Unit 3

(sy kk bca)

Cascading style sheets

Introduction to CSS, Creating style sheets, common tasks with CSS, Colors, the font family, font metrics, length units, absolute units, relative units, the pixel unit, percentages as values, keywords as values, various properties such as the font -size property, font size property etc., Assigning classes, tags and attributes for applying classes, applying classes to an HTML tag, applying classes to other document parts, the layer tag, CSS Tags.

Introduction to CSS

- CSS stands for Cascading Style Sheet.
- CSS is used to design HTML tags.
- CSS is a widely used language on the web.
- HTML, CSS and JavaScript are used for web designing. It helps the web designers to apply style on HTML tags.
- **1994-** HÃ¥kon Wium Lie proposed the idea of CSS.
- **1996-** The first version of CSS was invented.
- **1998-** CSS 2 was released and work on CSS 3 began.
- **2011-** CSS 2.1 was released, which fixed the errors found in CSS 2

What is CSS

- CSS stands for Cascading Style Sheets. It is a style sheet language which is used to describe the look and formatting of a document written in markup language. It provides an additional feature to HTML. It is generally used with HTML to change the style of web pages and user interfaces. It can also be used with any kind of XML documents including plain XML, SVG and XUL.
- CSS is used along with HTML and JavaScript in most websites to create user interfaces for web applications and user interfaces for many mobile applications.

What does CSS do

- You can add new looks to your old HTML documents.
- You can completely change the look of your website with only a few changes in CSS code.

Why use CSS

These are the three major benefits of CSS:

1) Solves a big problem

Before CSS, tags like font, color, background style, element alignments, border and size had to be repeated on every web page. This was a very long process. For example: If you are developing a large website where fonts and color information are added on every single page, it will be become a long and expensive process. CSS was created to solve this problem.

2) Saves a lot of time

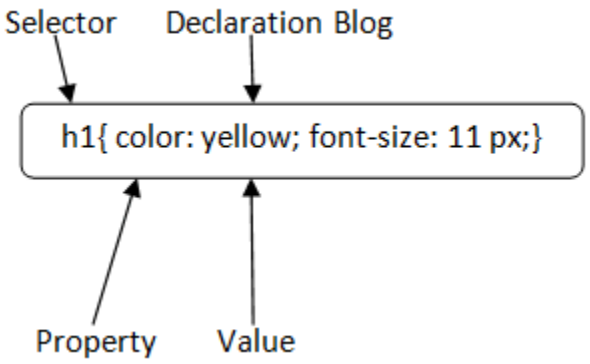
CSS style definitions are saved in external CSS files so it is possible to change the entire website by changing just one file.

3) Provide more attributes

CSS provides more detailed attributes than plain HTML to define the look and feel of the website.

CSS Syntax

A CSS rule set contains a selector and a declaration block.



Selector: Selector indicates the HTML element you want to style. It could be any tag like <h1>, <title> etc.

Declaration Block: The declaration block can contain one or more declarations separated by a semicolon. For the above example, there are two declarations:

1. color: yellow;
2. font-size: 11 px;

Each declaration contains a property name and value, separated by a colon.

Property: A Property is a type of attribute of HTML element. It could be color, border etc.

Value: Values are assigned to CSS properties. In the above example, value "yellow" is assigned to color property.

Creating Style Sheets.

CSS is added to HTML pages to format the document according to information in the style sheet. There are three ways to insert CSS in HTML documents.

1. Inline CSS
2. Internal CSS
3. External CSS

1) Inline CSS

Inline CSS is used to apply CSS on a single line or element.

For example:

```
<p style="color:blue">Hello CSS</p>
```

For more visit here: Inline CSS

2) Internal CSS

Internal CSS is used to apply CSS on a single document or page. It can affect all the elements of the page. It is written inside the style tag within head section of html.

For example:

```
<style>
p{color:blue}
</style>
```

3) External CSS

External CSS is used to apply CSS on multiple pages or all pages. Here, we write all the CSS code in a css file. Its extension must be .css for example style.css.

For example:

```
p{color:blue}
```

You need to link this style.css file to your html pages like this:

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

The link tag must be used inside head section of html.

Inline CSS

We can apply CSS in a single element by inline CSS technique.

The inline CSS is also a method to insert style sheets in HTML document. This method mitigates some advantages of style sheets so it is advised to use this method sparingly.

If you want to use inline CSS, you should use the style attribute to the relevant tag.

Syntax:

```
<htmltag style="cssproperty1:value; cssproperty2:value;"> </htmltag>
```

Example:

```
<h2 style="color:red;marginleft:40px;">Inline CSS is applied on this heading.</h2>
<p>This paragraph is not affected.</p>
```

Output:

Inline CSS is applied on this heading.

This paragraph is not affected.

Disadvantages of Inline CSS

- You cannot use quotations within inline CSS. If you use quotations the browser will interpret this as an end of your style value.
- These styles cannot be reused anywhere else.
- These styles are tough to be edited because they are not stored at a single place.
- It is not possible to style pseudo-codes and pseudo-classes with inline CSS.
- Inline CSS does not provide browser cache advantages.

Internal CSS

The internal style sheet is used to add a unique style for a single document. It is defined in <head> section of the HTML page inside the <style> tag.

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: linen;
}
h1 {
    color: red;
    margin-left: 80px;
}
</style>
</head>
<body>
<h1>The internal style sheet is applied on this heading.</h1>
<p>This paragraph will not be affected.</p>
</body>
</html>
```

External CSS

The external style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.

It uses the <link> tag on every page and the <link> tag should be put inside the head section.

Example:

```
1. <head>
2. <link rel="stylesheet" type="text/CSS" href="style.css">
3. </head>
```

The external style sheet may be written in any text editor but must be saved with a.CSS extension. This file should not contain HTML elements.

Let's take an example of a style sheet file named "style.css".

```
body {
    background-color: lightblue;
}
h1 {
    color: navy;
    margin-left: 20px;
}
```

Note: You should not use a space between the property value and the unit. For example: It should be margin-left:20px not margin-left:20 px.

Common tasks with CSS, Colors, the font family, font metrics, length units, absolute units, relative units, the pixel unit, percentages as values, keywords as values, various properties such as the font -size property, font size property etc.,

CSS Colors

The color property in CSS is used to set the color of HTML elements. Typically, this property is used to set the background color or the font color of an element.

In CSS, we use color values for specifying the color. We can also use this property for the border-color and other decorative effects.

We can define the color of an element by using the following ways:

- RGB format.
- RGBA format.
- Hexadecimal notation.
- HSL.
- HSLA.
- Built-in color.

Let's understand the syntax and description of the above ways in detail.

RGB Format

RGB format is the short form of '**RED GREEN** and **BLUE**' that is used for defining the color of an HTML element simply by specifying the values of R, G, B that are in the range of 0 to 255.

The color values in this format are specified by using the **rgb()** property. This property allows three values that can either be in percentage or integer (range from 0 to 255).

This property is not supported in all browsers; that's why it is not recommended to use it.

Syntax

1. color: rgb(R, G, B);

RGBA Format

It is almost similar to RGB format except that **RGBA** contains **A (Alpha)** that specifies the element's transparency. The value of alpha is in the range **0.0 to 1.0**, in which **0.0** is for fully transparent, and **1.0** is for not transparent.

Syntax

1. color:rgba(R, G, B, A);

Hexadecimal notation

Hexadecimal can be defined as a six-digit color representation. This notation starts with the **# symbol** followed by six characters ranges from **0 to F**. In hexadecimal notation, the first two digits represent the **red (RR)** color value, the next two digits represent the **green (GG)** color value, and the last two digits represent the **blue (BB)** color value.

The black color notation in hexadecimal is #000000, and the white color notation in hexadecimal is #FFFFFF. Some of the codes in hexadecimal notation are #FF0000, #00FF00, #0000FF, #FFFF00, and many more.

Syntax

1. color:#(0-F)(0-F)(0-F)(0-F)(0-F)(0-F);

Built-in Color

As its name implies, built-in color means the collection of previously defined colors that are used by using a name such as red, blue, green, etc.

Syntax

1. color: color-name;

Let's see the list of built-in colors along with their decimal and hexadecimal values.

S.no.	Color name	Hexadecimal Value	Decimal Value or rgb() value
1.	Red	#FF0000	rgb(255,0,0)
2.	Orange	#FFA500	rgb(255,165,0)
3.	Yellow	#FFFF00	rgb(255,255,0)
4.	Pink	#FFC0CB	rgb(255,192,203)
5.	Green	#008000	rgb(0,128,0)
6.	Violet	#EE82EE	rgb(238,130,238)
7.	Blue	#0000FF	rgb(0,0,255)
8.	Aqua	#00FFFF	rgb(0,255,255)
9.	Brown	#A52A2A	rgb(165,42,42)
10.	White	#FFFFFF	rgb(255,255,255)
11.	Gray	#808080	rgb(128,128,128)
12.	Black	#000000	rgb(0,0,0)

The illustration of CSS colors, which includes the above properties, is given below.

Example

```
<html>
   <head>
      <title>CSS hsl color property</title>
      <style>
         h1{
            text-align:center;
         }
         #rgb{
           color:rgb(255,0,0);
         }
         #rgba{
           color:rgba(255,0,0,0.5);
         }
         #hex{
           color:#EE82EE;
         }
         #short{
           color: #E8E;
         }
         #hsl{
           color:hsl(0,50%,50%);
         }
         #hsla{
           color:hsla(0,50%,50%,0.5);
         }
         #built{
           color:green;
         }
      </style>
```

```
    </head>
    <body>
        <h1 id="rgb">
            Hello World. This is RGB format.
        </h1>
        <h1 id="rgba">
          Hello World. This is RGBA format.
      </h1>
      <h1 id="hex">
       Hello World. This is Hexadecimal format.
    </h1>
    <h1 id="short">
     Hello World. This is Short-hexadecimal format.
    </h1>
     <h1 id="hsl">
     Hello World. This is HSL format.
    </h1>
    <h1 id="hsla">
     Hello World. This is HSLA format.
    </h1>
    <h1 id="built">
     Hello World. This is Built-in color format.
    </h1>
        </body>
    </html>
```

CSS font-family

This CSS property is used to provide a comma-separated list of font families. It sets the font-face for the text content of an element. This property can hold multiple font names as a fallback system, i.e., if one font is unsupported in the browser, then others can be used. The different font-family is used for making attractive web pages.

There are two types of font-family names in CSS, which are defined below:

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

- **generic-family:** It is the name of the generic family that includes five categories, which are "serif", "sans-serif", "cursive", "fantasy", and "monospace". It should be placed at last in the list of the font family names.

Let's define the generic-family categories.

serif: It is mainly used when we are writing the text for printing, such as books, magazines, newspapers, etc. It includes the font-family such as Georgia, Garamond, Times New Roman, Minion, and many more

sans-serif: It is a modern, formal, and simple style. It is widely used but most often in the digital form of text. It includes the font-family that are Arial, Calibri, Verdana, Futura, Lato, and many more.

cursive: It is mainly used for writing the invitation letter, informal messages, etc. It is like a handwritten text which is written by a pen or a brush. The font-family that it includes is Insolente, Corsiva, Flanella, Belluccia, Zapfino, and many more.

monospace: It is for instructions, mailing address, typewritten text, etc. It includes the font-family that is Monaco, SimSun, Courier, Consolas, Inconsolata, and many more.

fantasy: It makes the text expressive, decorative, and impactful. It includes the font-family that is Impact, Copperplate, Cracked, Critter, and many more.

Syntax

1. font-family: family-name|generic-family|initial|inherit;

Values

Let's see the values of the font-family property.

family-name/generic-family: It is the list of font-family names and the generic family names.

initial: It is used to set the property to its default value.

inherit: It is used to inherit the property from its parent element.

Let's understand it by using an illustration.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
body{
text-align:center;
}
h1.a {
  font-family: "Times New Roman", Times, serif;
  color:Red;
}

h2.b {
  font-family: Arial, Helvetica, sans-serif;
  color:blue;
}
</style>
</head>
<body>
<h1>The font-family Property</h1>

<h1 class="a">Hello World :) :)</h1>

<h2 class="b">Welcome to the college</h2>

</body>
</html>
```

CSS Background

CSS background property is used to define the background effects on element. There are 5 CSS background properties that affects the HTML elements:

1. background-color
2. background-image
3. background-repeat
4. background-attachment
5. background-position

1) CSS background-color

The background-color property is used to specify the background color of the element.

You can set the background color like this:

```
<!DOCTYPE html>
<html>
<head>
<style>
h2,p{
    background-color: #b0d4de;
}
</style>
</head>
<body>
<h2>My first CSS page.</h2>
<p>Hello . This is an example of CSS background-color.</p>
</body>
</html>
```

Output:

My first CSS page.

Hello Javatpoint. This is an example of CSS background-color.

---

2) CSS background-image

The background-image property is used to set an image as a background of an element. By default the image covers the entire element. You can set the background image for a page like this.

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
body {
background-image: url("paper1.gif");
margin-left:100px;
}
</style>
</head>
<body>
<h1>Hello  </h1>
</body>
</html>
```

CSS units

There are various units in CSS to express the measurement and length. A CSS unit is used to determine the property size, which we set for an element or its content. The units in CSS are required to define the measurement such as *margin: 20px;* in which the **px** (or pixel) is the CSS unit. They are used to set margin, padding, lengths, and so on.

We cannot apply the whitespace between the number and the unit. The unit can be omitted for the value 0. Some properties of CSS allow the negative values of length.

The length unit in CSS is of two types:

- Absolute length.
- Relative length.

Absolute lengths

These are the fixed-length units, and the length expressed using the absolute units will appear as exactly that size. It is not recommended to use on-screen, because the size of the screen varies too much. So, the absolute units should be used when the medium of output is known, such as the print layout

Absolute units are useful when the responsiveness is not considered in a project. They are less favorable for the responsive sites because they do not scale when the screen changes.

Generally, absolute lengths are considered to be the same size always. The absolute length units are tabulated as follows:

Unit	Name	Explanation

cm	Centimeters	It is used to define the measurement in centimeters.
mm	Millimeters	It is used to define the measurement in millimeters.
in	Inches	It is used to define the measurement in inches. 1in = 96px = 2.54cm
pt	Points	It is used to define the measurement in points. 1pt = 1/72 of 1 inch.
pc	Picas	It is used to define the measurement in picas. 1pc = 12pt so, there 6 picas is equivalent to 1 inch.
px	Pixels	It is used to define the measurement in pixels. 1px = 1/96th of inch

Example

In this example, we are using the **font-size** property for the paragraphs for defining the values using the above absolute length units.

```
<!DOCTYPE html>
<html>
<head>
<style>
body{
text-align: center;
}
</style>
</head>
<body>
<h1> Absolute units </h1>
<p style = "font-size: 20px;" > It has a font-size: 20px; </p>
<p style = "font-size: 1.2cm;" > It has a font-size: 1.2cm; </p>
<p style = "font-size: .7in;" > It has a font-size: .7in; </p>
```

Unit	Name
em	It is relative to the font-size of the element.
ex	It is relative to the x-height of the font of the element. It is rarely used. The x-height is determined by the height of the lowercase letter 'x'.
ch	It is similar to the unit ex, but instead of using the height of the letter x, it measures the width of the integer "0" (zero).
rem	It is the font-size of the root element
vh	It is relative to the height of the viewport. 1vh = 1% or 1/100 of the height of the viewport.
vw	It is relative to the width of the viewport. 1vw = 1% or 1/100 of the width of viewport
vmin	It is relative to the smaller dimension of the viewport. 1vmin = 1% or 1/100 of the viewport's smaller dimension.
vmax	It is relative to the larger dimension of the viewport. 1vmax = 1% or 1/100 of the viewport's larger dimension.
%	It is used to define the measurement as a percentage that is relative to another value.

```
<p style = "font-size: 18pt;" > It has a font-size: 18pt; </p>
<p style = "font-size: 2pc;" > It has a font-size: 2pc; </p>
<p style = "font-size: 10mm;" > It has a font-size: 10mm; </p>
</body>
</html>
```

Relative lengths

Relative units are good to style the responsive site because they scale relative to the window size or the parent. They specify the length, which is relative to another length property.

Depending on the device, if the size of the screen varies too much, then the relative length units are the best because they scale better between the different rendering mediums. We can use the relative units as the default for the responsive units. It helps us to avoid update styles for different screen sizes.

The relative length units are tabulated as follows:

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
body{
text-align: center;
}
p{
line-height: 0.1cm;
color: blue;
}
</style>
</head>
<body>
<h1> Relative units </h1>
<p style = "font-size: 2em;" > It has a font-size: 2em; </p>
<p style = "font-size: 8ex;" > It has a font-size: 8ex; </p>
<p style = "font-size: 6ch;" > It has a font-size: 6ch; </p>
<p style = "font-size: 4rem;" > It has a font-size: 4rem; </p>
<p style = "font-size: 4vw;" > It has a font-size: 4vw; </p>
<p style = "font-size: 10vh;" > It has a font-size: 10vh; </p>
<p style = "font-size: 10vmin;" > It has a font-size: 10vmin; </p>
<p style = "font-size: 8vmax;" > It has a font-size: 8vmax; </p>
<p style = "font-size: 400%;" > It has a font-size: 400%; </p>
</body>
```

```
</html>
```

CSS units: Time

Some animation properties require values to express in time.

Unit	Explanation
s	It is the duration of time in seconds.
ms	It is the duration of time in milliseconds. 1ms = 1/100 of a second

Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
width: 200px;
height: 200px;
background: red;
border-radius: 50px;
transition-property: background, width, height;
transition-duration: 1s, 2s, 3s;
}

div:hover
{
width:300px;
background: blue;
height:300px;
border-radius: 80px;
}
</style>
```

```
</head>
<body>
<center>
<h2>Hover to see effects.</h2>
<div></div>
</center>
</body>
</html>
```

Output

CSS units: Angles

The transform properties in CSS require values to express in angles

Unit	Explanation
deg	It expresses the angles in degrees.
grad	It expresses the angles in gradians, i.e., 1/400 of a turn.
turn	It expresses the angles in turns, i.e., 360 degrees.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>

img
 {
 border:9px ridge gray;
 border-radius:30px;
 margin:10px;
 transition-duration:2s;
  }
```

```
#img1:hover{
transform: rotate(30deg);
transform-origin: bottom left 50px;
}

</style>
</head>

<body>
<center>
  <img src="img.jpg" id="img1"/>

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

CSS Font-size

The font-size property in CSS is used to specify the height and size of the font. It affects the size of the text of an element. Its default value is medium and can be applied to every element. The values of this property include **xx-small**, **small**, **x-small**, etc.

Syntax

1. font-size: medium|large|x-large|xx-large|xx-small|x-small|small|;

The font-size can be relative or absolute.

Absolute-size

It is used to set the text to a definite size. Using absolute-size, it is not possible to change the size of the text in all browsers. It is advantageous when we know the physical size of the output.

Relative-size

It is used to set the size of the text relative to its neighboring elements. With relative-size, it is possible to change the size of the text in browsers.

NOTE: If we do not define a font-size, then for the normal text such as paragraphs, the default size is 16px, which is equal to 1em.

Font-size with pixels

When we set the size of text with pixels, then it provides us the full control over the size of the text.

Example

```
<!DOCTYPE html>
<html>
<head>
<style>

#first {
  font-size: 40px;
}
#second {
  font-size: 20px;
}
</style>
</head>
<body>

<p id="first">This is a paragraph having size 40px.</p>
<p id="second">This is another paragraph having size 20px.</p>

</body>
</html>
```

Font-size with em

It is used to resize the text. Most of the developers prefer **em** instead of **pixels**. It is recommended by the world wide web consortium (W3C). As stated above, the default text size in browsers is 16px. So, we can say that the default size of **1em** is **16px**.

The formula for calculating the size from **pixels** to **em** is **em = pixels/16**.

Example

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

```
<head>
<style>
#first {
  font-size: 2.5em; /* 40px/16=2.5em */
}

#second {
  font-size: 1.875em; /* 30px/116=1.875em */
 }

#third {
  font-size: 0.875em; /* 14px/16=0.875em */
}
</style>
</head>
<body>

<p id='first'>First paragraph.</p>
<p id='second'>Second paragraph</p>
<p id='third'>Third Paragraph.</p>
</body>
</html>
```

Responsive font size

We can set the size of the text by using a **vw unit**, which stands for the '**viewport width**'. The viewport is the size of the browser window.

1vw = 1% of viewport width.

If the width of the viewport is 50cm, then the 1vw is equal to 0.5 cm.

Example

```
<!DOCTYPE html>
<html>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<body>
```

```
<p style="font-size:5vw;">First paragraph having the width of 5vw.</p>
<p style="font-size:10vw;">Second paragraph having the width of 10vw.</p>

</body>
</html>
```

Font-size with the length property

It is used to set the size of the font in length. The length can be in cm, px, pt, etc. Negative values of **length** property are not allowed in font-size.

Syntax

1. font-size: length;

Example

```
<!DOCTYPE html>
<html>
  <head>
    <style>
      .length {
        color:red;
        font-size: 5cm;
      }
    </style>
  </head>

  <body>
    <h1>font-size property</h1>

    <p class = "length">A paragraph having length 5cm.</p>
  </body>
</html>
```

CSS Border

The CSS border is a shorthand property used to set the border on an element.

The CSS border properties are use to specify the style, color and size of the border of an element. The CSS border properties are given below

- border-style
- border-color
- border-width
- border-radius

1) CSS border-style

The Border style property is used to specify the border type which you want to display on the web page.

There are some border style values which are used with border-style property to define a border.

Value	Description
none	It doesn't define any border.
dotted	It is used to define a dotted border.
dashed	It is used to define a dashed border.
solid	It is used to define a solid border.
double	It defines two borders wIth the same border-width value.
groove	It defines a 3d grooved border. effect is generated according to border-color value.
ridge	It defines a 3d ridged border. effect is generated according to border-color value.
inset	It defines a 3d inset border. effect is generated according to border-color value.
outset	It defines a 3d outset border. effect is generated according to border-color value.

```
1. <!DOCTYPE html>
```

```
<html>
<head>
<style>
p.none {border-style: none;}
p.dotted {border-style: dotted;}
p.dashed {border-style: dashed;}
p.solid {border-style: solid;}
p.double {border-style: double;}
p.groove {border-style: groove;}
p.ridge {border-style: ridge;}
p.inset {border-style: inset;}
p.outset {border-style: outset;}
p.hidden {border-style: hidden;}
</style>
</head>
<body>
<p class="none">No border.</p>
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
<p class="outset">An outset border.</p>
<p class="hidden">A hidden border.</p>
</body>
</html>
```

Output:

No border.

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

A hidden border.

2) CSS border-width

The border-width property is used to set the border's width. It is set in pixels. You can also use the one of the three pre-defined values, thin, medium or thick to set the width of the border.

Note: The border-width property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
    border-style: solid;
    border-width: 5px;
}
p.two {
    border-style: solid;
    border-width: medium;
}
p.three {
    border-style: solid;
    border-width: 1px;
}
</style>
```

```
</head>
<body>
<p class="one">Write your text here.</p>
<p class="two">Write your text here.</p>
<p class="three">Write your text here.</p>
</body>
</html>
```

3) CSS border-color

There are three methods to set the color of the border.

- Name: It specifies the color name. For example: "red".
- RGB: It specifies the RGB value of the color. For example: "rgb(255,0,0)".
- Hex: It specifies the hex value of the color. For example: "#ff0000".

There is also a border color named "transparent". If the border color is not set it is inherited from the color property of the element.

Note: The border-color property is not used alone. It is always used with other border properties like "border-style" property to set the border first otherwise it will not work.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one {
    border-style: solid;
    border-color: red;
}
p.two {
    border-style: solid;
    border-color: #98bf21;
}
</style>
</head>
<body>
<p class="one">This is a solid red border</p>
```

```
<p class="two">This is a solid green border</p>
</body>
</html>
```

CSS Images

Images are an important part of any web application. Including a lot of images in a web application is generally not recommended, but it is important to use the images wherever they required. CSS helps us to control the display of images in web applications.

The styling of an image in CSS is similar to the styling of an element by using the borders and margins. There are multiple CSS properties such as **border** property, **height** property, **width** property, etc. that helps us to style an image.

Let's discuss the styling of images in CSS by using some illustrations.

Thumbnail Image

The border property is used to make a thumbnail image.

Example

```
<!DOCTYPE html>
<html>
<head>
  <style>
img{
border: 2px solid red;
border-radius:5px;
padding:10px;
}
h2{
color:red;
}
  </style>
</head>

<body>
```

```
 <h1>Thumbnail Image</h1>
<img src="jtp.png"></img>
<h2>Welcome too world</h2>
</body>
</html>
```

Transparent image

To make an image transparent, we have to use the **opacity** property. The value of this property lies between 0.0 to 1.0, respectively.

Example

```
 <!DOCTYPE html>
<html>
<head>
   <style>
img{
border: 2px solid red;
border-radius:5px;
padding:10px;
opacity:0.3;
}
h2{
color:red;
}
   </style>
</head>

<body>
 <h1>Transparent Image</h1>
<img src="jtp.png"></img>
<h2>Welcome to Rose vally</h2>
</body>
</html>
```

Rounded image

The **border-radius** property sets the radius of the bordered image. It is used to create the rounded images. The possible values for the rounded corners are given as follows:

- **border-radius:** It sets all of the four border-radius property.
- **border-top-right-radius:** It sets the border of the top-right corner.
- **border-top-left-radius:** It sets the border of the top-left corner.
- **border-bottom-right-radius:** It sets the border of the bottom-right corner.
- **border-bottom-left-radius:** It sets the border of the bottom-left corner.

Example

```
 <!DOCTYPE html>
<html>
<head>
   <style>
#img1{
border: 2px solid green;
border-radius:10px;
padding:5px;
}
#img2{
border: 2px solid green;
border-radius:50%;
padding:5px;
}

h2{
color:red;
}
   </style>
</head>

<body>
```

```
 <h1>Rounded Image</h1>
<img src="jtp.png" id="img1"></img>
<h2>Welcome to javaTpoint</h2>

 <h1>Circle Image</h1>
<img src="jtp.png" id="img2"></img>
<h2>Welcome to Thane</h2>
</body>
</html>
```

Responsive Image

It automatically adjusts to fit on the screen size. It is used to adjust the image to the specified box automatically.

Example

```
<!DOCTYPE html>
<html>
<head>
   <style>
#img1{
max-width:100%;
height:auto;
}
h2{
color:red;
}
   </style>
</head>

<body>
 <h1>Responsive image</h1>
 <h2>You can resize the browser to see the effect</h2>
<img src="jtp.png" id="img1" width="1000" height="300"></img>
<h2>Welcome to kalwa</h2>
</body>
```

```
</html>
```

Center an Image

We can center an image by using the **left-margin** and **right-margin** property. We have to set these properties to **auto** in order to make a block element.

Example

```
<!DOCTYPE html>
<html>
<head>
   <style>
#img1{
margin-left:auto;
margin-right:auto;
display:block;
}
h1,h2{
text-align:center;
}
   </style>
</head>

<body>
 <h1>Center image</h1>
<img src="jtp.png" id="img1"></img>
<h2>Welcome tohome </h2>
</body>
</html>
```

Assigning classes, tags and attributes for applying classes, applying classes to an HTML tag, applying classes to other document parts, the layer tag, CSS Tags.

The *.class* selector selects elements with a specific class attribute.

To select elements with a specific class, write a period (.) character, followed by the name of the class.

You can also specify that only specific HTML elements should be affected by a class. To do this, start with the element name, then write the period (.) character, followed by the name of the class.

Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
.class1.class2	.name1.name2	Selects all elements with both *name1* and *name2* set within its class attribute
.class1 .class2	.name1 .name2	Selects all elements with *name2* that is a descendant of an element with *name1*
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element.class	p.intro	Selects all <p> elements with class="intro"
element,element	div, p	Selects all elements and all <p> elements

element element	div p	Selects all <p> elements inside elements
element>element	div > p	Selects all <p> elements where the parent is a element
element+element	div + p	Selects the first <p> element that is placed immediately after elements
element1~element2	p ~ ul	Selects every element that is preceded by a <p> element
[*attribute*]	[target]	Selects all elements with a target attribute
[*attribute=value*]	[target=_blank]	Selects all elements with target="_blank"
[*attribute~=value*]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
[*attribute|=value*]	[lang|=en]	Selects all elements with a lang attribute value equal to "en" or starting with "en-"
[*attribute^=value*]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
[*attribute$=value*]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"

[*attribute*=value*]	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"
:active	a:active	Selects the active link
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
:checked	input:checked	Selects every checked <input> element
:default	input:default	Selects the default <input> element
:disabled	input:disabled	Selects every disabled <input> element
:empty	p:empty	Selects every <p> element that has no children (including text nodes)
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> element that is the first child of its parent

::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the input element which has focus
:fullscreen	:fullscreen	Selects the element that is in full-screen mode
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects input elements with a value within a specified range
:indeterminate	input:indeterminate	Selects input elements that are in an indeterminate state
:invalid	input:invalid	Selects all input elements with an invalid value
:lang(*language*)	p:lang(it)	Selects every <p> element with a lang attribute equal to "it" (Italian)

:last-child	p:last-child	Selects every <p> element that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:link	a:link	Selects all unvisited links
::marker	::marker	Selects the markers of list items
:not(*selector*)	:not(p)	Selects every element that is not a <p> element
:nth-child(*n*)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(*n*)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(*n*)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(*n*)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent

:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent
:optional	input:optional	Selects input elements with no "required" attribute
:out-of-range	input:out-of-range	Selects input elements with a value outside a specified range
::placeholder	input::placeholder	Selects input elements with the "placeholder" attribute specified
:read-only	input:read-only	Selects input elements with the "readonly" attribute specified
:read-write	input:read-write	Selects input elements with the "readonly" attribute NOT specified
:required	input:required	Selects input elements with the "required" attribute specified
:root	:root	Selects the document's root element

::selection	::selection	Selects the portion of an element that is selected by a user
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	input:valid	Selects all input elements with a valid value
:visited	a:visited	Selects all visited links

END