**Chapter 3**

**Cascading style sheets (CSS)**

|  |  |
| --- | --- |
|  |  |

### Introduction to CSS

**CSS** stands for **Cascading Style Sheet**, and it allows us to separate our web sites HTML content from its style. As always we use HTML file to arrange the content, but all of the presentation/formatting (fonts, colors, background, borders, text formatting, link effects & so on...) are accomplished within a CSS.

CSS is a web page layout method that has been added to HTML to give web developers more control over their design and content layout. Using **CSS** allows a designer to **create** a standard set of commands (either embedded inside the web page or from an external page) that controls the style of all subsequent pages.

CSS separates the layout and the styles of a web page. This is often difficult to comprehend for web designers that are used to compiling their creative and HTML coding in a single web page document. Styles such as fonts, font sizes, margins, can be specified in one place, and then the Web pages feed off this one master list, with the styles cascading throughout the page or an entire site.

It is more and more difficult to create Web sites where the content of HTML documents was clearly separated from the document's presentation layout. The layout of the document was supposed to be taken care of by the browser, without using any formatting tags. This is because styles sheets define **HOW HTML** elements are to be displayed, just like the font tag and the color attribute in HTML.

**CSS** is a breakthrough in web design because it allows developers to control the **style** and **layout** of multiple web pages all at once. As a web developer, it is possible to define a style for each HTML element and apply it to as many web pages as we want. To make a global change, simply change the style, and all elements in the Web are updated automatically.

Style sheets allow style information to be specified in many ways. Styles can be specified **inside** a single HTML element, inside the **<head>** element of an HTML page, or in an external CSS file. Even multiple external style sheets can be referenced inside a single HTML document.

What is CSS?

* **CSS** stands for **C**ascading **S**tyle **S**heets
* Styles define **how to display** HTML elements
* Styles were added to HTML 4.0 **to solve a problem**
* **External Style Sheets** can save a lot of work
* External Style Sheets are stored in **CSS files**

**CSS Saves a Lot of Work!**

CSS defines HOW HTML elements are to be displayed.

Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in a Web site, just by **editing** one single file!

**CSS Syntax**

A CSS rule has two main parts: a selector, and one or more declarations:



* The selector is normally the **HTML** element you want to **style**.
* Each declaration consists of a **property** and a **value**.
* The property is the style **attribute** you want to change. Each property has a **value**.
* CSS declarations always ends with a **semicolon**, and declaration groups are surrounded by **curly brackets**:

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

To make the CSS more readable, you can put one declaration on each line, like this:

|  |  |
| --- | --- |
| Example   |  | | --- | | p { color:red; text-align:center; } | |

**CSS Comments**

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

A CSS comment begins with "/\*", and ends with "\*/", like this:

|  |
| --- |
| /\*This is a comment\*/ p { text-align:center; /\*This is another comment\*/ color:black; font-family:arial;} |

**Grouping Content**

The **<div>** and **<span>** elements allow you to group together several elements to create sections or subsections of a page.

This is the very important block level tag which plays a big role in grouping various other HTML tags and applying CSS on group of elements. Even now <div> tag can be used to create webpage layout where we define different parts (Left, Right, Top etc.) of the page using <div> tag. This tag does not provide any visual change on the block but this has more meaning when it is used with CSS.

**For example,** you might want to put all of the footnotes on a page within a **<div>** element to indicate that all of the elements within that **<div>** element relate to the footnotes. You might then attach a style to this **<div>** element so that they appear using a special set of **style rules**.

**Example 1:**

<!DOCTYPE html>

<html>

<head>

<title> Div Tag Example</title>

</head>

<body>

<div id="menu" align="middle" >

<a href="/index.htm">HOME</a> |

<a href="/about/contact\_us.htm">CONTACT</a> |

<a href="/about/index.htm">ABOUT</a>

</div>

<div id="content" align="left" bgcolor="white">

<h5>Content Articles</h5>

<p>Actual content goes here.....</p>

</div>

</body>

</html>

**Example**

<!DOCTYPE html>

<html>

<head>

<title>HTML div Tag</title>

</head>

<body>

<!-- **First group of tags** -->

<div style="color:red">

<h4>This is first group</h4>

<p>Following is a list of vegetables</p>

<ul>

<li>Beetroot</li>

<li>Ginger</li>

<li>Potato</li>

<li>Radish</li>

</ul>

</div>

<!-- **Second group of tags** -->

<div style="color:green">

<h4>This is second group</h4>

<p>Following is a list of fruits</p>

<ul>

<li>Apple</li>

<li>Banana</li>

<li>Mango</li>

<li>Strawberry</li>

</ul>

</div>

</body>

</html>

The **<span>** element, on the other hand, can be used to group inline elements only. So, if you have a part of a sentence or paragraph which you want to group together, you could use the **<span>** element as follows.

The HTML **<span>** is an inline element and it can be used to group inline-elements in an HTML document. This tag also does not provide any visual change on the block but has more meaning when it is used with CSS.

**Example1**

<!DOCTYPE html>

<html>

<head>

<title>Span Tag Example</title>

</head>

<body>

<p>This is the example of <span style="color:green">span tag</span> and the <span style="color:red">div tag</span> alongwith CSS</p>

</body>

</html>

**Example 2**

<!DOCTYPE html>

<html>

<head>

<title>HTML span Tag</title>

</head>

<body>

<p>This is <span style="color:red">red</span> and this is <span style="color:green">green</span></p>

</body> </html>

* The difference between the **<span>** tag and the **<div>** tag is that the <span> tag is used with inline elements whereas the <div> tag is used with block-level elements.

**CSS Id and Class Selector**

|  |  |
| --- | --- |
|  |  |

In addition to setting a style for a HTML element, CSS allows you to specify your own **selectors** called "**id**" and "**class**".

**The 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="para1":

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

RemarkDo **NOT** start an ID name with a number! It will not work in Mozilla/Firefox.

**The 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 any HTML elements with the **same** class.
* The **class** selector uses the HTML class attribute, and is defined with a "**.**"

In the example below, all HTML elements with class="center" will be center-aligned:

|  |  |
| --- | --- |
| Example   |  | | --- | | .center {text-align:center;} | |

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

In the example below, all **p** elements with class="center" will be **center-aligned**:

|  |  |
| --- | --- |
| Example   |  | | --- | | p.center {text-align:center;} | |

RemarkDo **NOT** start a class name with a number! This is only supported in **Internet Explorer.**

# CSS How To

|  |  |
| --- | --- |
|  |  |

When a browser reads a style sheet, it will format the document according to it.

## Three Ways to Insert CSS

There are **three** ways of inserting a style sheet:

* External style sheet
* Internal style sheet
* Inline style
* **External Style Sheet**: Define style sheet rules in a separate .css file and then include that file in your HTML document using HTML **<link>** tag.
* **Internal Style Sheet**: Define style sheet rules in header section of the HTML document using **<style>** tag.
* **Inline Style Sheet**: Define style sheet rules directly along-with the HTML elements using **style** attribute.

## External Style Sheet

An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the **<link>** tag. The **<link>** tag goes **inside** the head section:

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

An external style sheet can be written in any text editor. The file should not contain any **html tags.** Your style sheet should be saved with **a .css** extension. An example of a **style sheet file** is shown below:

|  |
| --- |
| hr {color:sienna;} p {margin-left:20px;} body {background-image:url("images/back40.gif");} |
|  |

Save **sytle.css**

/\* CSS class \*/

.red{

color: Blue;

}

.thick{

font-size:20px;

}

.green{

color:green;

}

<!DOCTYPE html>

<html>

<head>

<title>HTML External CSS</title>

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

</head>

<body>

<p class="red">This is red</p>

<p class="thick">This is thick</p>

<p class="green">This is green</p>

<p class="thick green">This is thick and green</p>

</body>

</html>

* Do not leave spaces between the property value and the units! "margin-left:20 px" (instead of "margin-left:20px") will work in **IE**, but **not** in **Firefox or Opera**.

## Internal Style Sheet

An internal style sheet should be used when a **single** document has a **unique style**. You define **internal styles** in the head section of an HTML page, by using the **<style>** tag, like this:

|  |
| --- |
| <head> <style type="text/css"> hr {color:sienna;} p {margin-left:20px;} body {background-image:url("images/back40.gif");} </style> </head> |

**Example:**

<!DOCTYPE html>

<html>

<head>

<title>HTML Internal CSS</title>

<style type="text/css">

.red{

color: red;

}

.thick{

font-size:20px;

}

## .green{

color:green;

}

</style>

</head>

<body>

<p class="red">This is red</p>

<p class="thick">This is thick</p>

<p class="green">This is green</p>

<p class="thick green">This is thick and green</p>

</body>

## </html>

## Inline Styles

An **inline style** loses many of the advantages of style sheets by mixing content with presentation. Use this method sparingly!

To use inline styles you use the **style** attribute in the relevant tag. The style attribute can contain any **CSS** property. The example shows how to change the **color** and the **left margin** of a paragraph:

|  |
| --- |
| <p style="color:sienna;margin-left:20px">This is a paragraph.</p> |

**Example:**

<!DOCTYPE html>

<html>

<head>

<title>HTML Inline CSS</title>

</head>

<body>

<p style="color:red;">This is red</p>

<p style="font-size:20px;">This is thick</p>

<p style="color:green;">This is green</p>

<p style="color:green;font-size:20px;">This is thick and green</p>

</body>

</html>

## Multiple Style Sheets

If some properties have been set for the same selector in different style sheets, the values will be inherited from the more specific style sheet.

For example, an external style sheet has these **properties** for the h3 selector:

|  |
| --- |
| h3 { color:red; text-align:left; font-size:8pt; } |

And an internal style sheet has these properties for the **h3** selector:

|  |
| --- |
| h3 { text-align:right; font-size:20pt; } |

If the page with the **internal style sheet** also links to the **external style sheet** the properties for **h3** will be:

|  |
| --- |
| color:red; text-align:right; font-size:20pt; |

The **color** is inherited from the **external style sheet** and the text**-alignment** and the **font-size** is replaced by the **internal style sheet**.

## Multiple Styles Will Cascade into One

Styles can be specified:

* inside an HTML element
* inside the head section of an HTML page
* in an external CSS file

**Tip:** Even multiple external style sheets can be referenced inside a single HTML document.

### Cascading order

What style will be used when there is more than **one style** specified for an **HTML** element?

Generally speaking we can say that all the styles will **"cascade**" into a new "**virtual**" style sheet by the following rules, where number four has the **highest** priority:

1. Browser default
2. External style sheet
3. Internal style sheet (in the head section)
4. Inline style (inside an HTML element)

So, an inline style (inside an HTML element) has the highest priority, which means that it will override a style defined inside the **<head>** tag, or in an external style sheet, or in a browser (a default value).

Remark**Note:** If the link to the external style sheet is placed after the **internal style sheet** in HTML <head>, the **external style sheet** will **override** the internal style sheet!

CSS Background

|  |  |
| --- | --- |
|  |  |

CSS background properties are used to define the background effects of an element.

CSS properties used for background effects:

* background-color
* background-image
* background-repeat
* background-attachment
* background-position

Background Color

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

The background color of a page is defined in the **body** selector:

|  |  |
| --- | --- |
| Example   |  | | --- | | body {background-color:#b0c4de;} | |

The background color can be specified by:

* name - a color name, like "red"
* RGB - an RGB value, like "rgb(255,0,0)"
* Hex - a hex value, like "#ff0000"

In the example below, the h1, p, and div elements have different background colors:

|  |  |
| --- | --- |
| Example   |  | | --- | | h1 {background-color:#6495ed;} p {background-color:#e0ffff;} div {background-color:#b0c4de;} | |

**Background Image**

The background-image property specifies an image to use as the **background** of an element.

By default, the image is **repeated** so it covers the entire element.

The background image for a page can be set like this:

|  |  |
| --- | --- |
| Example   |  | | --- | | body {background-image:url('paper.gif');} | |

Below is an example of a bad combination of text and background image. The text is almost not readable:

|  |  |
| --- | --- |
| Example   |  | | --- | | body {background-image:url('bgdesert.jpg');} | |

Background Image - Repeat Horizontally or Vertically

By default, the **background-image** property repeats an image both **horizontally** and **vertically**.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

|  |  |
| --- | --- |
| Example   |  | | --- | | body {  background-image:url('gradient2.png');} | |

If the image is repeated only horizontally (repeat-x), the background will look better:

|  |  |
| --- | --- |
| Example   |  | | --- | | body { background-image:url('gradient2.png'); background-repeat:repeat-x; } | |

Background Image - Set position and no-repeat

RemarkWhen using a background image, use an image that does not disturb the text.

Showing the image only once is specified by the background-repeat property:

|  |  |
| --- | --- |
| Example   |  | | --- | | body { background-image:url('img\_tree.png'); background-repeat:no-repeat; } | |

In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the **text** too much.

The position of the image is specified by the **background-position** property:

|  |  |
| --- | --- |
| Example   |  | | --- | | body { background-image:url('img\_tree.png'); background-repeat:no-repeat; background-position:right top; } | |

All CSS Background Properties

The number in the "**CSS**" column indicates in which CSS version the property is defined (CSS1 or CSS2).

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Values** | **CSS** |
| [background](http://www.w3schools.com/css/pr_background.asp) | Sets all the background properties in one declaration | *background-color background-image background-repeat background-attachment background-position* inherit | 1 |
| [background-attachment](http://www.w3schools.com/css/pr_background-attachment.asp) | Sets whether a background image is fixed or scrolls with the rest of the page | scroll fixed inherit | 1 |
| [background-color](http://www.w3schools.com/css/pr_background-color.asp) | Sets the background color of an element | *color-rgb color-hex color-name* transparent inherit | 1 |
| [background-image](http://www.w3schools.com/css/pr_background-image.asp) | Sets the background image for an element | url(*URL*) none inherit | 1 |
| [background-position](http://www.w3schools.com/css/pr_background-position.asp) | Sets the starting position of a background image | left top left center left bottom right top right center right bottom center top center center center bottom *x% y%* *xpos ypos* inherit | 1 |
| [background-repeat](http://www.w3schools.com/css/pr_background-repeat.asp) | Sets if/how a background image will be repeated | repeat repeat-x repeat-y no-repeat inherit | 1 |

# CSS Text

|  |  |
| --- | --- |
|  |  |

# Text formatting

This text is styled with some of the text formatting properties. The heading uses the text-align, text-transform, and color properties. The paragraph is indented, aligned, and the space between characters is specified. The underline is removed from the ["Try it yourself"](http://www.w3schools.com/css/tryit.asp?filename=trycss_text) link.

## Text Color

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

* name - a color name, like "red"
* RGB - an RGB value, like "rgb(255,0,0)"
* Hex - a hex value, like "#ff0000"

The default color for a page is defined in the body selector.

|  |  |
| --- | --- |
| Example  |  | | --- | | body {color:blue;} h1 {color:#00ff00;} h2 {color:rgb(255,0,0);} | |

## Text Alignment

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

Text can be centred , or aligned to the left or right, or justified.

When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

|  |  |
| --- | --- |
| Example  |  | | --- | | h1 {text-align:center;} p.date {text-align:right;} p.main {text-align:justify;} | |

## Text Decoration

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

The text-decoration property is mostly used to remove underlines from links for design purposes:

|  |  |
| --- | --- |
| Example  |  | | --- | | a {text-decoration:none;} | |

It can also be used to decorate text:

|  |  |
| --- | --- |
| Example  |  | | --- | | h1 {text-decoration:overline;} h2 {text-decoration:line-through;} h3 {text-decoration:underline;} h4 {text-decoration:blink;} | |

RemarkIt is not recommended to underline text that is not a link, as this often confuses users.

## 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.

|  |  |
| --- | --- |
| Example  |  | | --- | | p.uppercase {text-transform:uppercase;} p.lowercase {text-transform:lowercase;} p.capitalize {text-transform:capitalize;} | |

## Text Indentation

The text-indentation property is used to specify the indentation of the first line of a text.

|  |  |
| --- | --- |
| Example  |  | | --- | | p {text-indent:50px;} | |

## All CSS Text Properties

The number in the "**CSS**" column indicates in which **CSS** version the property is defined (**CSS1** or **CSmS2**).

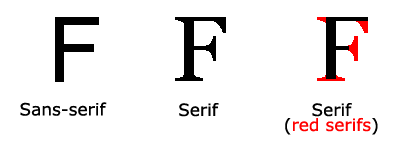
|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Values** | **CSS** |
| [color](http://www.w3schools.com/css/pr_text_color.asp) | Sets the color of a text | *color* | 1 |
| [direction](http://www.w3schools.com/css/pr_text_direction.asp) | Sets the text direction | ltr rtl | 2 |
| [line-height](http://www.w3schools.com/css/pr_dim_line-height.asp) | Sets the distance between lines | normal *number length %* | 1 |
| [letter-spacing](http://www.w3schools.com/css/pr_text_letter-spacing.asp) | Increase or decrease the space between characters | normal *length* | 1 |
| [text-align](http://www.w3schools.com/css/pr_text_text-align.asp) | Aligns the text in an element | left right center justify | 1 |
| [text-decoration](http://www.w3schools.com/css/pr_text_text-decoration.asp) | Adds decoration to text | none underline overline line-through blink | 1 |
| [text-indent](http://www.w3schools.com/css/pr_text_text-indent.asp) | Indents the first line of text in an element | *length %* | 1 |
| text-shadow |  | none *color* *length* |  |
| [text-transform](http://www.w3schools.com/css/pr_text_text-transform.asp) | Controls the letters in an element | none capitalize uppercase lowercase | 1 |
| unicode-bidi |  | normal embed bidi-override | 2 |
| [vertical-align](http://www.w3schools.com/css/pr_pos_vertical-align.asp) | Sets the vertical alignment of an element | baseline sub super top text-top middle bottom text-bottom *length* *%* | 1 |
| [white-space](http://www.w3schools.com/css/pr_text_white-space.asp) | Sets how white space inside an element is handled | normal pre nowrap | 1 |
| [word-spacing](http://www.w3schools.com/css/pr_text_word-spacing.asp) | Increase or decrease the space between words | normal *length* | 1 |

CSS Font

|  |  |
| --- | --- |
|  |  |

CSS font properties define the font family, boldness, size, and the style of a text.

Difference Between Serif and Sans-serif Fonts



RemarkOn computer screens, sans-serif fonts are considered easier to read than serif fonts.

CSS Font Families

In CSS, there are **two** types of font family names:

* **generic family** - a group of font families with a similar look (like "Serif" or "Monospace")
* **font family** - a specific font family (like "Times New Roman" or "Arial")

|  |  |  |
| --- | --- | --- |
| **Generic family** | **Font family** | **Description** |
| Serif | Times New Roman Georgia | Serif fonts have small lines at the ends on some characters |
| Sans-serif | Arial Verdana | "Sans" means without - these fonts do not have the lines at the ends of characters |
| Monospace | Courier New Lucida Console | All monospace characters have the same width |

**Font Family**

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

The font-family property should hold several font names as a "**fallback**" system. If the browser does not support the first font, it **tries** the next font.

Start with the font you want, and end with a generic family, to let the browser pick a similar font in the generic family, if no other fonts are available.

**Note**: If the name of a font family is more than one word, it must be in quotation marks, like font-family: "**Times New Roman**".

More than one font family is specified in a comma-separated list:

|  |  |
| --- | --- |
| Example   |  | | --- | | p{font-family:"Times New Roman", Times, serif;} | |

For more commonly used font combinations, look at our [Web Safe Font Combinations](http://www.w3schools.com/css/css_websafe_fonts.asp).

**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)

|  |  |
| --- | --- |
| Example   |  | | --- | | p.normal {font-style:normal;} p.italic {font-style:italic;} p.oblique {font-style:oblique;} | |

**Font Size**

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

Being able to manage the text size is important in web design. However, you should not use font size adjustments to make paragraphs look like headings, or headings look like paragraphs.

Always use the proper HTML tags, like <h1> - <h6> for headings and <p> for paragraphs.

The font-size value can be an absolute, or relative size.

Absolute size:

* Sets the text to a specified size
* Does not allow a user to change the text size in all browsers (bad for accessibility reasons)
* Absolute size is useful when the physical size of the output is known

Relative size:

* Sets the size relative to surrounding elements
* Allows a user to change the text size in browsers

RemarkIf you do not specify a font size, the default size for normal text, like paragraphs, is 16px (16px=1em).

**Set Font Size With Pixels**

Setting the text size with pixels, gives you full control over the text size:

|  |  |
| --- | --- |
| Example   |  | | --- | | h1 {font-size:40px;} h2 {font-size:30px;} p {font-size:14px;} | |

The example above allows Firefox, Chrome, and Safari to resize the text, **but not Internet Explorer**.

The text can be resized in all browsers using the zoom tool (however, this resizes the entire page, not just the text).

**Set Font Size With Em**

To avoid the resizing problem with Internet Explorer, many developers use em instead of pixels.

The em size unit is recommended by the W3C.

1em is equal to the current font size. The default text size in browsers is 16px. So, the default size of 1em is 16px.

The size can be calculated from pixels to em using this formula: *pixels*/16=*em*

|  |  |
| --- | --- |
| Example   |  | | --- | | h1 {font-size:2.5em;} /\* 40px/16=2.5em \*/ h2 {font-size:1.875em;} /\* 30px/16=1.875em \*/ p {font-size:0.875em;} /\* 14px/16=0.875em \*/ | |

In the example above, the text size in em is the same as the previous example in pixels. However, with the em size, it is possible to adjust the text size in all browsers.

Unfortunately, there is still a problem with IE. When resizing the text, it becomes larger than it should when made larger, and smaller than it should when made smaller.

Use a Combination of Percent and Em

The solution that works in all browsers, is to set a default font-size in percent for the body element:

|  |  |
| --- | --- |
| Example   |  | | --- | | body {font-size:100%;} h1 {font-size:2.5em;} h2 {font-size:1.875em;} p {font-size:0.875em;} | |

Our code now works great! It shows the same text size in all browsers, and allows all browsers to zoom or resize the text!

All CSS Font Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Values** | **CSS** |
| [font](http://www.w3schools.com/css/pr_font_font.asp) | Sets all the font properties in one declaration | *font-style font-variant font-weight font-size/line-height font-family* caption icon menu message-box small-caption status-bar inherit | 1 |
| [font-family](http://www.w3schools.com/css/pr_font_font-family.asp) | Specifies the font family for text | *family-name generic-family* inherit | 1 |
| [font-size](http://www.w3schools.com/css/pr_font_font-size.asp) | Specifies the font size of text | xx-small x-small small medium large x-large xx-large smaller larger *length %* inherit | 1 |
| [font-style](http://www.w3schools.com/css/pr_font_font-style.asp) | Specifies the font style for text | normal italic oblique inherit | 1 |
| [font-variant](http://www.w3schools.com/css/pr_font_font-variant.asp) | Specifies whether or not a text should be displayed in a small-caps font | normal small-caps inherit | 1 |
| [font-weight](http://www.w3schools.com/css/pr_font_weight.asp) | Specifies the weight of a font | normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit |  |

CSS Links

|  |  |
| --- | --- |
|  |  |

Links can be styled in different ways.

Styling Links

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

Special for links are that they 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   |  | | --- | | a:link {color:#FF0000;}      /\* unvisited link \*/ a:visited {color:#00FF00;}  /\* visited link \*/ a:hover {color:#FF00FF;}  /\* mouse over link \*/ a:active {color:#0000FF;}  /\* selected link \*/ | |

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

Common Link Styles

In the example above the link changes color depending on what state it is in.

Lets go through some of the other common ways to style links:

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 specifies the background color for links:

|  |  |
| --- | --- |
| Example   |  | | --- | | a:link {background-color:#B2FF99;} a:visited {background-color:#FFFF85;} a:hover {background-color:#FF704D;} a:active {background-color:#FF704D;} | |

CSS Lists

|  |  |
| --- | --- |
|  |  |

The CSS list properties allow you to:

* Set different list item markers for **ordered** lists
* Set different list item markers for **unordered** lists
* Set an image as the list item marker

List

In HTML, there are two types of lists:

* unordered lists - the list items are marked with bullets
* ordered lists - the list items are marked with numbers or letters

With CSS, lists can be styled further, and images can be used as the list item marker.

Different List Item Markers

The type of list item marker is specified with the list-style-type property:

|  |  |
| --- | --- |
| Example   |  | | --- | | 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;} | |

Some of the property values are for unordered lists, and some for ordered lists.

Values for Unordered Lists

|  |  |
| --- | --- |
| **Value** | **Description** |
| None | No marker |
| Disc | Default. The marker is a filled circle |
| Circle | The marker is a circle |
| Square | The marker is a square |

Values for Ordered Lists

|  |  |
| --- | --- |
| **Value** | **Description** |
| Armenian | The marker is traditional Armenian numbering |
| Decimal | The marker is a number |
| decimal-leading-zero | The marker is a number padded by initial zeros (01, 02, 03, etc.) |
| Georgian | The marker is traditional Georgian numbering (an, ban, gan, etc.) |
| lower-alpha | The marker is lower-alpha (a, b, c, d, e, etc.) |
| lower-greek | The marker is lower-greek (alpha, beta, gamma, etc.) |
| lower-latin | The marker is lower-latin (a, b, c, d, e, etc.) |
| lower-roman | The marker is lower-roman (i, ii, iii, iv, v, etc.) |
| upper-alpha | The marker is upper-alpha (A, B, C, D, E, etc.) |
| upper-latin | The marker is upper-latin (A, B, C, D, E, etc.) |
| upper-roman | The marker is upper-roman (I, II, III, IV, V, etc.) |

* **Note:** No versions of Internet Explorer (including IE8) support the property values "decimal-leading-zero", "lower-greek", "lower-latin", "upper-latin", "armenian", or "georgian" UNLESS a DOCTYPE is specified!

**An Image as The List Item Marker**

To specify an image as the list item marker, use the list-style-image property:

|  |  |
| --- | --- |
| Example   |  | | --- | | ul { list-style-image: url('sqpurple.gif'); } | |

The example above does not display equally in all browsers. IE and Opera will display the image-marker a little bit higher than Firefox, Chrome, and Safari.

If you want the image-marker to be placed equally in all browsers, a crossbrowser solution is explained below.

**Crossbrowser Solution**

The following example displays the image-marker equally in all browsers:

|  |  |
| --- | --- |
| Example   |  | | --- | | ul { list-style-type: none; padding: 0px; margin: 0px; } li { background-image: url(sqpurple.gif); background-repeat: no-repeat; background-position: 0px 5px;  padding-left: 14px;  } | |

Example explained:

* For ul:
  + Set the list-style-type to none to remove the list item marker
  + Set both padding and margin to 0px (for cross-browser compatibility)
* For li:
  + Set the URL of the image, and show it only once (no-repeat)
  + Position the image where you want it (left 0px and down 5px)
  + Position the text in the list with padding-left

All CSS List Properties

The number in the "CSS" column indicates in which CSS version the property is defined (CSS1 or CSS2).

|  |  |  |  |
| --- | --- | --- | --- |
| **Property** | **Description** | **Values** | **CSS** |
| [list-style](http://www.w3schools.com/css/pr_list-style.asp) | Sets all the properties for a list in one declaration | *list-style-type list-style-position list-style-image* inherit | 1 |
| [list-style-image](http://www.w3schools.com/css/pr_list-style-image.asp) | Specifies an image as the list-item marker | URL none inherit | 1 |
| [list-style-position](http://www.w3schools.com/css/pr_list-style-position.asp) | Specifies if the list-item markers should appear inside or outside the content flow | inside outside inherit | 1 |
| [list-style-type](http://www.w3schools.com/css/pr_list-style-type.asp) | Specifies the type of list-item marker | none disc circle square decimal decimal-leading-zero armenian georgian lower-alpha upper-alpha lower-greek lower-latin upper-latin lower-roman upper-roman inherit |  |

CSS Tables

|  |  |
| --- | --- |
|  |  |

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

Table Borders

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

The example below specifies a black border for **table**, **th**, and **td** elements:

|  |  |
| --- | --- |
| Example   |  | | --- | | table, th, td { border: 1px solid black; } | |

Notice that the table in the example above has double borders. This is because both the table, th, and td elements have separate borders.

To display a single border for the table, use the border-collapse property.

**Collapse Borders**

The border-collapse property sets whether the table borders are collapsed into a single border or separated:

|  |  |
| --- | --- |
| Example   |  | | --- | | table { border-collapse:collapse; } table,th, td { border: 1px solid black; } | |

**Table Width and Height**

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

The example below sets the width of the table to 100%, and the height of the th elements to **50px:**

|  |  |
| --- | --- |
| Example   |  | | --- | | table  { width:100%; } th { height:50px; } | |

**Table Text Alignment**

The text in a table is aligned with the text-align and vertical-align properties.

The text-align property sets the horizontal alignment, **like left,** **right**, or **center:**

|  |  |
| --- | --- |
| Example   |  | | --- | | td { text-align:right; } | |

The vertical-align property sets the vertical alignment, like top, bottom, or middle:

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

Table Padding

To control the space between the border and content in a table, use the padding property on td and th elements:

|  |  |
| --- | --- |
| Example   |  | | --- | | td { padding:15px; } | |

Table Color

The example below specifies the color of the borders, and the text and background color of th elements:

|  |  |
| --- | --- |
| Example   |  | | --- | | table, td, th { border:1px solid green; } th { background-color:green; color:white; } | |

**HTML – LAYOUTS**

A webpage layout is very important to give better look to your website. It takes considerable time to design a website's layout with great look and feel.

Now- a-days, all modern websites are using CSS and JavaScript based framework to come up with responsive and dynamic websites but you can create a good layout using simple HTML tables or division tags in combination with other formatting tags. This chapter will give you few examples on how to create a simple but working layout for your webpage using pure HTML and its attributes.

**HTML Layout - Using Tables**

The simplest and most popular way of creating layouts is using HTML <table> tag. These tables are arranged in columns and rows, so you can utilize these rows and columns in whatever way you like.

**Example**

For example, the following HTML layout example is achieved using a table with 3 rows and 2 columns but the header and footer column spans both columns using the colspan attribute:

<!DOCTYPE html>

<html>

<head>

<title>HTML Layout using Tables</title>

</head>

<body>

<table width="100%" border="0">

<tr>

<td colspan="2" bgcolor="#b5dcb3">

<h1>This is Web Page Main title</h1>

</td>

</tr>

<tr valign="top">

<td bgcolor="#aaa" width="50">

<b>Main Menu</b><br />

HTML<br />

PHP<br />

PERL...

</td>

<td bgcolor="#eee" width="100" height="200">

Technical and Managerial Tutorials

</td>

</tr>

<tr>

<td colspan="2" bgcolor="#b5dcb3">

<center>

Copyright © 2007 Tutorialspoint.com

</center>

</td>

</tr>

</table>

</body>

</html>

This will produce the following result:



**Multiple Columns Layout - Using Tables**

You can design your webpage to put your web content in multiple pages. You can keep your content in middle column and you can use left column to use menu and right column can be used to put advertisement or some other stuff. This layout will be very similar to what we have at our website tutorialspoint.com.

<!DOCTYPE html>

<html>

<head>

<title>Three Column HTML Layout</title>

</head>

<body>

<table width="100%" border="0">

<tr valign="top">

<td bgcolor="#aaa" width="20%">

<b>Main Menu</b><br />

HTML<br />

PHP<br />

PERL...

</td>

<td bgcolor="#b5dcb3" height="200" width="60%">

Technical and Managerial Tutorials

</td>

<td bgcolor="#aaa" width="20%">

<b>Right Menu</b><br />

HTML<br />

PHP<br />

PERL...

</td>

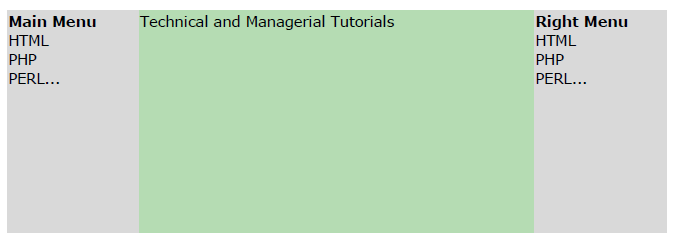
</tr>

<table>

</body>

</html>

This will produce the following result:



**HTML Layouts - Using DIV, SPAN**

The <div> element is a block level element used for grouping HTML elements. While the <div> tag is a block-level element, the HTML <span> element is used for grouping elements at an inline level.

Although we can achieve pretty nice layouts with HTML tables, but tables weren't really designed as a layout tool. Tables are more suited to presenting tabular data.

**Example**

<!DOCTYPE html>

<html>

<head>

<title>HTML Layouts using DIV, SPAN</title>

</head>

<body>

<div style="width:100%">

<div style="background-color:#b5dcb3; width:100%">

<h1>This is Web Page Main title</h1>

</div>

<div style="background-color:#aaa; height:200px;width:100px;float:left;">

<div><b>Main Menu</b></div>

HTML<br />

PHP<br />

PERL...

</div>

<div style="background-color:#eee; height:200px;width:350px;float:left;">

<p>Technical and Managerial Tutorials</p>

</div>

<div style="background-color:#aaa; height:200px;width:100px;float:right;">

<div><b>Right Menu</b></div>

HTML<br />

PHP<br />

PERL...

</div>

<div style="background-color:#b5dcb3;clear:both">

<center>

Copyright © 2007 Tutorialspoint.com

</center>

</div>

</div>

</body>

</html>