

Cascading Style Sheets

Cascading Style Sheets (CSS) is a fantastic tool to add layout to your websites. It saves lot of time and enables you to design websites in a different way. CSS is an important concept for people who are working on designing web pages.

Learning CSS is fun. Going further, take enough time to experiment with what you learnt in each session. Using CSS requires basic experience with HTML. If you are not familiar with HTML, be aware of required concepts and then start using CSS.

All you need is a free and simple text editor.

For example, Microsoft Windows comes with a program called Notepad. It is usually located in Accessories in the start menu under Programs. Alternatively, you can use a similar text editor e.g., vim, scite, gedit in linux operating system.

A simple text editor is ideal for learning HTML and CSS because it doesn't affect or change the codes you type. That way, your successes and errors can only be attributed to yourself - not the software.

You can use any of the browser. We encourage you to always keep your browser updated and use the latest version.

A browser and a simple text editor is all you need.

Let's get started!

What is CSS?

CSS is an acronym for Cascading Style Sheets.

CSS is a style language that defines layout of HTML documents.

For example, CSS covers fonts, colours, margins, lines, height, width, background images, advanced positions and many other things.

HTML can be (mis-)used to add layout to websites. But CSS offers more options and is more accurate and sophisticated. CSS is supported by all browsers today.

After this sessions you will be able to make your own style sheets using CSS to give your website a new great look.

What is the difference between CSS and HTML?

HTML is used to *structure content*. CSS is used for *formatting structured content*.

As the language HTML was only used to add structure to text. An author could mark his text by stating "this is a headline" or "this is a paragraph" using HTML tags such as `< h1 >` and `< p >`.

As the Web gained popularity, designers started looking for possibilities to add layout to online documents. To meet this demand, the browser producers (at that time Netscape and

Cascading Style Sheets

Microsoft) invented new HTML tags such as for example `< font >` which differed from the original HTML tags by defining layout - and not structure.

This also led to a situation where original structure tags such as `< table >` were increasingly being misused to layout pages instead of adding structure to text. Many new layout tags such as `< blink >` were only supported by one type of browser. "You need browser X to view this page" became a common disclaimer on web sites.

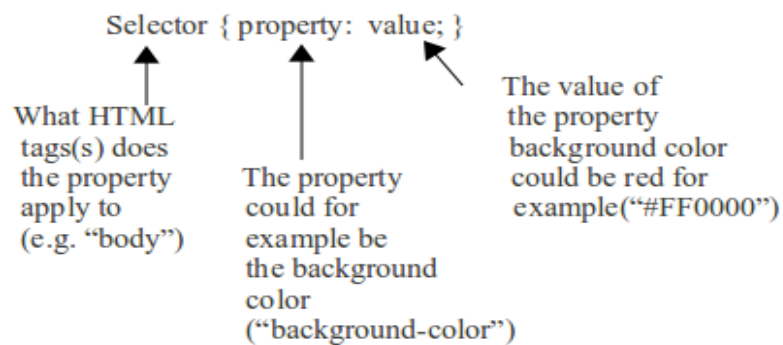
CSS was invented to remedy this situation by providing web designers with sophisticated layout opportunities supported by all browsers. At the same time, separate the presentation style of documents from the content of documents, makes site maintenance a lot easier.

How does CSS work?

In this section, we will learn how to make your first style sheet. You will get to know about the basic CSS model and which codes are necessary to use CSS in an HTML document.

Many of the properties used in Cascading Style Sheets (CSS) are similar to those of HTML. Thus, if you are used to use HTML for layout, you will most likely recognize many of the codes. Let us look at a concrete example.

Syntax:



- The selector is normally the HTML element(`< body >`,`< p >`,`< h >`....)you want to style.
- It can have more than one declaration and each consists of a property and a value.
- The property is the style attribute(color, font-size, text-align,.....) you want to change. Each property has a value(yellow, 15px, center,.....).
- Each declarations are separated by the symbol semicolon(;)and declaration groups are surrounded by curly brackets({...}).

Cascading Style Sheets

Example:

Applying red color as the background to a webpage:

Using HTML

```
<body bgcolor="#FF0000">
```

Using CSS

```
body {background-color: #FF0000;}
```

Benefits of CSS

CSS was a revolution in the world of web design. The concrete benefits of CSS include:

- control layout of many documents from one single style sheet.
- more precise control of layout.
- apply different layout to different media-types (screen, print, etc.).
- numerous advanced and sophisticated techniques.

Applying CSS to an HTML Document

When a browser reads a style sheet, it will format the document according to it. There are three ways to insert CSS to an HTML document.

There are three ways of inserting a style sheet:

- Inline style
- Internal style sheet
- External style sheet

Embedded CSS:

The `< style >` Element

Cascading Style Sheets

All the CSS rules can put together in an HTML document using the `< style >` element. This tag is placed inside `< head > ... < /head >` tags. Rules defined using this syntax will be applied to all the elements available in the document.

HTML permits any number of style elements in the head section of a document.

Syntax:

```
<head>
  <style type="text/css" media="...">
    Style Rules
    .....
  </style>
</head>
```

style This attribute specifies style information for the current element. Attributes associated with `< style >` elements are:

type = content-type This attribute specifies the style sheet language of the element's contents and overrides the default style sheet language. The style sheet language is specified as a content type (e.g., "text/css"). There is no default value for this attribute; we have to assign a value for this attribute. This is the required attribute.

media = media-descriptors This attribute specifies the intended destination medium for style information. It may be a single media descriptor or a comma-separated list. The default value for this attribute is "screen".

Example:

```
<html>
  <head>
    <style type="text/css" media="all">
      h1{color: #DD5600;}
    </style>
  </head>
  <body>
    <h1> Applying Styles to these text placed in h1 tag </h1>
    <p> These lines of text has no style.....<p>
  </body>
</html>
```



In-line style sheet

CSS styles can be assigned to individual HTML elements with the *style* attribute, Inline styles will have precedence over all other styles.

Syntax:

```
<tag style="property1: value; property2: value;"> </tag>
```

Example:

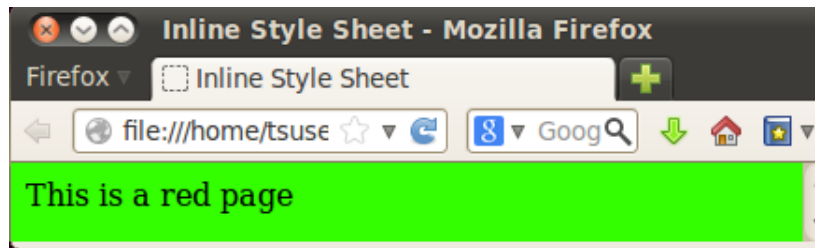
```
<h1 style="color: red; text-decoration: underline;">Hello</h1>
```

Sample Example:

Let's write a sample code how to use Inline style.

```
<html>
  <head>
    <title>Inline Style Sheet</title>
  </head>
  <body style="background-color: #FF0000;">
    <p>This is a red page</p>
  </body>
</html>
```

It displays as follows:

**Note**

Inline CSS should be avoided as it increases maintenance cost and blurs line between presentation and content.

So far, we were being used CSS through the *style* attribute, which is also known as inline styles, but inline styles don't separate content and style in a website. That's why internal and external style sheets are so important.

Before going further concept we have to learn about two more very important HTML tag attributes: *class=""* and *id=""* (used for identification and grouping of elements).

Identification and grouping of elements

If we want to apply a special style to a particular element or a particular group of elements. In addition to setting a style for a HTML element, CSS also allows to specify your own selectors called "**id**" and "**class**".

id Selector

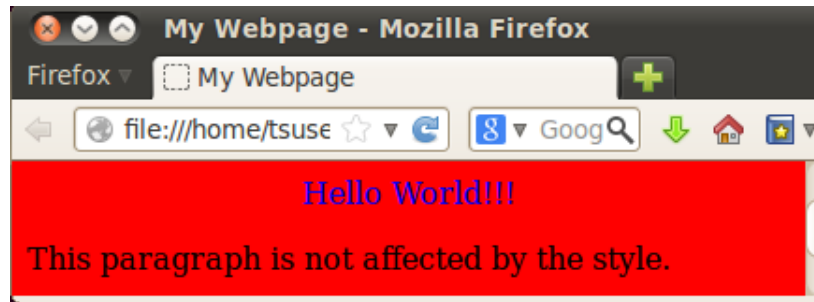
- The id selector is used to specify a style for a single, unique element.
- The id selector uses the id attribute of the HTML element, and is defined with a "#".
The style rule below will be applied to the element with id="first":

Example:

```
<html>
  <head><title>My Webpage</title>
    <style>
      body {background-color : #FF0000 ;}
      #first{color : blue ; text-align : center ;}
    </style>
  </head>
```

Cascading Style Sheets

```
<body>
  <p id = "first">Hello World!!!</p>
  <p>This paragraph is not affected by the style.</p>
</body>
</html>
```



class Selector

- The class selector is used to specify a style for a group of elements. Unlike the id selector, the class selector is most often used on several elements.
- This allows you to set a particular style for many HTML elements with the same class.
- The class selector uses the HTML class attribute, and is defined with a “.”

Example below shows, all `< p >` elements with `class=“changestyle”` will be center-aligned and text color is changed to green color.

If we have to apply style to all the HTML elements present in the document then we have to replace `“p.changestyle”` to `“.changestyle”` then the style will automatically applied to all the elements in the body.

Example:

Applying styles for the elements in the class.

```
<html>
  <head>
    <style>
      p.changestyle
```

Cascading Style Sheets

```

        {
            text-align:center;
            color:green;
        }
    </style>
</head>
<body>
    <h1 class="changestyle">This heading will not be
                                affected</h1>
    <p class="changestyle">This paragraph will be
                        center-aligned and changes its
                        text color to green.</p>
</body>
</html>

```



Internal style sheet

An internal style sheet should be used when a single document has a unique style. While using internal style sheet, You have to define internal styles in the head section(i.e., `< head >`) of an HTML page, by using the `< style >` tag.

Creating an internal CSS

CSS code is not written the same way as HTML code is. This makes sense because CSS is not HTML, but rather a way of manipulating existing HTML and finally put CSS inside style tags. The HTML code below contains an example of `< style >`'s usage.

Example: Designing an internal style sheet

Cascading Style Sheets

```
<html>
  <head>
    <title>Internal Style Sheet</title>
    <style type="text/css">
      p { color: blue; }
      .newStyle { font-size: 20px; }
      #change
      {
        font-size: 28px;
        text-align: center;
        color: green;
      }
    </style>
  </head>
  <body>
    <p>Woo hoo! I'm learning HTML!</p>
    <p class="newStyle">Woo hoo! I'm learning HTML!</p>
    <p class="newStyle" id="change">Woo hoo! I'm
      learning HTML!</p>
    <p class="newStyle">Woo hoo! I'm learning HTML!</p>
  </body>
</html>
```

Explanation

```
p { color: blue; }
.newStyle { font-size: 20px; }
#change { font-size: 28px; text-align: center; color: green; }
```

The first line of the internal style sheet has 3 parts.

selector p The selector points to a certain type of HTML tag, which here are the paragraph tags or `< p >....< /p >`.

property color

value blue

The second line of the internal style sheet also has these same three parts, but the HTML tag name out of the first part, the selector.

Cascading Style Sheets

```
.newStyle { font-size: 20px; }
```

Every html tag in the selector started with “ . ” to indicate that as a class name is going to be given, and from above they given class name as **newStyle**.

So, Every HTML tag with the class name of newStyle has this style,
{font-size: 20px;} will be applied.

The third line of the internal style sheet again has the same three parts, but this time the selector was started with a “#” to indicate that an id name is going to be given. From above they have given the id name as, #change.

```
#change{ font-size: 28px; text-align: center; color: green;}
```

Now, the HTML tag with the id name of change has this style,

font-size: 28px; text-align: center; color: green; , will be applied to it.

As from above code we got to know that we have added three properties and values to this line of the style sheet. We can add as many properties and values into your selector as you want. That doesn't mean they will all work. CSS can be TRICKY. Some CSS properties and values will only seem to work with certain HTML tags or even in certain internet browsers.

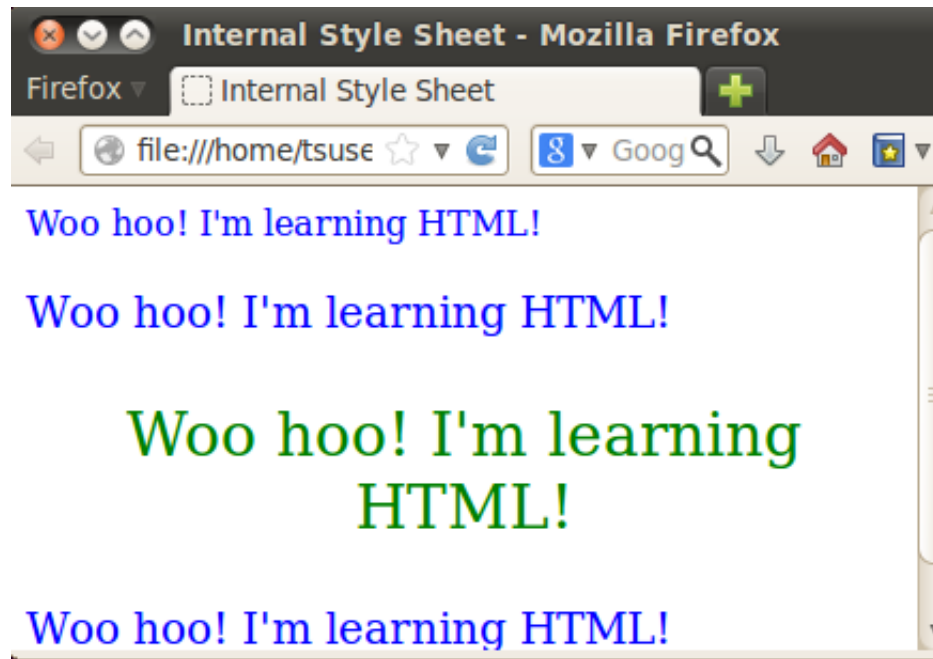
we can also be more specific in the naming of the selectors in the internal style sheet. If we have replaced those three lines with the below lines.

```
p{ color: blue;}
p.newStyle{font-size: 20px;}
p.newStyle#Change{font-size: 28px; text-align: center; color: green;}
```

Now, the second line will only apply its style to HTML paragraph tags with the class name “newStyle”, and the third line will only apply its style to the one HTML paragraph tag with the class name “

newStyle” and the id name “change”. Above code displays as follows.

Cascading Style Sheets



External Style Sheet

When using CSS it is preferable to keep the CSS separate from your HTML. Placing CSS in a separate file allows the web designer to completely differentiate between content (HTML) and design (CSS). External CSS is a file that contains only CSS code and is saved with a ".css" file extension. This CSS file is then referenced in your HTML using the `< link >` instead of `< style >`.

File Creation

For example, Design an External style sheet.

Let us get started by making that external CSS file. Open up notepad.exe, or any other plain text editor and type the following CSS code.

CSS Code

```
body { background-color: yellow; }
p { color: white; }
p.newStyle { font-size: 1.5px; text-align: center; }
p.newStyle#change { font-size: 2em; text-align: right;
                    color: blue; }
```

Now save the file as a CSS (.css) file. Make sure that you are not saving it as a text (.txt) file, as notepad likes to do by default. Name the file as "test.css" (without the quotes). Now create a new HTML file and fill it with the following code.

HTML Code

```
<html>
  <head>
    <title>External Style Sheet</title>
    <link rel="stylesheet" type="text/css"
          href="test.css" />
  </head>
  <body>
    <p>Woo hoo! I'm learning HTML!</p>
    <p class="newStyle">Woo hoo! I'm learning HTML!</p>
    <p class="newStyle" id="change">Woo hoo! I'm
                                   learning HTML!</p>
```

Cascading Style Sheets

```
<p class="newStyle">Woo hoo! I'm learning HTML!</p>
</body>
</html>
```

Then save this file as "index.html" (without the quotes) in the same directory as your CSS file. Now open your HTML file in your web browser and it should look something like this.

Uses:

- It keeps your website design and content separate.
- It's much easier to reuse your CSS code if you have it in a separate file. Instead of typing the same CSS code on every web page you have, simply have many pages refer to a single CSS file with the "link" tag.
- You can make drastic changes to your web pages with just a few changes in a single CSS file.

Styling Backgrounds

Styling Backgrounds		
Property	Example	Description
background-color	body { background-color:#b0c4de; }	Sets the background color of an element
background-image	body { background-image:url('image.gif'); }	Sets the background image for an element
background-repeat and background-position	body { background-image:url('image.png'); background-repeat:repeat-x; background-position:right top; } (repeat Values : repeat,no-repeat,repeat-x) (position Values : right, left, top, bottom, center)	Sets how a background image will be repeated and Sets the background position of the image
background-attachment	body { background-image:url('image.png'); background-repeat:repeat-x; background-position:right top; background-attachment:value; } (attachment Values : scroll(default),fixed)	Sets the background attachment of the image depends upon the value it changes

Cascading Style Sheets

Styling Text

Styling Text		
Property	Example	Description
color	<code>body {color:blue;} h1 {color:#00ff00;}</code>	Sets the color of text
direction	<code><style> p.ex1 {direction:rtl;} </style></code> (Values : rtl, ltr)	Specifies the text direction/writing direction(right to left ,left to right)
letter-spacing	<code><style> h1 {letter-spacing:2px;} </style></code> (Values : normal(default), number(postive or negative) length(px/-px/cm))	Increases or decreases the space between characters in a text
line-height	<code><style> p.small {line-height:70%;} p.big {line-height:200%;} </style></code> (Values : normal, number, length(px/pt/cm), percentage)	Sets the line height
text-align	<code><style> h1 {text-align:center} </style></code> (Values : center, left, right, justify)	Specifies the horizontal alignment of text

Cascading Style Sheets

text-decoration	<pre><style> h1 {text-decoration:underline;} </style></pre> <p>(Values : underline, line-through, underline, blink)</p>	Specifies the decoration added to text
text-indent	<pre><style> p {text-indent:50px;} </style></pre> <p>(Values : length(px/pt/cm/em), percentage)</p>	Specifies the indentation of the first line in a text-block
text-shadow	<pre>h1{ text-shadow: 20px 10px #ff0000;} (Syntax : text-shadow : h-shadow v-shadow blur color;)</pre>	Specifies the shadow effect added to text
text-transform	<pre><style> p.uppercase {text- transform:uppercase;} </style></pre> <p>(Values : none(default), capitalize, uppercase, lowercase)</p>	Controls the capitalization of text
vertical-align	<pre><style> img.top {vertical-align:text-top;} </style></pre> <p>(Values : length(px/cm/%),baseline(default) sub, super, top, bottom, middle, text-top, text-bottom)</p>	Sets the vertical alignment of an element
white-space	<pre>p{white-space:nowrap;} (Values : normal(default), nowrap, pre, pre-line, pre-wrap)</pre>	Specifies how white-space inside an element is handled
word-spacing	<pre>p{word-spacing:30px;} (Values : normal(default), length(px/pt/cm/em))</pre>	Increases or decreases the space between words in a text

Styling Fonts

Cascading Style Sheets

Styling Fonts		
Property	Example	Description
font-family	<code>p{font-family: "Times New Roman", Georgia, Serif;}</code> (family-name - "times", "courier", "arial" generic-family - "serif", "sans-serif", "cursive")	Specifies the font family for text
font-size	<code>h1 {font-size: 250%;}</code> (Values : xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, length(px/cm), %)	Specifies the font size of text
font-style	<code>p.normal {font-style: normal;}</code> (Values : normal, italic, oblique)	Specifies the font style for text
font-variant	<code>p.small{font-variant: small-caps;}</code> (Values : normal, small-caps)	Specifies whether or not a text should be displayed in a small-caps font
font-weight	<code>p.normal {font-weight: normal;}</code> (Values : normal, bold, bolder, lighter Multiples of 100 up to 900)	Specifies the weight of a font

Styling Links

Styling Links		
Anchors	Example	Description
a : link	<code>a:link {color: #FF0000;}</code>	Sets the color of a link
a : visited	<code>a:visited {color: #00FF00;}</code>	Changes the color of a link visited
a : hover	<code>a:hover {color: #FF00FF;}</code>	Changes the color of a link on mouse over
a : active	<code>a:active {color: #0000FF;}</code>	Sets the color of a link on click

Styling Lists

Styling Lists		
Property	Example	Description
list-style-image	<code>ul{list-style-image: url('image.jpg');}</code>	Specifies an image as the list-item marker
list-style-position	<code>ul{list-style-position: inside;}</code> (Values : inside, outside)	Specifies if the list-item markers should appear inside or outside the content flow
list-style-type	<code>ul.circle {list-style-type: circle}</code> <code>ul.square {list-style-type: square}</code> <code>ol.upper-roman {list-style-type: upper-roman}</code> <code>ol.lower-alpha {list-style-type: lower-alpha}</code> (Values : Disc, Circle, Square, Decimal, Lower-roman, Upper-roman, Lower-alpha, Upper-alpha, none)	Specifies the type of list-item marker

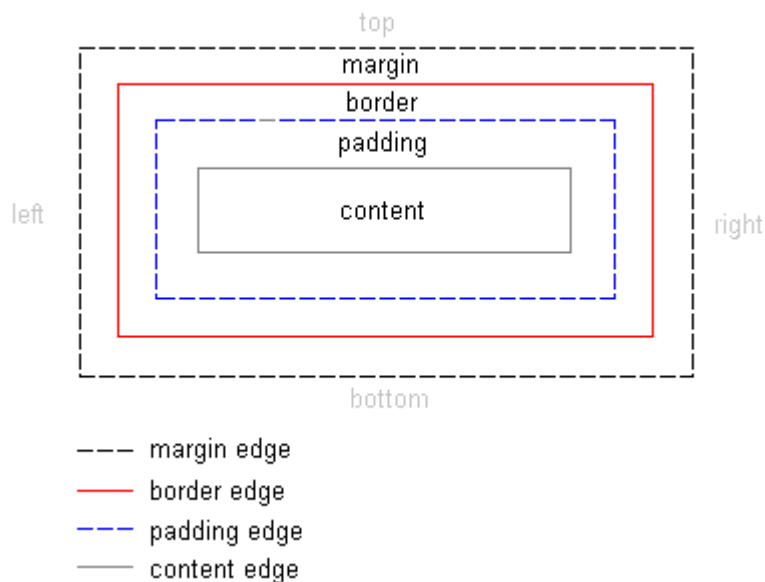
Cascading Style Sheets

Styling Tables

Styling Tables		
Property	Example	Description
border	table, th, td {border: 1px solid black;}	Used to display a double border for the table
border-collapse	table {border-collapse: collapse;} table, th, td {border: 1px solid black;}	Used to display a single border for the table
Width and Height	table{width:100%;} th{height:50px;}	Used to set Width and Height of the table
text-align	td{text-align:right;} td{vertical-align:top;} (Values : right, left, center. Vertical-align : top, bottom, middle.)	Used to align the text in a table
padding	td{padding:15px;}	Used to control the space between the border and content in a table
color	table, td, th {border:1px solid green;} th {background-color:green; color:white;}	Used to specify the color of the borders, and the text and background color of th elements

CSS Box Model

The box model in CSS describes the boxes which are being generated for HTML-elements. The CSS box model is essentially a box that wraps around HTML elements, and it consists of: *margins*, *borders*, *padding*, and the actual *content*. The diagram below shows how the box model is constructed:



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

CSS Border		
Property	Example	Description
border-style	<code>border-style:solid;</code> (<i>Values</i> : none, dotted, solid, double, groove, ridge, inset, outset)	It specifies what kind of border to display. Ex : (solid: Defines a solid border)
border-width	<code>p{ border-style:solid; border-width:5px; }</code> (<i>Values</i> : length(px/cm/em))	It is used to set the width of the border. Ex: (5px : It alters the width of the border.)
border-color	<code>p{ border-style:solid; border-color:red; }</code> (<i>Values</i> : name, RGB , Hex)	It is used to set the color of the border. Ex: (red : red color is applied to border)

- *Content* - The content of the box, where text and images appear