CSS

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

CSS is a language that describes the style of an HTML document.
CSS describes how HTML elements should be displayed.

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.

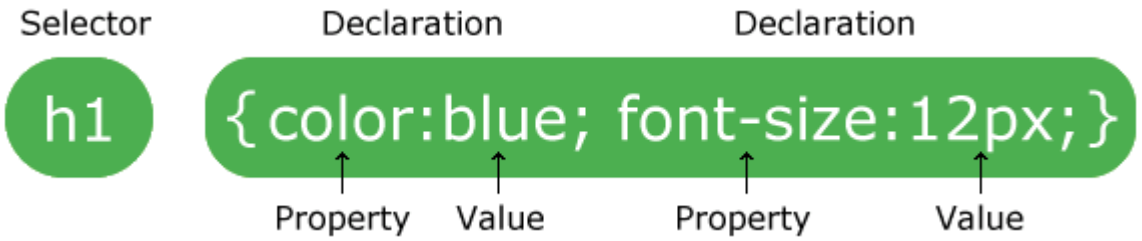
•**Superior styles to HTML** - CSS has a much wider array of attributes than HTML, so you can give a far better look to your HTML page in comparison to HTML attributes.

•**Multiple Device Compatibility** - Style sheets allow content to be optimized for more than one type of device.

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

CSS Syntax

A CSS rule-set consists of a selector and a declaration block:



The selector points to the HTML element you want to style.

The declaration block contains one or more declarations separated by semicolons.

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

Multiple CSS declarations are separated with semicolons, and declaration blocks are surrounded by curly braces.

CSS Selectors

CSS selectors are used to "find" (or select) the HTML elements you want to style.

The Class Selectors

You can define style rules based on the class attribute of the elements. All the elements having that class will be formatted according to the defined rule.

```
.black { color: #000000; }
```

The ID Selectors

You can define style rules based on the *id* attribute of the elements. All the elements having that *id* will be formatted according to the defined rule.

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

Three Ways to Insert CSS

There are three ways of inserting a style sheet:

- External CSS
- Internal CSS
- Inline CSS

External CSS

With an external style sheet, you can change the look of an entire website by changing just one file!
Each HTML page must include a reference to the external style sheet file inside the <link> element, inside the head section.

EX:

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

"mystyle.css“

```
body {
  background-color: lightblue;
}

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

Internal CSS

An internal style sheet may be used if one single HTML page has a unique style.

The internal style is defined inside the <style> element, inside the head section.

```
<head>
<style>
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>
```

Inline CSS

An inline style may be used to apply a unique style for a single element.
To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.

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

<h1 style="color:red;text-align:center;">This is a red heading</h1>
<p style="color:green;">This is a paragraph.</p>

</body>
</html>
```

CSS Comments

Comments are used to explain the code, and may help when you edit the source code later

Comments are ignored by browsers.

A CSS comment is placed inside the `<style>` element, and starts with `//` and ends with `//`:

```
// This is a single-line comment
p {
  color: red;
}
```

Comments can also span multiple lines:

```
/* This is
a multi-line
comment */

p {
  color: red;
}
```

CSS - Colors

Format	Syntax	Example
Hex Code	#RRGGBB	p{color:#FF0000;}
Short Hex Code	#RGB	p{color:#6A7;}
RGB %	rgb(rrr%,ggg%,bbb%)	p{color:rgb(50%,50%,50%);}
RGB Absolute	rgb(rrr,ggg,bbb)	p{color:rgb(0,0,255);}
keyword	aqua, black, etc.	p{color:teal;}

Color	Color HEX
	#000000
	#FF0000
	#00FF00
	#0000FF
	#FFFF00
	#00FFFF
	#FF00FF
	#C0C0C0
	#FFFFFF

CSS - Backgrounds

- background-color
- background-image
- background-size
- background-repeat
- background-attachment
- background-position

CSS background-color

The `background-color` property specifies the background color of an element.

Example

The background color of a page is set like this:

```
body {
  background-color: lightblue;
}
```

Set the Background Image

We can set the background image by calling local stored images as shown below −

```
<style>
body
 {
 background-image: url("/css/images/css.jpg");
 background-color: #cccccc;
}
 </style>
```

Repeat the Background Image

The following example demonstrates how to repeat the background image if an image is small.
You can use *no-repeat* value for *background-repeat* property if you don't want to repeat an image. default *background-repeat* property will have *repeat* value.

```
body
{ background-image: url("/css/images/css.jpg");
 background-repeat: no-repeat; }
```

Other possible values for are:

- repeat-x
- repeat-y

Set the Background Image Position

The following example demonstrates how to set the background image position center away from the left side.

```
body
{
background-image:
url("/css/images/css.jpg");
background-position:center;
 }
```

The following example demonstrates how to set the background image position 100 pixels away from the left side and 200 pixels down from the top.

```
body
{ background-image: url("/css/images/css.jpg");
 background-position:100px 200px; }
```

Font Properties

- 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.

Set the Font Family

```
<p style = "font-family:georgia,garamond,serif;">
```

Font Style

The `font-style` property is mostly used to specify italic text.
This property has three values:

- normal - The text is shown normally
- italic - The text is shown in italics
- oblique - The text is "leaning" (oblique is very similar to italic, but less supported)

Ex: `font-style: italic;`

Font Weight

The `font-weight` property specifies the weight of a font
Possible values could be *normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900*.

Ex:

```
<p style = "font-weight:bold;"> This font is bold. </p>
```

Set the Font Variant

The following example demonstrates how to set the font variant of an element.
Possible values are *normal and small-caps*.

```
<p style = "font-variant:small-caps;">
 This text will be rendered as small caps
</p>
```

TEXT FORMATTING

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

Ex

```
body {
  color: blue;
}
```

```
body {
  background-color: lightgrey;
  color: blue;
}

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

Text Alignment

The `text-align` property is used to set the horizontal alignment of a text.
A text can be left or right aligned, centered, or justified.

```
h1 {
  text-align: center;
}

h2 {
  text-align: left;
}

h3 {
  text-align: right;
}
P{text-align: justify;}
```

Text Decoration

The `text-decoration` property is used to set or remove decorations from text.

The value `text-decoration: none;` is often used to remove underlines from links:

```
h1 {
  text-decoration: overline;
}

h2 {
  text-decoration: line-through;
}

h3 {
  text-decoration: underline;
}
```

Text Transformation

The `text-transform` property is used to specify uppercase and lowercase letters in a text.
It can be used to turn everything into uppercase or lowercase letters, or capitalize the first letter of each word:

```
p.uppercase {
  text-transform: uppercase;
}

p.lowercase {
  text-transform: lowercase;
}

p.capitalize {
  text-transform: capitalize;
}
```

Set the Space between Characters

The following example demonstrates how to set the space between characters. Possible values are *normal or a number specifying space.*.

```
<p style = "letter-spacing:5px;">
This text is having space between letters. </p>
```

Set the Space between Words

The following example demonstrates how to set the space between words. Possible values are *normal or a number specifying space*.

```
<p style = "word-spacing:5px;">
This text is having space between words. </p>
```

Set the Text Indent

The following example demonstrates how to indent the first line of a paragraph. Possible values are *% or a number specifying indent space*.

```
<p style = "text-indent:1cm;">
```

Link- properties

The four links states are:

- `a:link` - a normal, unvisited link
- `a:visited` - a link the user has visited
- `a:hover` - a link when the user mouses over it
- `a:active` - a link the moment it is clicked

```
<html>
<head>
<style>
/* unvisited link */
a:link {
  color: red;
}

/* visited link */
a:visited {
  color: green;
}

/* mouse over link */
a:hover {
  color: hotpink;
}

/* selected link */
a:active {
  color: blue;
}
</style>
</head>p>
<body>

<p><b><a href="default.html" target=“new">This is a
link</a></b></p>
</body>
</html>
```

CSS - Using Images

- The **border** property is used to set the width of an image border.
- The **height** property is used to set the height of an image.
- The **width** property is used to set the width of an image.

```
<img style = "border:1px solid red;
height:100px;width:100px;" src = "1.jpg" />
<br />
<img style = "border:1px solid red; height:50%;" src = "1.jpg">
 </body>
```

CSS - Borders

- 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.

```
p.example1
{
border:1px solid ;
border-bottom-color:green;
border-top-color:red;
border-left-color:black
border-right-color:#0000CC; /* Blue */ }
```

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.

List- Properties

- The **list-style-type** allows you to control the shape or appearance of the marker.
- The **list-style-image** specifies an image for the marker rather than a bullet point or number.
- The **list-style** serves as shorthand for the preceding properties.

Sr. No.	Value & Description
1	**none** NA
2	**disc (default)** A filled-in circle
3	**circle** An empty circle
4	**square** A filled-in square

Value	Description	Example
decimal	Number	1,2,3,4,5
decimal-leading-zero	0 before the number	01, 02, 03, 04, 05
lower-alpha	Lowercase alphanumeric characters	a, b, c, d, e
upper-alpha	Uppercase alphanumeric characters	A, B, C, D, E
lower-roman	Lowercase Roman numerals	i, ii, iii, iv, v
upper-roman	Uppercase Roman numerals	I, II, III, IV, V
lower-greek	The marker is lower-greek	alpha, beta, gamma
lower-latin	The marker is lower-latin	a, b, c, d, e
upper-latin	The marker is upper-latin	A, B, C, D, E

```
ul.a {
  list-style-type: circle;
}

ul.b {
  list-style-type: square;
}

ol.c {
  list-style-type: upper-roman;
}

ol.d {
  list-style-type: lower-alpha;
}
```

An Image as The List Item Marker

The `list-style-image` property specifies an image as the list item marker:

```
ul {
  list-style-image: url(‘ff.jpg');
}
```

Table properties

Table – border

```
table{
  border: 1px solid black;
}
```

Table Width and Height

Width and height of a table are defined by the `width` and `height` properties.

```
table {
  width: 100%;
}

th {
  height: 50px;
}
```

Horizontal Alignment

The `text-align` property sets the horizontal alignment (like left, right, or center) of the content in <th> or <td>. By default, the content of <th> elements are center-aligned and the content of <td> elements are left-aligned.

```
th {
  text-align: left;
}
```

Vertical Alignment

The `vertical-align` property sets the vertical alignment (like top, bottom, or middle) of the content in <th> or <td>. By default, the vertical alignment of the content in a table is middle (for both <th> and <td> elements).

```
td {
  height: 50px;
  vertical-align: bottom;
}
```

Table Padding

To control the space between the border and the content in a table, use the `padding` property on <td> and <th> ele

```
th, td {
  padding: 15px;
  text-align: left;
}
```

Hoverable Table

Use the `:hover` selector on <tr> to highlight table rows on mouse over:

```
tr:hover {background-color: lightgrey;}
```

CSS Margins

The CSS `margin` properties are used to create space around elements, outside of any defined borders.

Margin - Individual Sides

CSS has properties for specifying the margin for each side of an element:

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

```css
p {
  margin-top: 100px;
  margin-bottom: 100px;
  margin-right: 150px;
  margin-left: 80px;
}
```

```css
p {
  margin: 25px 50px 75px 100px;
}
```

The auto Value

You can set the margin property to `auto` to horizontally center the element within its container.
The element will then take up the specified width, and the remaining space will be split equally between the left and right margins.

```
<html>
<head>
<style>
div {
  width:300px;
  margin: auto;
  border: 1px solid red;
}
</style>
</head>
<body>

<h2>Use of margin:auto</h2>
<p>This is a sample paragraph This is a sample paragraph This is a sample paragraph This is a sample paragraph This is a sample paragraph This is a sample paragraph This is a sample paragraph This is a sample paragraph
</p>
<div>
This div will be horizontally centered because it has margin: auto;
</div>
</body>
</html>
```

The position Property

The `position` property specifies the type of positioning method used for an element.
There are five different position values:

- `static`
- `relative`
- `fixed`
- `absolute`

position: static;

HTML elements are positioned static by default.
Static positioned elements are not affected by the top, bottom, left, and right properties.
An element with `position: static;` is not positioned in any special way; it is always positioned according to the normal flow of the page:

position: relative;

An element with `position: relative;` is positioned relative to its normal position.
Setting the top, right, bottom, and left properties of a relatively-positioned element will cause it to be adjusted away from its normal position. Other content will not be adjusted to fit into any gap left by the element.

position: fixed;

An element with `position: fixed;` is positioned relative to the viewport, which means it always stays in the same place even if the page is scrolled. The top, right, bottom, and left properties are used to position the element.
A fixed element does not leave a gap in the page where it would normally have been located.

position: absolute;

An element with `position: absolute;` is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed).
However; if an absolute positioned element has no positioned ancestors, it uses the document body, and moves along with page scrolling.
Note: A "positioned" element is one whose position is anything except `static`.

```css
div.static {
  position: static;
  border: 3px solid #73AD21;
}
```

```css
div.relative {
  position: relative;
  left: 30px;
  border: 3px solid #73AD21;
}
```

```css
div.fixed {
  position: fixed;
  bottom: 0;
  right: 0;
  width: 300px;
  border: 3px solid #73AD21;
}
```

```css
div.absolute {
  position: absolute;
  top: 80px;
  right: 0;
  width: 200px;
  height: 100px;
  border: 3px solid #73AD21;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
div.relative {
  position: relative;
  width: 400px;
  height: 200px;
  border: 3px solid #73AD21;
}

div.absolute {
  position: absolute;
  top: 80px;
  right: 0;
  width: 200px;
  height: 100px;
  border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>position: absolute;</h2>

<p>An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):</p>

<div class="relative">This div element has position: relative;
  <div class="absolute">This div element has position: absolute;</div>
</div>

</body>
</html>
```