

Dr. SNS RAJALAKSHMI COLLEGE OF ARTS & SCIENCE(AUTONOMOUS)

Accredited by NAAC (Cycle III) with 'A+' Grade

Affiliated to Bharathiar University

Coimbatore-641049



DEPARTMENT OF COMPUTER APPLICATIONS

III BCA

WEB DEDIGNING-16UCA504

PREPARED BY: Mrs.Lalitha

UNIT-III

What is CSS?

- CSS stands for Cascading Style Sheets
- CSS describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in CSS files

CSS Demo - One HTML Page - Multiple Styles!

Here we will show one HTML page displayed with four different stylesheets. Click on the "Stylesheet 1", "Stylesheet 2", "Stylesheet 3", "Stylesheet 4" links below to see the different styles:

Why Use CSS?

CSS is used to define styles for your web pages, including the design, layout and variations in display for different devices and screen sizes.

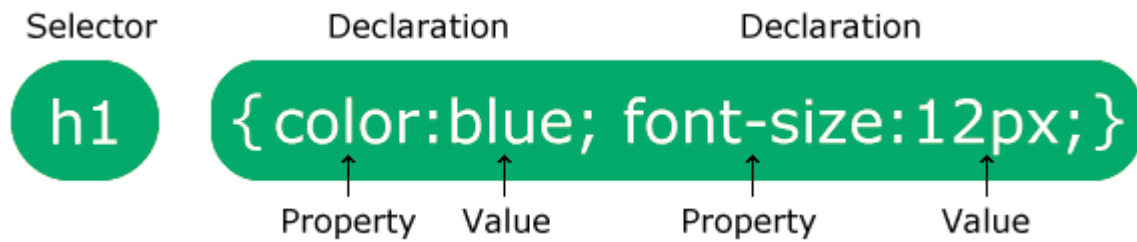
CSS Example

```
body {  
  background-color: lightblue;  
}  
  
h1 {  
  color: white;  
  text-align: center;  
}  
  
p {  
  font-family: verdana;  
  font-size: 20px;  
}
```

CSS Rules:

A CSS rule consists of a selector and a declaration block.

CSS Syntax



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.

Example

In this example all <p> elements will be center-aligned, with a red text color:

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

Example Explained

- **p** is a selector in CSS (it points to the HTML element you want to style: <p>).
- **color** is a property, and **red** is the property value
- **text-align** is a property, and **center** is the property value

CSS Properties

[accent-color](#)

Specifies an accent color for user-interface controls

[align-content](#)

Specifies the alignment between the lines inside a flexible container when

	use all available space
align-items	Specifies the alignment for items inside a flexible container
align-self	Specifies the alignment for selected items inside a flexible container
all	Resets all properties (except unicode-bidi and direction)
animation	A shorthand property for all the <i>animation-*</i> properties
animation-delay	Specifies a delay for the start of an animation
animation-direction	Specifies whether an animation should be played forwards, backwards or
animation-duration	Specifies how long an animation should take to complete one cycle
animation-fill-mode	Specifies a style for the element when the animation is not playing (before, after, both, or none)
animation-iteration-count	Specifies the number of times an animation should be played
animation-name	Specifies a name for the @keyframes animation

animation-play-state	Specifies whether the animation is running or paused
animation-timing-function	Specifies the speed curve of an animation
aspect-ratio	Specifies preferred aspect ratio of an element

backdrop-filter	Defines a graphical effect to the area behind an element
backface-visibility	Defines whether or not the back face of an element should be visible when facing the user
background	A shorthand property for all the <i>background-*</i> properties
background-attachment	Sets whether a background image scrolls with the rest of the page, or is fixed
background-blend-mode	Specifies the blending mode of each background layer (color/image)
background-clip	Defines how far the background (color or image) should extend within an element
background-color	Specifies the background color of an element

<u>background-image</u>	Specifies one or more background images for an element
<u>background-origin</u>	Specifies the origin position of a background image
<u>background-position</u>	Specifies the position of a background image
<u>background-position-x</u>	Specifies the position of a background image on x-axis
<u>background-position-y</u>	Specifies the position of a background image on y-axis
<u>background-repeat</u>	Sets if/how a background image will be repeated
<u>background-size</u>	Specifies the size of the background images
<u>block-size</u>	Specifies the size of an element in block direction
<u>border</u>	A shorthand property for <i>border-width</i> , <i>border-style</i> and <i>border-color</i>
<u>border-block</u>	A shorthand property for <i>border-block-width</i> , <i>border-block-style</i> and <i>border-block-color</i>
<u>border-block-color</u>	Sets the color of the borders at start and end in the block direction

border-block-end-color	Sets the color of the border at the end in the block direction
border-block-end-style	Sets the style of the border at the end in the block direction
border-block-end-width	Sets the width of the border at the end in the block direction
border-block-start-color	Sets the color of the border at the start in the block direction
border-block-start-style	Sets the style of the border at the start in the block direction
border-block-start-width	Sets the width of the border at the start in the block direction
border-block-style	Sets the style of the borders at start and end in the block direction
border-block-width	Sets the width of the borders at start and end in the block direction
border-bottom	A shorthand property for <i>border-bottom-width</i> , <i>border-bottom-style</i> and <i>border-bottom-color</i>
border-bottom-color	Sets the color of the bottom border
border-bottom-left-radius	Defines the radius of the border of the bottom-left corner

border-bottom-right-radius	Defines the radius of the border of the bottom-right corner
border-bottom-style	Sets the style of the bottom border
border-bottom-width	Sets the width of the bottom border
border-collapse	Sets whether table borders should collapse into a single border or be separated
border-color	Sets the color of the four borders
border-image	A shorthand property for all the <i>border-image-*</i> properties
border-image-outset	Specifies the amount by which the border image area extends beyond the border box
border-image-repeat	Specifies whether the border image should be repeated, rounded or stretched
border-image-slice	Specifies how to slice the border image
border-image-source	Specifies the path to the image to be used as a border
border-image-width	Specifies the width of the border image

[border-inline](#)

A shorthand property for *border-inline-width*, *border-inline-style* and *border-inline-color*

[border-inline-color](#)

Sets the color of the borders at start and end in the inline direction

[border-inline-end-color](#)

Sets the color of the border at the end in the inline direction

[border-inline-end-style](#)

Sets the style of the border at the end in the inline direction

[border-inline-end-width](#)

Sets the width of the border at the end in the inline direction

[border-inline-start-color](#)

Sets the color of the border at the start in the inline direction

[border-inline-start-style](#)

Sets the style of the border at the start in the inline direction

[border-inline-start-width](#)

Sets the width of the border at the start in the inline direction

[border-inline-style](#)

Sets the style of the borders at start and end in the inline direction

[border-inline-width](#)

Sets the width of the borders at start and end in the inline direction

[border-left](#)

A shorthand property for all the *border-left-** properties

border-left-color	Sets the color of the left border
border-left-style	Sets the style of the left border
border-left-width	Sets the width of the left border
border-radius	A shorthand property for the four <i>border-*-radius</i> properties
border-right	A shorthand property for all the <i>border-right-*</i> properties
border-right-color	Sets the color of the right border
border-right-style	Sets the style of the right border
border-right-width	Sets the width of the right border
border-spacing	Sets the distance between the borders of adjacent cells
border-style	Sets the style of the four borders
border-top	A shorthand property for <i>border-top-width</i> , <i>border-top-style</i> and <i>border-top-color</i>

border-top-color	Sets the color of the top border
border-top-left-radius	Defines the radius of the border of the top-left corner
border-top-right-radius	Defines the radius of the border of the top-right corner
border-top-style	Sets the style of the top border
border-top-width	Sets the width of the top border
border-width	Sets the width of the four borders
bottom	Sets the elements position, from the bottom of its parent element
box-decoration-break	Sets the behavior of the background and border of an element at page-break, or, for in-line elements, at line-break.
box-reflect	The box-reflect property is used to create a reflection of an element.
box-shadow	Attaches one or more shadows to an element
box-sizing	Defines how the width and height of an element are calculated: should they include padding and borders, or not

break-after	Specifies whether or not a page-, column-, or region-break should occur after the specified element
break-before	Specifies whether or not a page-, column-, or region-break should occur before the specified element
break-inside	Specifies whether or not a page-, column-, or region-break should occur inside the specified element

C

caption-side	Specifies the placement of a table caption
caret-color	Specifies the color of the cursor (caret) in inputs, textareas, or any element
@charset	Specifies the character encoding used in the style sheet
clear	Specifies what should happen with the element that is next to a floating element
clip	Clips an absolutely positioned element
color	Sets the color of text
column-count	Specifies the number of columns an element should be divided into

column-fill	Specifies how to fill columns, balanced or not
column-gap	Specifies the gap between the columns
column-rule	A shorthand property for all the <i>column-rule-*</i> properties
column-rule-color	Specifies the color of the rule between columns
column-rule-style	Specifies the style of the rule between columns
column-rule-width	Specifies the width of the rule between columns
column-span	Specifies how many columns an element should span across
column-width	Specifies the column width
columns	A shorthand property for <i>column-width</i> and <i>column-count</i>
content	Used with the :before and :after pseudo-elements, to insert generated content
counter-increment	Increases or decreases the value of one or more CSS counters

[counter-reset](#)

Creates or resets one or more CSS counters

[cursor](#)

Specifies the mouse cursor to be displayed when pointing over an element

D

[direction](#)

Specifies the text direction/writing direction

[display](#)

Specifies how a certain HTML element should be displayed

E

[empty-cells](#)

Specifies whether or not to display borders and background on empty cells in a table

F

[filter](#)

Defines effects (e.g. blurring or color shifting) on an element before the element is displayed

[flex](#)

A shorthand property for the *flex-grow*, *flex-shrink*, and the *flex-basis* properties

[flex-basis](#)

Specifies the initial length of a flexible item

[flex-direction](#)

Specifies the direction of the flexible items

flex-flow	A shorthand property for the <i>flex-direction</i> and the <i>flex-wrap</i> properties
flex-grow	Specifies how much the item will grow relative to the rest
flex-shrink	Specifies how the item will shrink relative to the rest
flex-wrap	Specifies whether the flexible items should wrap or not
float	Specifies whether an element should float to the left, right, or not at all
font	A shorthand property for the <i>font-style</i> , <i>font-variant</i> , <i>font-weight</i> , <i>font-size/line-height</i> , <i>family</i> properties
@font-face	A rule that allows websites to download and use fonts other than the "web-safe" fonts
font-family	Specifies the font family for text
font-feature-settings	Allows control over advanced typographic features in OpenType fonts

CSS Formatting Elements:

Using Another Default Font

Using another default font in W3.CSS is very easy:

Example

```
body, h1, h2, h3, h4, h5, h6 {  
  font-family: Arial, Helvetica, sans-serif;  
}
```

Using Font Classes

With W3.CSS you can use 4 different built-in font classes:

- w3-serif
 - w3-sans-serif
 - w3-monospace
-
- w3-cursive

This is w3-serif

Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.

Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.

Example

```
<p class="w3-serif">
```

This is w3-sans-serif

Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.

Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.

Example

```
<p class="w3-sans-serif">
```

This is w3-monospace

Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.

Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.

Example

```
<div class="w3-monospace">
```

This is w3-cursive

Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.

Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.

Example

```
<div class="w3-cursive">
```

Font Family	Safe Fallback
Arial	Helvetica, sans-serif
Helvetica	Arial, sans-serif
Verdana	Geneva, sans-serif
Geneva	Verdana, sans-serif

Tahoma	Arial, sans-serif
Trebuchet MS	Arial, sans-serif

Arial is considered the safest sans-serif font on the web. It is installed on all Windows machines (99.8%) and all Mac machines (98.7%).

Helvetica is the preferred fallback because the two fonts are almost identical. Helvetica runs on all Mac machines (100%) but on fewer Windows machines (7.3%).

Using Arial with Helvetica fallback, or Helvetica with Arial fallback, covers both worlds.

Verdana is also considered to be a web safe sans-serif font. It runs on all Windows machines (99.8%), and all Mac Machines (99.1%).

Geneva is Verdanas preferred fallback, because the two fonts are quite similar, and Geneva runs on all Mac machines (99.6%).

Tahoma was the standard font in Windows 95. It works on 99.9% of all Windows machines, and 91.7% of all Mac machines.

Trebuchet MS has been released with all Windows operating systems (99.7%) since Microsoft Windows 2000. It is also included on Macs machines (97.1%), iOS and Chrome OS.

Source: [CSS Font Stack](#)

This is Arial

Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.

Norway has a total area of 385,252 square kilometers and a population of 5,438,657 (December 2020). Norway is bordered by Sweden, Finland and Russia to the north-east, and the Skagerrak to the south, with Denmark on the other side.

Example

```
<style>
body, h1, h2, h3, h4, h5, h6 {
  font-family: Arial, Helvetica, sans-serif;
```

```
}  
</style>
```

CSS Selectors

A CSS selector selects the HTML element(s) you want to style.

CSS Selectors

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

We can divide CSS selectors into five categories:

- Simple selectors (select elements based on name, id, class)
- [Combinator selectors](#) (select elements based on a specific relationship between them)
- [Pseudo-class selectors](#) (select elements based on a certain state)
- [Pseudo-elements selectors](#) (select and style a part of an element)
- [Attribute selectors](#) (select elements based on an attribute or attribute value)

This page will explain the most basic CSS selectors.

The CSS element Selector

The element selector selects HTML elements based on the element name.

Example

Here, all <p> elements on the page will be center-aligned, with a red text color:

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

The CSS id Selector

The id selector uses the id attribute of an HTML element to select a specific element.

The id of an element is unique within a page, so the id selector is used to select one unique element!

To select an element with a specific id, write a hash (#) character, followed by the id of the element.

Example

The CSS rule below will be applied to the HTML element with id="para1":

```
#para1 {  
  text-align: center;  
  color: red;  
}
```

The CSS class Selector

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

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

Example

In this example all HTML elements with class="center" will be red and center-aligned:

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

You can also specify that only specific HTML elements should be affected by a class.

Example

In this example only <p> elements with class="center" will be red and center-aligned:

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

HTML elements can also refer to more than one class.

Example

In this example the <p> element will be styled according to class="center" and to class="large":

```
<p class="center large">This paragraph refers to two classes.</p>
```

The CSS Universal Selector

The universal selector (*) selects all HTML elements on the page.

Example

The CSS rule below will affect every HTML element on the page:

```
* {  
  text-align: center;  
  color: blue;  
}
```

The CSS Grouping Selector

The grouping selector selects all the HTML elements with the same style definitions.

Look at the following CSS code (the h1, h2, and p elements have the same style definitions):

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

```
h2 {  
  text-align: center;  
  color: red;  
}
```

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

It will be better to group the selectors, to minimize the code.

To group selectors, separate each selector with a comma.

Example

In this example we have grouped the selectors from the code above:

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```

CSS Box Model

All HTML elements can be considered as boxes.

The CSS Box Model

In CSS, the term "box model" is used when talking about design and layout.

The CSS box model is essentially a box that wraps around every HTML element. It consists of: margins, borders, padding, and the actual content. The image below illustrates the box model:

Explanation of the different parts:

- **Content** - The content of the box, where text and images appear
- **Padding** - Clears an area around the content. The padding is transparent
- **Border** - A border that goes around the padding and content
- **Margin** - Clears an area outside the border. The margin is transparent

The box model allows us to add a border around elements, and to define space between elements.

Example

Demonstration of the box model:

```
div {  
  width: 300px;  
  border: 15px solid green;  
  padding: 50px;  
  margin: 20px;  
}
```

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**
- **sticky**

Elements are then positioned using the top, bottom, left, and right properties. However, these properties will not work unless the **position** property is set first. They also work differently depending on the position value.

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:

This <div> element has position: static;

Here is the CSS that is used:

Example

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

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.

This <div> element has position: relative;

Here is the CSS that is used:

Example

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

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.

Notice the fixed element in the lower-right corner of the page. Here is the CSS that is used:

Example

```
div.fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 300px;
```

```
border: 3px solid #73AD21;
}
```

This <div> element has **position: fixed;**

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: Absolute positioned elements are removed from the normal flow, and can overlap elements.

Here is a simple example:

This <div> element has position: relative;

This <div> element has position: absolute;

Here is the CSS that is used:

Example

```
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;
}
```

position: sticky;

An element with **position: sticky;** is positioned based on the user's scroll position.

A sticky element toggles between **relative** and **fixed**, depending on the scroll position. It is positioned relative until a given offset position is met in the viewport - then it "sticks" in place (like position:fixed).

Note: Internet Explorer does not support sticky positioning. Safari requires a `-webkit-` prefix (see example below). You must also specify at least one of `top`, `right`, `bottom` or `left` for sticky positioning to work.

In this example, the sticky element sticks to the top of the page (`top: 0`), when you reach its scroll position.

Example

```
div.sticky {  
  position: -webkit-sticky; /* Safari */  
  position: sticky;  
  top: 0;  
  background-color: green;  
  border: 2px solid #4CAF50;  
}
```

Positioning Text In an Image

How to position text over an image:

Example



Bottom Left

Top Left

Top Right

Bottom Right

Centered

Try it Yourself:

[Top Left »](#) [Top Right »](#) [Bottom Left »](#) [Bottom Right »](#) [Centered »](#)

More Examples

[Set the shape of an element](#)

This example demonstrates how to set the shape of an element. The element is clipped into this shape, and displayed.

Exercise:

Position the `<h1>` element to always be 50px from the top, and 10px from the right, relative to the window/frame edges.

```
<style>
h1 {
  : ;
  : 50px;
  : 10px;
}
</style>

<body>
  <h1>This is a heading</h1>
  <p>This is a paragraph</p>
  <p>This is a paragraph</p>
</body>
```

What are Pseudo-classes?

A pseudo-class is used to define a special state of an element.

For example, it can be used to:

- Style an element when a user mouse over it
- Style visited and unvisited links differently
- Style an element when it gets focus

Syntax

The syntax of pseudo-classes:

```
selector:pseudo-class {  
  property: value;  
}
```

Anchor Pseudo-classes

Links can be displayed in different ways:

Example

```
/* unvisited link */  
a:link {  
  color: #FF0000;  
}  
  
/* visited link */  
a:visited {  
  color: #00FF00;  
}  
  
/* mouse over link */  
a:hover {  
  color: #FF00FF;  
}  
  
/* selected link */  
a:active {  
  color: #0000FF;  
}
```

Pseudo-classes and HTML Classes

Pseudo-classes can be combined with HTML classes:

When you hover over the link in the example, it will change color:

Example

```
a.highlight:hover {  
  color: #ff0000;  
}
```

Hover on <div>

An example of using the `:hover` pseudo-class on a `<div>` element:

Example

```
div:hover {  
  background-color: blue;  
}
```

Simple Tooltip Hover

Hover over a `<div>` element to show a `<p>` element (like a tooltip):

Hover over me to show the `<p>` element.

Example

```
p {  
  display: none;  
  background-color: yellow;  
  padding: 20px;  
}
```

```
div:hover p {  
  display: block;  
}
```

CSS - The `:first-child` Pseudo-class

The `:first-child` pseudo-class matches a specified element that is the first child of another element.

Match the first `<p>` element

In the following example, the selector matches any `<p>` element that is the first child of any element:

Example

```
p:first-child {  
  color: blue;  
}
```

Match the first `<i>` element in all `<p>` elements

In the following example, the selector matches the first `<i>` element in all `<p>` elements:

Example

```
p i:first-child {  
  color: blue;  
}
```

Match all `<i>` elements in all first child `<p>` elements

In the following example, the selector matches all `<i>` elements in `<p>` elements that are the first child of another element:

Example

```
p:first-child i {  
  color: blue;  
}
```

CSS - The `:lang` Pseudo-class

The `:lang` pseudo-class allows you to define special rules for different languages.

In the example below, `:lang` defines the quotation marks for `<q>` elements with `lang="no"`:

Example

```
<html>  
<head>  
<style>  
q:lang(no) {  
  quotes: "~" "~";  
}  
</style>  
</head>  
<body>
```

```
<p>Some text <q lang="no">A quote in a paragraph</q> Some text.</p>
```

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

Multiple style sheet in CSS

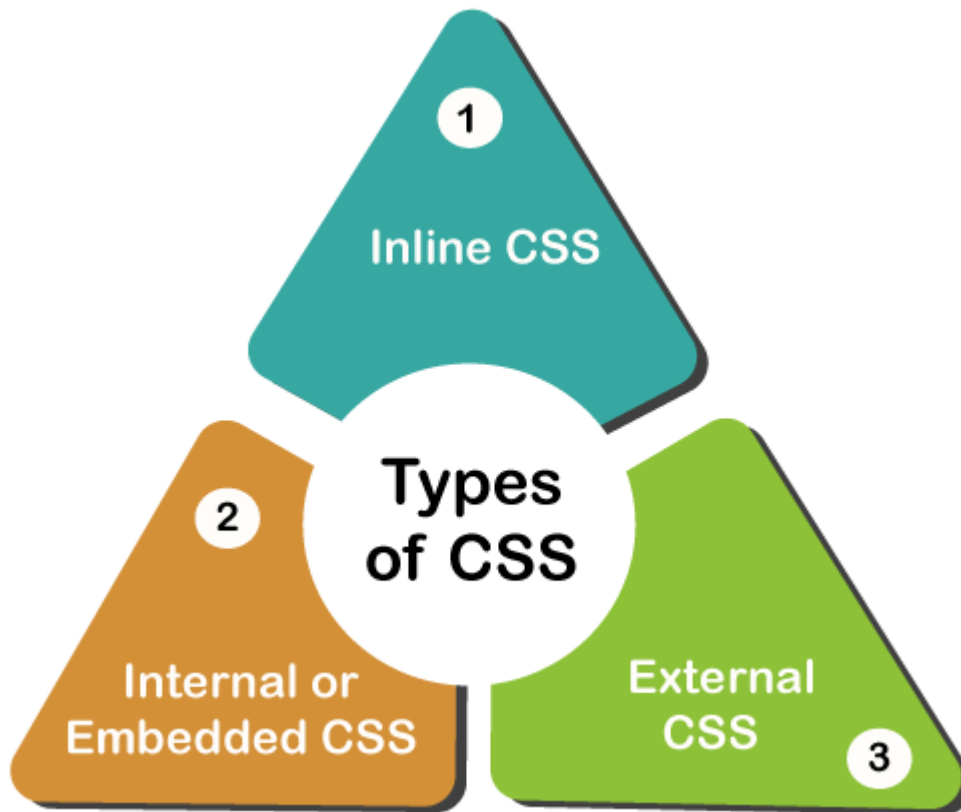
CSS (Cascading Style Sheet) describes the HTML elements which are displayed on **screen, paper**, or in **other media**. It saves a lot of time. It controls the layout of multiple web pages at one time. It sets the **font-size, font-family, color, background color** on the page.

It allows us to add **effects** or **animations** to the website. We use **CSS** to display **animations** like **buttons, effects, loaders** or **spinners**, and also **animated backgrounds**.

Without using **CSS**, the website will not look attractive. There are **3** types of **CSS** which are below:

- Inline CSS
- Internal/ Embedded CSS

- External CSS



1. Internal CSS

The Internal CSS has `<style>` tag in the `<head>` section of the **HTML** document. This CSS style is an effective way to style single pages. Using the CSS style for multiple web pages is time-consuming because we require placing the **style** on each web page.

We can use the internal CSS by using the following steps:

1. Firstly, open the **HTML** page and locate the `<head>`
2. Put the following code after the `<head>`

1. `<style type="text/css">`

3. Add the **rules** of CSS in the new line.

Example:

1. `body {`
2. `background-color: black;`

```
3. }
4. h1 {
5.   color: white;
6.   padding: 50px;
7. }
```

4. Close the style tag.

```
1. </style>
```

After adding the internal CSS, the complete [HTML](#) file looks like the following:

```
1. <!DOCTYPE html>
2. <html>
3. <head>
4. <style>
5. body {
6.   background-color: black;
7. }
8. h1 {
9.   color: red;
10.  padding: 50px;
11. }
12. </style>
13. </head>
14. <body>
15. <h2>CSS types</h2>
16. <p>Cascading Style sheet types: inline, external and internal</p>
17. </body>
18. </html>
```

We can also use the selectors (**class and ID**) in the style sheet.

Example:

```
1. .class {
2.   property1 : value1;
3.   property2 : value2;
4.   property3 : value3;
5. }
6.
7. #id {
8.   property1 : value1;
9.   property2 : value2;
10.  property3 : value3;
11. }
```

Pros of Internal CSS

- **Internal CSS** cannot upload multiple files when we add the code with the HTML page.

Cons of Internal CSS:

- Adding code in the **HTML** document will reduce the **page size** and **loading time** of the webpage.

2. External CSS

In [external CSS](#), we link the web pages to the external **.css** file. It is created by **text editor**. The CSS is more efficient method for styling a website. By editing the **.css** file, we can change the whole site at once.

To use the external CSS, follow the steps, given below:

1. Create a new **.css** file with **text editor**, and add **Cascading Style Sheet** rules too.

For example:

1. `.xleftcol {`
2. `float: right;`
3. `width: 35%;`
4. `background:#608800;`
5. `}`
6. `.xmiddlecol {`
7. `float: right;`
8. `width: 35%;`
9. `background:#eff3df;`
10. `}`

2. Add a reference to the external **.css** file right after **<title> tag** in the **<head>** section of **HTML sheet**:

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

Pros of External CSS:

- Our files have a cleaner structure and smaller in size.
- We use the same **.css** file for multiple web pages in external CSS.

Cons of External CSS:

- The pages cannot be delivered correctly before the external CSS is loaded.
- In External CSS, uploading many CSS files can increase the download time of a website.

3. Inline CSS

Inline CSS is used to style a specific **HTML** element. Add a **style** attribute to each HTML tag without using the selectors. Managing a website may difficult if we use only **inline CSS**. However, Inline **CSS** in HTML is useful in some situations. We have not access the **CSS files** or to apply styles to element.

In the following example, we have used the inline CSS in `<p>` and `<h1>` tag.

1. `<!DOCTYPE html>`
2. `<html>`
3. `<body style="background-color:white;">`
4. `<h1 style="color:Red;padding:20px;">CSS Tutorials</h1>`
5. `<p style="color:blue;">It will be useful here.</p>`
6. `</body>`
7. `</html>`

Pros of inline CSS:

- We can create CSS rules on the HTML page.
- We cannot create and upload a separate document in inline CSS.

Cons of inline CSS:

- Inline CSS, adding **CSS** rules to HTML elements is **time-consuming** and **messes up** the HTML structure.
- It styles multiple elements at the same time which can affect the page size and download time of the page.

CSS Links

Styling Links

Links can be styled with any CSS property (e.g. **color**, **font-family**, **background**, etc.).

Example

```
a {  
  color: hotpink;  
}
```

In addition, links can be styled differently depending on what **state** they are in.

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

Example

```
/* 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;  
}
```

When setting the style for several link states, there are some order rules:

- `a:hover` MUST come after `a:link` and `a:visited`
- `a:active` MUST come after `a:hover`

Text Decoration

The `text-decoration` property is mostly used to remove underlines from links:

Example

```
a:link {  
  text-decoration: none;  
}
```

```
a:visited {  
  text-decoration: none;  
}
```

```
a:hover {  
  text-decoration: underline;  
}
```

```
a:active {
  text-decoration: underline;
}
```

Background Color

The `background-color` property can be used to specify a background color for links:

Example

```
a:link {
  background-color: yellow;
}

a:visited {
  background-color: cyan;
}

a:hover {
  background-color: lightgreen;
}

a:active {
  background-color: hotpink;
}
```

Link Buttons

This example demonstrates a more advanced example where we combine several CSS properties to display links as boxes/buttons:

Example

```
a:link, a:visited {
  background-color: #f44336;
  color: white;
  padding: 14px 25px;
  text-align: center;
  text-decoration: none;
  display: inline-block;
}

a:hover, a:active {
  background-color: red;
}
```

CSS background Property

Example

Set different background properties in one declaration:

```
body {  
  background: lightblue url("img_tree.gif") no-repeat fixed center;  
}
```

Definition and Usage

The **background** property is a shorthand property for:

- [background-color](#)
- [background-image](#)
- [background-position](#)
- [background-size](#)
- [background-repeat](#)
- [background-origin](#)
- [background-clip](#)
- [background-attachment](#)

It does not matter if one of the values above are missing, e.g. `background:#ff0000 url(smiley.gif);` is allowed.

Default value:	<i>see individual properties</i>
-----------------------	----------------------------------

Inherited:	No
-------------------	----

Animatable:	yes, <i>see individual properties</i> . Read about animatable Try it
--------------------	--

Version:	CSS1 + new properties in CSS3
-----------------	-------------------------------

JavaScript syntax:	<code>object.style.background="red url(smiley.gif) top left no-repeat"</code> Try it
---------------------------	--

CSS Tables

The look of an HTML table can be greatly improved with CSS:

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy

Table Borders

To specify table borders in CSS, use the **border** property.

The example below specifies a solid border for <table>, <th>, and <td> elements:

Firstname	Lastname
Peter	Griffin
Lois	Griffin

Example

```
table, th, td {  
  border: 1px solid;  
}
```

Full-Width Table

The table above might seem small in some cases. If you need a table that should span the entire screen (full-width), add `width: 100%` to the <table> element:

Firstname	Lastname
Peter	Griffin
Lois	Griffin

Example

```
table {  
  width: 100%;  
}
```

Double Borders

Notice that the table in the examples above have double borders. This is because both the table and the <th> and <td> elements have separate borders.

To remove double borders, take a look at the example below.

Collapse Table Borders

The `border-collapse` property sets whether the table borders should be collapsed into a single border:

Lois	Griffin
------	---------

Example

```
table {
```

```
  border-collapse: collapse;
}
```

If you only want a border around the table, only specify the `border` property for `<table>`:

Firstname	Lastname
Peter	Griffin
Lois	Griffin

Example

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