**CSS (Cascading Style Sheets)**

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language like HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.

CSS is designed to enable the separation of presentation and content, including layout, colors, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content.

## **History**

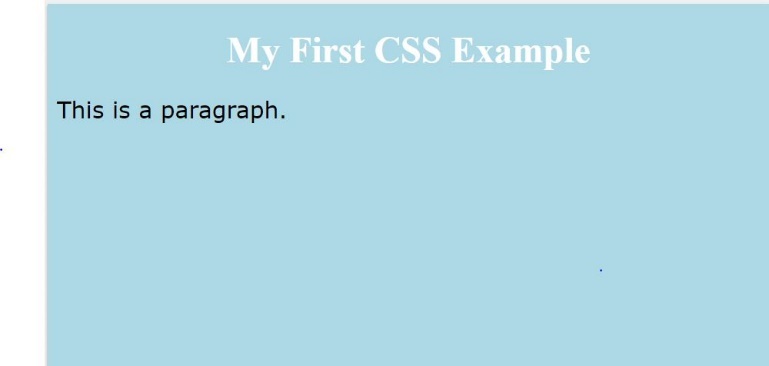
CSS was first proposed by Håkon Wium Lie on October 10, 1994. At the time, Lie was working with Tim Berners-Lee at CERN. Several other style sheet languages for the web were proposed around the same time, and discussions on public mailing lists and inside World Wide Web Consortium resulted in the first W3C CSS Recommendation (CSS1) being released in 1996. In particular, a proposal by Bert Bos was influential; he became co-author of CSS1, and is regarded as co-creator of CSS.

Style sheets have existed in one form or another since the beginnings of Standard Generalized Markup Language (SGML) in the 1980s, and CSS was developed to provide style sheets for the web. One requirement for a web style sheet language was for style sheets to come from different sources on the web. Therefore, existing style sheet languages like DSSSL and FOSI were not suitable. CSS, on the other hand, let a document's style be influenced by multiple style sheets by way of "cascading" styles.

CSS Example

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

**Output:**

. 

Cascading Style Sheets (CSS) is a stylesheet language used to describe the presentation of a document written in HTML or XML (including XML dialects such as SVG, MathML or XHTML). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is one of the core languages of the open Web and is standardized across Web browsers according to the W3C specification. Developed in levels, CSS1 is now obsolete, CSS2.1 is a recommendation, and CSS3, now split into smaller modules, is progressing on the standardization track.

**CSS** is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning CSS:

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

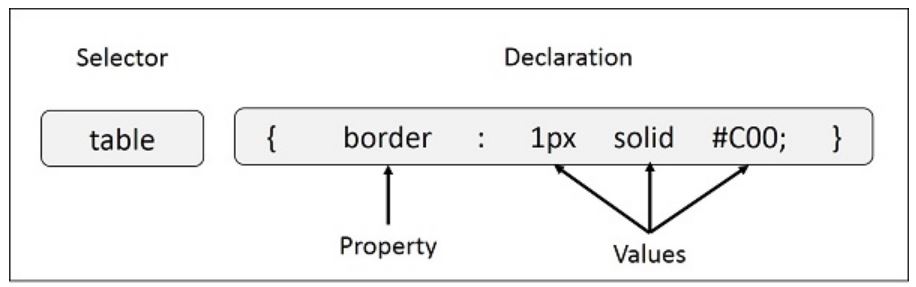
## **Syntax**

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

* **Selector** − A selector is an HTML tag at which a style will be applied. This could be any tag like <h1> or <table> etc.
* **Property** − A property is a type of attribute of HTML tag. Put simply, all the HTML attributes are converted into CSS properties. They could be *color*, *border* etc.
* **Value** − Values are assigned to properties. For example, *color* property can have value either *red* or *#F1F1F1* etc.

You can put CSS Style Rule Syntax as follows –





The Type 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](https://www.w3schools.com/css/css_combinators.asp) (select elements based on a specific relationship between them)
* [Pseudo-class selectors](https://www.w3schools.com/css/css_pseudo_classes.asp) (select elements based on a certain state)
* [Pseudo-elements selectors](https://www.w3schools.com/css/css_pseudo_elements.asp) (select and style a part of an element)
* [Attribute selectors](https://www.w3schools.com/css/css_attribute_selectors.asp) (select elements based on an attribute or attribute value)

## **The Universal Selectors**

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



## **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:**

<!DOCTYPE html>

<html>

<head>

<style>

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

</style>

</head>

<body>

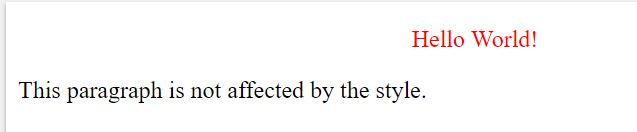
<p id="para1">Hello World!</p>

<p>This paragraph is not affected by the style.</p>

</body>

</html>

**Output**



## **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**

<!DOCTYPE html>

<html>

<head>

<style>

.center {

text-align: center;

color: red;

}

</style>

</head>

<body>

<h1 class="center">GNDU</h1>

<p class="center">Guru Nanak Dev University</p> </body>

</html>



## **Grouping Selectors**

You can apply a style to many selectors if you like. Just separate the selectors with a comma, as given in the following example −

h1, h2, h3 {

color: #36C;

font-weight: normal;

letter-spacing: .4em;

margin-bottom: 1em;

text-transform: lowercase;

}

This define style rule will be applicable to h1, h2 and h3 element as well. The order of the list is irrelevant. All the elements in the selector will have the corresponding declarations applied to them.

## **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.

**Example:**

<!DOCTYPE html>

<html>

<head>

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

</head>

<body>

<h1>GNDU</h1>

<p>Guru Nanak Dev University</p>

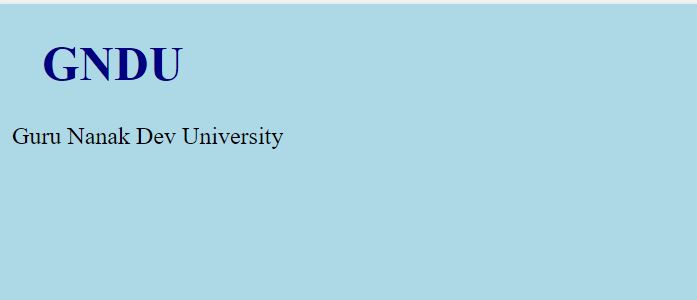
</body>

</html>

### "mystyle.css"



**Output:**



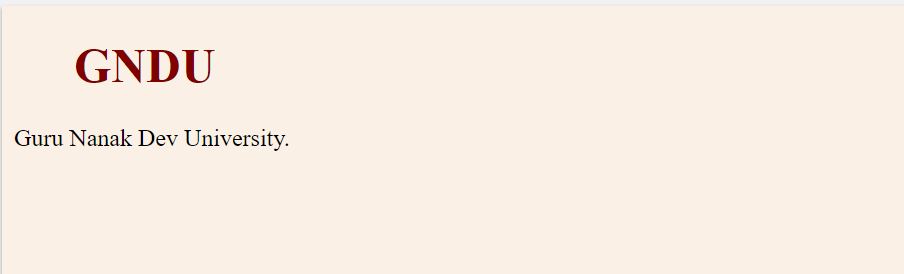
1. **Internal CSS**

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.

<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {  
  background-color: linen;  
}  
  
h1 {  
  color: maroon;  
  margin-left: 40px;  
}  
</style>  
</head>  
<body>  
  
<h1>GNDU</h1>  
<p>Guru Nanak Dev University.</p>  
  
</body>  
</html>

**Output:**



1. **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:blue;text-align:center;">This is a heading</h1>  
<p style="color:red;">This is a paragraph.</p>  
</body>  
</html>

**Output:**



**CSS Comments:**

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

Comments are ignored by browsers.

A CSS comments starts with /\* and ends with \*/.

p {  
  color: red;  
  /\* This is a single-line comment \*/  
  text-align: center;  
}

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

**CSS Properties:**

**CSS Text Color:**

<h1 style="color:Tomato;">Hello World</h1>

**Output:**

### Hello World

### CSS Background Color:

### <h1 style="background-color:DodgerBlue;">Hello World</h1>

### 

### CSS Color Values:

### In CSS, colors can also be specified using RGB values, HEX values, HSL values, RGBA values, and HSLA values:

### Example:

### <h1 style="background-color:rgb(255, 99, 71);">...</h1> <h1 style="background-color:#ff6347;">...</h1> <h1 style="background-color:hsl(9, 100%, 64%);">...</h1>

### 

### <h1 style="background-color:rgba(255, 99, 71, 0.5);">...</h1> <h1 style="background-color:hsla(9, 100%, 64%, 0.5);">...</h1>

### Same as color name "Tomato", but 50% transparent:

### 

### CSS background-image:

### body {   background-image: url("paper.gif"); }

### CSS background-attachment:

### The background-attachment property specifies whether the background image should scroll or be fixed (will not scroll with the rest of the page):

### body {   background-image: url("img\_tree.png");   background-repeat: no-repeat;   background-position: right top;   background-attachment: fixed; }

### CSS background - Shorthand property:

### body {   background: #ffffff url("img\_tree.png") no-repeat right top; }

### CSS Border Properties:

The border-style property specifies what kind of border to display.

The following values are allowed:

* dotted - Defines a dotted border
* dashed - Defines a dashed border
* solid - Defines a solid border
* double - Defines a double border
* groove - Defines a 3D grooved border. The effect depends on the border-color value
* ridge - Defines a 3D ridged border. The effect depends on the border-color value
* inset - Defines a 3D inset border. The effect depends on the border-color value
* outset - Defines a 3D outset border. The effect depends on the border-color value
* none - Defines no border
* hidden - Defines a hidden border

The border property is a shorthand property for the following individual border properties:

* border-width
* border-style (required)
* border-color

### Example:

### p {   border: 2px solid red;   border-radius: 5px; }

### Output:

### 

### CSS Margins:

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

* margin-top
* margin-right
* margin-bottom
* margin-left

### Example:

### <html>

### <head>

### <style>

### div {

### border: 1px solid black;

### margin-top: 100px;

### margin-bottom: 100px;

### margin-right: 150px;

### margin-left: 80px;

### background-color: lightblue;

### }

### </style>

### </head>

### <body>

### <h2>Using individual margin properties</h2>

### <div>This div element has a top margin of 100px, a right margin of 150px, a bottom margin of 100px, and a left margin of 80px.</div>

### </body>

### </html>

### Output:

### 

### CSS Padding:

### The CSS padding properties are used to generate space around an element's content, inside of any defined borders.

### With CSS, you have full control over the padding. There are properties for setting the padding for each side of an element (top, right, bottom, and left).

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

### padding-top

### padding-right

### padding-bottom

### padding-left

### All the padding properties can have the following values:

### *length* - specifies a padding in px, pt, cm, etc.

### *%* - specifies a padding in % of the width of the containing element

### inherit - specifies that the padding should be inherited from the parent element

### Note: Negative values are not allowed.

### Example:

### <!DOCTYPE html>

### <html>

### <head>

### <style>

### div.ex1 {

### width: 300px;

### background-color: yellow;

### }

### div.ex2 {

### width: 300px;

### padding: 25px;

### background-color: lightblue;

### }

### </style>

### </head>

### <body>

### <h2>Padding and element width</h2>

### <div class="ex1">This div is 300px wide.</div>

### <br>

### <div class="ex2">The width of this div is 350px, even though it is defined as 300px in the CSS.</div>

### </body>

### </html>

### Output:

### 

### CSS Text:

### The color property is used to set the color of the text.

### The text-align property is used to set the horizontal alignment of a text.

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

### The text-transform property is used to specify uppercase and lowercase letters in a text.

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

### The text-shadow property adds shadow to text.

### Example:

### <!DOCTYPE html>

### <html>

### <head>

### <style>

### div {

### border: 1px solid gray;

### padding: 8px;

### }

### h1 {

### text-align: center;

### text-transform: uppercase;

### color: #4CAF50;

### }

### p {

### text-indent: 50px;

### text-align: justify;

### letter-spacing: 3px;

### }

### a {

### text-decoration: none;

### color: #008CBA;

### }

### </style>

### </head>

### <body>

### <div>

### <h1>GNDU</h1>

### <p>GNDU's journey started in January 2005 with a vision of bringing World Class Technical Education to the holy city of Amritsar. To turn this dream into reality we create a talent pool of bright young minds who would power the engines of growth of the global economy. <a target="\_blank" href="tryit.asp?filename=trycss\_text">"Try it Yourself"</a> link.</p>

### </div>

### </body>

### </html>

### Output:

### 

### CSS Fonts:

### The font family of a text is set with the font-family property.

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)

### The font-size property sets the size of the text.

### The font-weight property specifies the weight of a font.

### Example:

### <!DOCTYPE html>

### <html>

### <head>

### <style>

### p.serif {

### font-family: "Times New Roman", Times, serif;

### font-style: italic;

### }

### p.sansserif {

### font-family: Arial, Helvetica, sans-serif;

### font-size: 30px;

### }

### </style>

### </head>

### <body>

### <h1>GNDU</h1>

### <p class="serif">The University is successfully catering to twenty thousand students in various faculties at University Campuses and Constituent Colleges.</p>

### <p class="sansserif">This is a paragraph, shown in the Arial font.</p>

### </body>

### </html>

### Output:

### 

### CSS 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:

### <!DOCTYPE html>

### <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>

### <body>

### <p><b><a href="default.asp" target="\_blank">This is a link</a></b></p>

### <p><b>Note:</b> a:hover MUST come after a:link and a:visited in the CSS definition in order to be effective.</p>

### <p><b>Note:</b> a:active MUST come after a:hover in the CSS definition in order to be effective.</p>

### </body>

### </html>

### Output:

### CSS List Properties:

Lists are very helpful in conveying a set of either numbered or bullet points. This chapter teaches you how to control list type, position, style, etc., using CSS.

We have the following five CSS properties, which can be used to control lists −

* The **list-style-type** allows you to control the shape or appearance of the marker.
* The **list-style-position** specifies whether a long point that wraps to a second line should align with the first line or start underneath the start 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.
* The **marker-offset** specifies the distance between a marker and the text in the list.

### Example:

[Live Demo](http://tpcg.io/9LSg6t)

<html>

<head>

</head>

<body>

<ul style = "list-style-type:circle;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ul>

<ul style = "list-style-type:square;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ul>

<ol style = "list-style-type:decimal;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

<ol style = "list-style-type:lower-alpha;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

<ol style = "list-style-type:lower-roman;">

<li>Maths</li>

<li>Social Science</li>

<li>Physics</li>

</ol>

</body>

</html>

### Output:

### 

### CSS Tables:

This tutorial will teach you how to set different properties of an HTML table using CSS. You can set following properties of a table −

* The **border-collapse** specifies whether the browser should control the appearance of the adjacent borders that touch each other or whether each cell should maintain its style.
* The **border-spacing** specifies the width that should appear between table cells.
* The **caption-side** captions are presented in the <caption> element. By default, these are rendered above the table in the document. You use the *caption-side* property to control the placement of the table caption.
* The **empty-cells** specifies whether the border should be shown if a cell is empty.
* The **table-layout** allows browsers to speed up layout of a table by using the first width properties it comes across for the rest of a column rather than having to load the whole table before rendering it.

### Example:

[Live Demo](http://tpcg.io/2wJiMF)

<html>

<head>

<style type = "text/css">

table.one {border-collapse:collapse;}

table.two {border-collapse:separate;}

td.a {

border-style:dotted;

border-width:3px;

border-color:#000000;

padding: 10px;

}

td.b {

border-style:solid;

border-width:3px;

border-color:#333333;

padding:10px;

}

</style>

</head>

<body>

<table class = "one">

<caption>Collapse Border Example</caption>

<tr><td class = "a"> Cell A Collapse Example</td></tr>

<tr><td class = "b"> Cell B Collapse Example</td></tr>

</table>

<br />

<table class = "two">

<caption>Separate Border Example</caption>

<tr><td class = "a"> Cell A Separate Example</td></tr>

<tr><td class = "b"> Cell B Separate Example</td></tr>

</table>

</body>

</html>

### Output:

### 

### CSS Layout - The display Property:

The display property specifies if/how an element is displayed.

Every HTML element has a default display value depending on what type of element it is. The default display value for most elements is block or inline.

### display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them.

### visibility:hidden; also hides an element.

### li {   display: inline; }

### CSS Opacity / Transparency:

### The opacity property specifies the opacity/transparency of an element.

### img {   opacity: 0.5; }

### Output:

### 

**CSS Shadow Effects:**

With CSS you can add shadow to text and to elements.

In this chapter you will learn about the following properties:

* text-shadow
* box-shadow

The CSS text-shadow property applies shadow to text.

In its simplest use, you only specify the horizontal shadow (2px) and the vertical shadow (2px):

### h1 {   text-shadow: 2px 2px 5px red; }

### Output:

### 

The CSS box-shadow property applies shadow to elements.

In its simplest use, you only specify the horizontal shadow and the vertical shadow:

### div {   box-shadow: 10px 10px; }

### Output:

### 

### CSS Media Queries:

### The @media rule, introduced in CSS2, made it possible to define different style rules for different media types.

One way to use media queries is to have an alternate CSS section right inside your style sheet.

The following example changes the background-color to lightgreen if the viewport is 480 pixels wide or wider (if the viewport is less than 480 pixels, the background-color will be pink):

@media screen and (min-width: 480px) {  
  body {  
    background-color: lightgreen;  
  }  
}