is an ordinary paragraph. Notice that this text is blue. The default text color for a page is defined in the body selector.</p>
<p>Another paragraph.</p>
</body>
</html>
```

Output:

Text Background Color

The background-color property sets the background color of an element.The background of an element is the total size of the element, including padding and border (but not the margin).Use a background color and a text color that makes the text easy to read.

CSS Syntax

background-color: color|transparent|initial|inherit;

Property Values

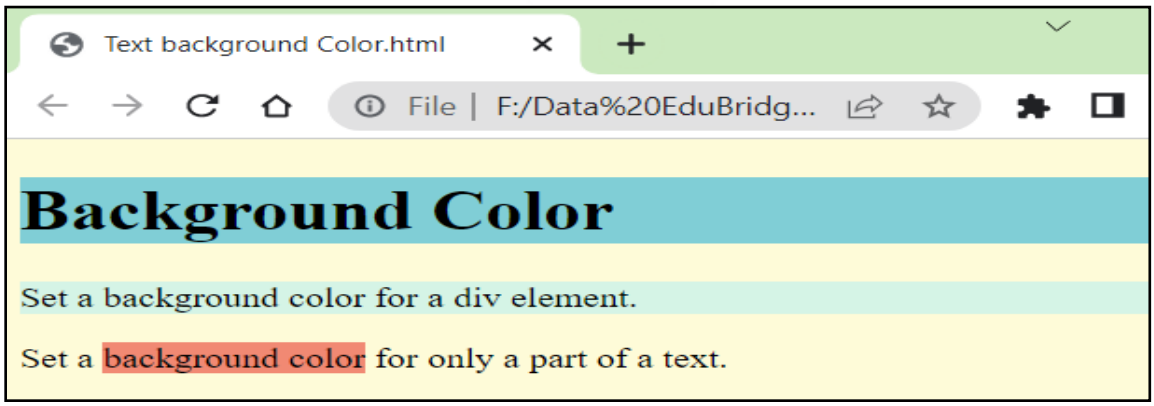
- **Color**Specifies the background color. Look at CSS Color Values for a complete list of possible color values.
- **Transparent** Specifies that the background color should be transparent. This is default
- **Initial Sets** this property to its default value. Read about initial
- **Inherit** Inherits this property from its parent element. Read about inherit

High contrast is very important for people with vision problems. So, always ensure that the contrast between the text color and the background color (or background image) is good!

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: #fefbd8;
}
h1 {
  background-color: #80ced6;
}
div {
  background-color: #d5f4e6;
}
span {
  background-color: #f18973;
}
</style>
</head>
<body>
<h1>Background Color</h1>
<div>Set a background color for a div element.</div>
<p>Set a <span>background color</span> for only a part of a text.</p>
</body>
</html>
```

Output:



CSS Border

The *border* properties is use to specify how the border of the box representing an element should look. There are three properties of a border you can change −

- The **border-color** specifies the color of a border.
- The **border-style** specifies whether a border should be solid, dashed line, double line, or one of the other possible values.
- The **border-width** specifies the width of a border.

Now, we will see how to use these properties with examples.

The border-color Property

CSS Borders

- **border-bottom-color** changes the color of bottom border.
- **border-top-color** changes the color of top border.
- **border-left-color** changes the color of left border.
- **border-right-color** changes the color of right border.

The following example shows the effect of all these properties −

```
<html>
<head>
<styletype="text/css">
      p.example1 {
        border:1px solid;
        border-bottom-color:#009900;/* Green */
        border-top-color:#FF0000;/* Red */
        border-left-color:#330000;/* Black */
        border-right-color:#0000CC;/* Blue */
}
      p.example2 {
        border:1px solid;
        border-color:#009900;/* Green */
```

```
}
</style>
</head>

<body>
<pclass="example1">
    This example is showing all borders in different colors.
</p>

<pclass="example2">
    This example is showing all borders in green color only.
</p>
</body>
</html>
```

It will produce the following result −

The border-style Property

The border-style property allows you to select one of the following styles of border −

- **none** − No border. (Equivalent of border-width:0;)
- **solid** − Border is a single solid line.
- **dotted** − Border is a series of dots.
- **dashed** − Border is a series of short lines.
- **double** − Border is two solid lines.
- **groove** − Border looks as though it is carved into the page.
- **ridge** − Border looks the opposite of groove.
- **inset** − Border makes the box look like it is embedded in the page.
- **outset** − Border makes the box look like it is coming out of the canvas.
- **hidden** − Same as none, except in terms of border-conflict resolution for table elements.

You can individually change the style of the bottom, left, top, and right borders of an element using the following properties −

- **border-bottom-style** changes the style of bottom border.
- **border-top-style** changes the style of top border.
- **border-left-style** changes the style of left border.
- **border-right-style** changes the style of right border.

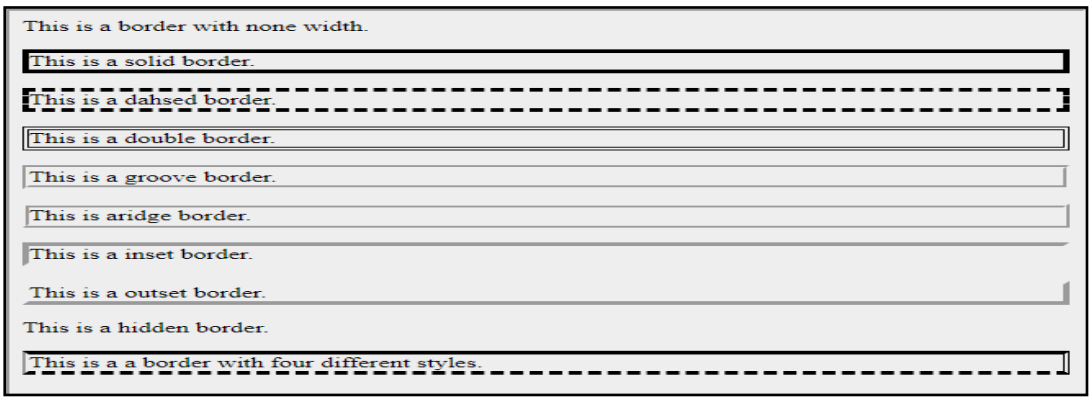
Example:

```
<html>
<head>
</head>
<body>
<pstyle="border-width:4px;border-style:none;">
    This is a border with none width.
</p>
<pstyle="border-width:4px;border-style:solid;">
```

```
        This is a solid border.
    </p>
    <pstyle="border-width:4px;border-style:dashed;">
        This is a dashed border.
    </p>
    <pstyle="border-width:4px;border-style:double;">
        This is a double border.
    </p>
    <pstyle="border-width:4px;border-style:groove;">
        This is a groove border.
    </p>
    <pstyle="border-width:4px;border-style:ridge">
        This is a ridge  border.
    </p>
    <pstyle="border-width:4px;border-style:inset;">
        This is a inset border.
    </p>
    <pstyle="border-width:4px;border-style:outset;">
        This is a outset border.
    </p>
    <pstyle="border-width:4px;border-style:hidden;">
        This is a hidden border.
    </p>

    <pstyle="border-width:4px;
    border-top-style:solid;
    border-bottom-style:dashed;
    border-left-style:groove;
    border-right-style:double;">
        This is a a border with four different styles.
    </p>
    </body>
    </html>
```

Output:



The border-width Property

The border-width property allows you to set the width of an element borders. The value of this property could be either a length in px, pt or cm or it should be set to *thin, medium or thick.*

You can individually change the width of the bottom, top, left, and right borders of an element using the following properties −

- **border-bottom-width** changes the width of bottom border.
- **border-top-width** changes the width of top border.
- **border-left-width** changes the width of left border.
- **border-right-width** changes the width of right border.

The following example shows all these border width −

```
<html>
<head>
</head>
<body>
<pstyle="border-width:4px;border-style:solid;">
    This is a solid border whose width is 4px.
</p>
<pstyle="border-width:4pt;border-style:solid;">
    This is a solid border whose width is 4pt.
</p>
<pstyle="border-width:thin;border-style:solid;">
    This is a solid border whose width is thin.
</p>
<pstyle="border-width:medium;border-style:solid;">
    This is a solid border whose width is medium;
</p>
<pstyle="border-width:thick;border-style:solid;">
    This is a solid border whose width is thick.
</p>
<pstyle="border-bottom-width:4px;border-top-width:10px;
    border-left-width:2px;border-right-width:15px;border-style:solid;">
    This is a a border with four different width.
</p>
</body>
</html>
```

It will produce the following result –

This is a solid border whose width is 4px.

This is a solid border whose width is 4pt.

This is a solid border whose width is thin.

This is a solid border whose width is medium;

This is a solid border whose width is thick.

This is a a border with four different width.

Border Properties Using Shorthand

The border property allows you to specify color, style, and width of lines in one property –

The following example shows how to use all the three properties into a single property. This is the most frequently used property to set border around any element.

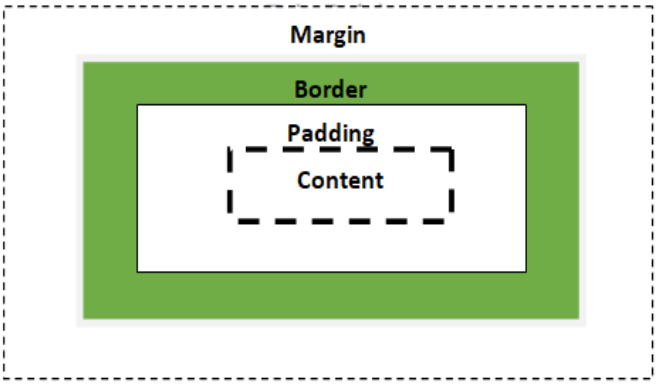
```
<html>
<head>
</head>
<body>
<pstyle="border:4px solid red;">
      This example is showing shorthand property for border.
</p>
</body>
</html>
```

Learner will create above code and analyze the output

CSS Box Model

CSS Box Model

All HTML elements can be considered as boxes. The term "box model" is used when talking about design and layout. The box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content. The box model allows us to place a border around elements and space elements in relation to other elements. The image below illustrates the box model:



- **Margin -** Clears an area around the border. The margin does not have a background color, it is completely transparent
- **Border -** A border that goes around the padding and content. The border is affected by the background color of the box
- **Padding -** Clears an area around the content. The padding is affected by the background color of the box
- **Content -** The content of the box, where text and images appear.

Elements of the width and height

Typically, when you assign the width and height of an attribute using the CSS width and height assets, it means you just positioned the height and width of the subject areas of that component. The additional height and width of the unit box is based on a range of influences.The specific area that an element box may occupy on a web page is measured asfollows-

Size of the box	Properties of CSS
Height	height + padding-top + padding-bottom + border-top + border-bottom + margin-top + margin-bottom
Width	width + padding-left + padding-right + border-left + border-right + margin-left + margin-right

CSS Box Example

```
<!DOCTYPE html>
<head>
<style>
      .main
    {
        font-size:30px;
font-weight:bold;
text-align:left;
      }
#box
   {
        padding-top:30px;
        width: 300px;
        height: 100px;
        border: 40px solid red;
        margin: 30px;
text-align:center;
```

```
        font-size:32px;
font-weight:bold;
      }
</style>
</head>
<body>
<div class="main">CSS Box-Model Property</div>
<div id="box">JavaTpoint</div>
</body>
</html>
```

Output:

CSS Box Dimension

Box Dimension/ Box-Sizing

The CSS box-sizing or dimension property is used to adjust or control the size of any element that accepts a **width** or **height**. It specifies how to calculate the total **width** and **height** of that element.

Syntax:

box-sizing: content-box|border-box;

content-box: This is the default value of the box-sizing property. In this mode, the width and height properties include only the content. Border and padding are not included in it i.e if we set an element's width to 200 pixels, then the element's content box will be 200 pixels wide, and the width of any border or padding will be added to the final rendered width.

Syntax:

box-sizing: content-box;

border-box: In this mode, the width and height properties include content, padding, and borders i.e if we set an element's width to 200 pixels, that 200 pixels will include any border or padding we added, and the content box will shrink to absorb that extra width. This typically makes it much easier to size elements.

Syntax:

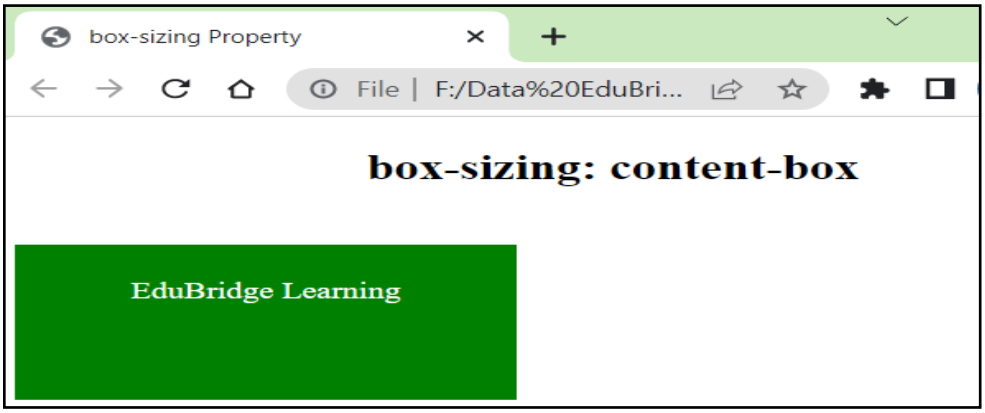
box-sizing: border-box;

Example:

```
<!DOCTYPE html>
<html>
```

```
<head>
    <title>box-sizing Property</title>
    <style>
    div {
            width: 200px;
            height: 60px;
            padding: 20px;
            border: 2px solid green;
            background: green;
            color: white;
    }
            .content-box {
            box-sizing: content-box;
    }
    </style>
</head>
<body style="text-align: center;">
    <h2>box-sizing: content-box</h2>
    <br>
    <div class="content-box">EduBridge Learning </div>
</body>
</html>
```

Output:



Activity

Q1. Put the appropriate syntax values in the blank fields

<style>

_____ {

_____:____;

}

</style>

<body>
*** ***
This is a test and submits your answer into blank space
*** ***
</body>

Q2. Using CSS Box Model draw the following with font size 16, font color Blue, box border dashed

Q3. Using external style sheet create a web page with font size 14, font generic family, and background colour blue and font colour Black