



CSS introduction by [Marc Oscar Schwager](#)

Background of this document + what it is

I put this information together in order to have a supplement to the school program in computer science for my son.

This course contains a guideline for teachers.

The CSS introduction gives you an idea how to use CSS to design the look (style) of your web pages (HTML documents).

Prerequisites

- You have to know the basic concepts mentioned in the [Computer Crash Course](#).
- You have to understand the course [HTML introduction](#).
- The [Build a Website easily](#) document explains you which steps you can take to create a Website. This is the umbrella document for this course.

Computer Crash Course (CCC)

The Computer Crash Course contains a guideline for teachers in basic computer knowledge. It can also be used as an introduction for beginners. You have to follow the Internet links and read additional information concerning the keywords mentioned in the CCC document, otherwise you won't understand this course.

HTML introduction

The HTML introduction gives you an idea how to create HTML pages on a base of examples and a minimum of theoretical background.

Build a Website easily

The "Build a Website easily" document explains you which steps you can take to create a Website. This is an umbrella document for the following courses : HTML introduction, CSS introduction and SEO introduction.

CSS introduction by [Marc Oscar Schwager](#)

Further reading

SEO introduction

The [SEO introduction](#) document describes how to write web pages search engine friendly with the goal that your web pages will be indexed and have a high ranking in search engines.

Table of Contents

Abstract.....	6
Applying CSS to a HTML document.....	7
In-line.....	7
Internal.....	7
External.....	8
Cascading.....	9
@import.....	10
Color.....	10
Text style possibilities.....	11
Text-indent.....	11
Text-align.....	11
Text-transform.....	11
Text-decoration.....	11
Line-height.....	11
Letter-spacing.....	12
Word-spacing.....	12
Font.....	13
font-family.....	13
font-size.....	13
font-style.....	13
font-variant.....	14
font-weight	14
Background.....	15
CSS background code follows this format.....	15
Text color and background color.....	15
Background image.....	15
Links.....	16
Link states.....	16

Lists.....	17
List properties.....	17
List-style-position	17
List-style-type.....	17
List-style-image.....	17
Selectors.....	18
The Class selector.....	18
The ID selector.....	19
Multiple selectors.....	20
Pseudo elements.....	20
Box model.....	21
Introduction.....	21
Margin.....	22
Border.....	23
Padding.....	24
Outline.....	24
Table	25
Positioning content.....	26
The float property.....	26
The position property	27
Split a web page into columns.....	28
Table columns.....	28
CSS columns.....	30
Group a HTML document.....	31
<div>.....	31
Semantic HTML.....	32
Miscellaneous.....	33
@import and multiple CSS files.....	33
Validate the HTML and CSS code.....	33
Check for browser compatibility.....	33

Abstract

CSS means Cascading Style Sheets.

CSS is a language to mainly style (format) HTML documents. CSS defines how HTML elements are to be displayed. CSS defines for example the style of a font or background and the position of texts, tables or images.

CSS can be defined and modified without affecting the HTML code, because the CSS code can reside in a separate file.

You can use a simple editor to create a CSS file. This file is a text file with the .css ending.

CSS syntax (code)

Selector **CSS declaration**

h1 **{color: red;}**
 where color is the **Property**
 where red is the **Value**

The **Selector** can be a HTML element, a class selector or ID selector. The Selector designs the element you want to style/format.

The **CSS declaration** contains a **Property** and a **Value**.

The **Property** is the style attribute you want to change.

The **Value** is the type (or variation) of the Property.

There can be a multiple of CSS declarations, for example `h1 {color: red; font-size:100%;}`

A **CSS comment** begins with `/*`, and ends with `*/`, like this: `/*this is a comment*/`

Applying CSS to a HTML document

In-line

This method defines the style for one HTML element.
This method loses many of the advantages of style sheets by mixing structure (HTML) with presentation. Use this method very restrictive!

HTML code / CSS code

```
<html>
  <head>
    <title>example</title>
  </head>
  <body>
    <p style="color: blue">only this text will be blue</p>
    <p> this text will be NOT blue</p>
  </body>
</html>
```

Internal

This method defines the style(s) of one HTML element(s) in the `<head>` section.

HTML code / CSS code

```
<html>
  <head>
    <title>example</title>
    <style type="text/css">p { color: blue;}</style>
  </head>
  <body>
    <p>all paragraphs in this HTML document will be blue</p>
  </body>
</html>
```

External

You can use one CSS file for your whole Website, in order to have one file where all the style declarations are located. This external style sheet allows you to change the style of all the pages in a Website, just by editing one single line in the CSS file. **This is the basic/recommended method using CSS!**

Each HTML document must have a `link` in the `<head>` area to connect it to the CSS file.

test.html

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="master.css" title="default"/>
  <title>example</title>
</head>
<body>
  <p>the style sheet definition makes this text red</p>
</bod>
</html>
```

master.css

```
p {color: red;}
.something {border:none;}
```


Cascading

CSS is called cascading because there is an order of priorities for styles.

Priorities from height to low :

- in-line style
- style defined inside the `<head>` tag
- external style sheet
- browser (= default value)

The external style sheet overrides the default/browser values

The style defined inside the `<head>` tag overrides the external style sheet values

The in-line style overrides the the style defined inside the `<head>` tag

Example :

You can override a `css file entry` (valuable for one Web page or more) in placing the following statement in a HTML file. `<style type="text/css">p {color:green}</style>`

master.css

```
p {color: red;}  
.something {border:none;}
```

test.html

```
<html>  
<head>  
  <link rel="stylesheet" type="text/css" href="master.css" title="default"/>  
  <style type="text/css">p {color:green}</style>  
  <title>example</title>  
</head>  
<body>  
  <p>all paragraphs in this HTML file will be green - not red!</p>  
</bod>  
</html>
```

@import

If your master CSS file defines that all paragraphs have to be green for the whole Website, but you want to have an exception for tree specific HTML documents, then you can define an exception with the **@import** rule. Usually you use this technique in order to change multiple styles (not only one!).

master.css

```
p {color: green;}
```

col_red.css

```
p {color: red;}
```

test.html

```
<html>
<head>
  <link rel="stylesheet" type="text/css" href="master.css" title="default"/>
  <style type="text/css"> @import url(col_red.css); </style>
  <title>example</title>
</head>
<body>
  <p>this text is red now</p>
</bod>
</html>
```

Color

The simplest way to use colors is to use the 16 common VGA colors like : aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.

For special colors outside the 16 ones use the **#xxxxxx** hexadecimal form.

Example of a hex equivalent : `p {color: #FF0000;}` is the same as `p {color: red;}`

Text style possibilities

Text-indent

Text-indent is the indentation at the beginning of an element, such as a `<p>` for example.

CSS code : `p {text-indent: 1em;}`

Text-align

The **value** of text-align can be : center, right, left, justify

CSS code : `p {text-align: value}`

Text-transform

It can be used to turn everything into **uppercase** or **lowercase** letters, or **capitalize** the first letter of each word.

CSS code : `p {text-transform: uppercase;}`

Text-decoration

This is one of the most widely used properties.

The values are **none** (regular text), **underline**, **overline**, **line-through** and **blink**.

CSS code : `p {text-decoration: underline}`

Line-height

CSS code : `p {line-height:220%}`

CSS introduction by [Marc Oscar Schwager](#)

Letter-spacing

CSS code : `p { letter-spacing:2px }`

Word-spacing

CSS code : `p { word-spacing:30px;}`

Font

font-family

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 one.

1) Traditional fallback solutions

- can be a group of font families with a similar look (like "Serif" or "Monospace")
- can be a specific font family (like "Times New Roman" or "Arial")
 - Example : `body {font-family: 'Times New Roman', Times, Serif;}`

2) Use [Google fonts](#) in order to have a fallback

font-size

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

Absolute size:

- Sets the font/text to a specified size
- Does not allow a user to change the font/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 font/text size in browsers

If you do not specify a font size, the default size for normal text is 16px (16px=1em=100%).

CSS code : `p {font-size: 120%;}`

font-style

The font-style property can use one of three values: normal, italic, oblique.

CSS code : `p {font-style: oblique;}`

font-variant

The font-variant property can be normal or small-caps.

CSS code : `p {font-variant: small-caps;}`

font-weight

The font-weight can be used with four build-in values: normal, bold, bolder, lighter.

The value can be in the range from 100 to 900, in increments of 100.

For a baseline, normal is defined as 400, and bold is defined as 700.

CSS code : `p {font-weight: bold;}`

Background

With CSS, you are able to set the background color or an image of any HTML element!

CSS background code follows this format

background-color || background-image || background-repeat || background-attachment || background-position

Keep in mind that the **background-size** property is part of CSS3.

Text color and background color

CSS code : `body {color: white; background-color: black;}`

HTML code equivalent : `<body text="white" bgcolor="black">`

Background image

CSS code : `body {background-image: url(graphic.jpg);
background-repeat:no-repeat;
background-attachment:fixed; background-position: 60% 55%;}`

HTML code equivalent : `<body background="graphic.jpg">`

Links

You can style links with any CSS property.

Link states

The following order rule is necessary if you use all link states .

a:link	indicates an unvisited link
a:visited	indicates if you visited the link
a:hover	indicates if the mouse is over the link
a:active	indicates if you are clicking on the link

CSS code :

```
a:link {background-color:yellow;}  
a:visited {background-color:fuchsia;}  
a:hover {background-color:green;}  
a:active {background-color:red;}
```


Lists

CSS allows you to set different list item markers for ordered and unordered lists.

List properties

- list-style sets all the properties for a list in one declaration
- list-style-position specifies if the list-item markers should appear inside or outside the content flow
- list-style-type specifies the type of list-item marker
- list-style-image specifies an image as the list-item marker

List-style-position

CSS code : `ul {list-style-position:inside;}`

List-style-type

CSS code : `ol {list-style-type: decimal;}`
`ul {list-style-type: circle;}`

List-style-image

CSS code : `ul {list-style-image: url('test.gif');}`

Check if your image marker style works on most browsers !

Selectors

Class and ID selectors can be used to apply a special style to a particular HTML element or a group of HTML elements.

The Class selector

The class selector can be used to identify one or **multiple elements**. CSS class selectors are normally used for many instances or tags.

The class selector can also be used to select multiple items in JavaScript.

Element with Class selector

CSS code : `p.test {color: green;}`

HTML code : `<p class="test">this text is green</p>`
`<p>this text is default</p>`

Element-less Class selector

This can be used for every HTML element.

CSS code : `.test {color: red;}`

HTML code : `<h1 class="test">this title must be red</h1>`
`<p class="test">this text must also be red</p>`
`<p>this text is default</p>`

The ID selector

The ID selector can be used to identify **only one element** on a web page. CSS ID selectors are used for a single instance or tag.

The ID selector can be used to select one item in JavaScript.

Element with ID selector

CSS code :
`p#parrd {color: red;}`
`p#paror {color: orange;}`
`p#parbl {color: blue;}`

HTML code :
`<p id="parrd">this paragraph is red</p>`
`<p id="paror">this paragraph is orange</p>`
`<p id="parbl">this paragraph is blue</p>`

Element-less ID selector

This can be used for every HTML element.

CSS code :
`#abcblue {color: blue;}`

HTML code :
`<h1 id="abcblue">the chosen HTML element is blue</h1>`

Multiple selectors

CSS code : `h3 em {color: blue;}`

HTML code : `<h3>This is a normal title this text is blue</h3>`

CSS code : `h3, em {color: blue;}`

HTML code : `<h3>This title is now bluethis text is blue</h3>`

or

`<h3>This title is now blue</h3>this text is blue`

CSS code : `h1, h2, h3 {color: green;}`

HTML code : `<h1>this is a green h1 title</h1>`

`<h2>this is a green h2 title</h2>`

`<h3>this is a green h3 title</h3>`

CSS code : `blockquote p strong, h1 .important {color: blue;}`

HTML code : `<blockquote>normal style<p>blue</p></blockquote>`

`<h1>normal "h1" <strong class="important">blue "h1" </h1>`

Pseudo elements

CSS code : `p:first-letter {color: red;}`
`p:first-line {color: green;}`

HTML code : `<p>this is the first line
this is the second line</p>`

CSS code : `h3:before {content: "***";}`
`h3:after {content:url(test.gif);}`

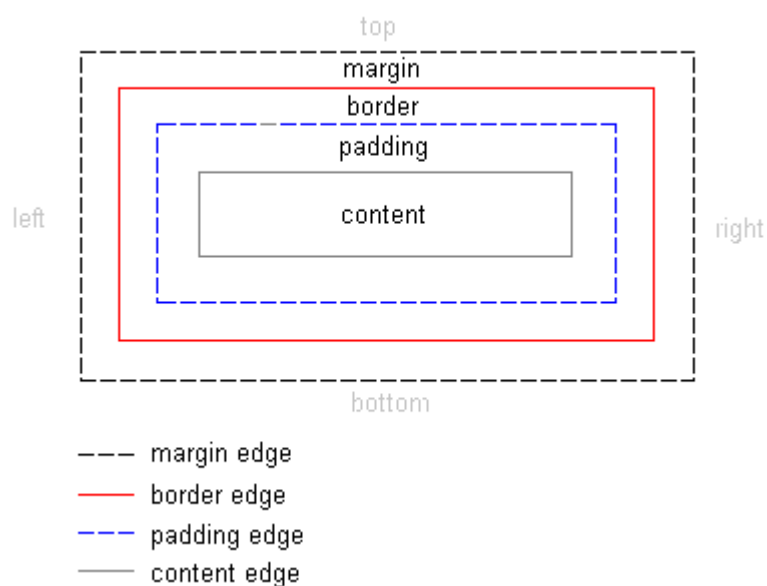
HTML code : `<h3>look, what is before and after this text</h3>`

Box model

Introduction

All HTML elements can be considered as boxes. The CSS box model is essentially a box that wraps around HTML elements.

The box model allows us to place a border around elements and space elements in relation to other elements.



- Margin : The transparent and empty space in between different boxes (space outside the box)
- Border : The "decorative" border around the box
- Padding : The space in between the content and the border of the box (space inside the box)
- Content : The content space of the box, where text and images appear

Content size

If you specify the width and height properties of an element with CSS, you are just setting the width and height of the content area! The full size of the element has to be calculated, you must add the padding, border and margin.

Margin

Margin properties define the space around elements.

CSS code :

```
p
{
background-color:yellow;
}
p.test
{
margin:100px 50px;
}
```

HTML code :

```
<p>paragraph with no specified margins.</p>
<p class="test">paragraph with specified margins.</p>
```

Border

The CSS border properties allow you to specify the style and color of an HTML element's border. None of the border properties will have any effect unless the border-style property is set! When using the border property, the order of the values are : border-width, border-style, border-color.

CSS code :

```
p.one
{
border-width:5px;
border-style:solid;
}
p.two
{
border-width:medium;
border-style:dashed;
}
p.three
{
border:1px dotted red; /* this is the short form for border */
}
```

HTML code :

```
<p class="one">first box.</p>
<p class="two">second box.</p>
<p class="three">third box</p>
```

Padding

Padding properties define the space between the element border and the element content.

CSS code :

```
p
{
background-color:yellow;
}
p.test
{
padding:25px 50px;
}
```

HTML code :

```
<p>This is a paragraph with no specified padding.</p>
<p class="test">This is a paragraph with specified paddings.</p>
```

Outline

Outline is a line that is drawn around HTML elements, outside the border edge.

CSS code :

```
p
{
outline:red dotted thick;
}
```

HTML code :

```
<p>look on this fantastic red doted outline !.</p>
```


Table

CSS code :

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

HTML code :

```
<table>
<tr>
<th>Name</th>
<th>Hobby</th>
<th>Country</th>
</tr>
<tr>
<td>Shaw</td>
<td>Sport</td>
<td>Australia</td>
</tr>
<tr>
<td>Harter</td>
<td>Sailing</td>
<td>Madagascar</td>
</tr>
<tr>
<td>Smith</td>
<td>Literature</td>
<td>New Zealand</td>
</tr>
</table>
```

Positioning content

The float property

The float property comes from the print layout - the text wraps around an image. In web design this property is also used for page layout (see the example below "CSS columns").

Example text floating around image

CSS code :

```
div
{
height:3em;
width:3em;
background-color:yellow;
float:left;
}
```

HTML code :

```
<div>img</div>
<p>This text floats around the box This text floats around the box This text floats around the box
This text floats around the box This text floats around the box This text floats around the box This
text floats around the box This text floats around the box This text floats around the box This text
floats around the box This text floats around the box This text floats around the box This text floats
around the box This text floats around the box This text floats around the box
</p>
```

The position property

The position property can be (mostly) used as replacement for floats. With "position" you don't have to care about "clear".

Example two columns, without "float"

CSS code :

```
#first {
  position:absolute;
  top:0;
  right:0;
  width:50%;
  height:10em;
  background-color:yellow;
}
#second {
  position:absolute;
  top:0;
  left:0;
  width:50%;
  height:10em;
  background-color:silver;
}
```

HTML code :

```
<div id="first">right text ... right text ...right text ...right text ...right text ...right text ...</div>
<div id="second">left text ... left text ...left text ...left text ...left text ...left text ...left text ...</div>
```

Split a web page into columns

Table columns

Don't use the HTML `<table>` element to layout your web pages. For example don't split a web page into columns. Use CSS instead to do that ! (see next title below)

Example two columns, with "table"

HTML code :

```
<table><tr>
```

```
<td id="menu">
```

```
<h2>menu block</h2>
```

```
<a href="#">menu link</a><br>
```

```
<a href="#">menu link</a><br>
```

```
<a href="#">menu link</a><br>
```

```
<a href="#">menu link</a>
```

```
</td>
```

```
<td id="content">
```

```
<h2>content block</h2>
```

```
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam cursus. Morbi ut mi. Nullam enim  
leo, egestas id, condimentum at, laoreet mattis, massa. Sed eleifend nonummy diam.
```

```
</td>
```

```
</tr></table>
```

CSS introduction by [Marc Oscar Schwager](#)

CSS code :

```
table  
{width: 100%;  
height: 100%;  
border: 0;  
}
```

```
#menu  
{width: 20%;  
vertical-align: text-top;  
background-color: yellow;  
}
```

```
#content  
{width: 80%;  
vertical-align: text-top;  
background-color: green;  
}
```

CSS columns

Example two columns with float

CSS code :

```
#leftBox
{
  background:silver;
  color:black;
  float:left;
  width:30%;
  height:6em;
  padding:2px;
}
```

```
#rightBox
{
  background:yellow;
  color:black;
  float:left;
  width:69%;
  height:6em;
  padding:2px;
}
```

HTML code :

```
<div id="leftBox">
<p>Here is my left box text... Here is my left box text... Here is my left box text... Here is my left
box text... Here is my left box text... </p>
</div>
```

```
<div id="rightBox">
<p>Here is the right box text ... Here is the right box text ... Here is the right box text ... Here is the
right box text ...</p>
</div>
```

Group a HTML document

<div>

The HTML element `<div>` can be used to group a document. This will be the exception in the future for the main blocks like header, footer, navigation, etc - because the semantic HTML elements will be used instead ! (more about semantic HTML - see next title below)

CSS code :

```
#colyell {background:yellow;}  
#colred {background:red;}
```

HTML code :

```
<div id="colyell">  
<ul>  
<li>watermelon</li>  
<li>boat</li>  
<li>swimmer</li>  
</ul>  
</div>
```

```
<div id="colred">  
<ul>  
<li>wooden fire</li>  
<li>heat</li>  
<li>burning</li>  
</ul>  
</div>
```

Semantic HTML

The semantic HTML elements (HTML5) allows you to group your content. Instead of using `<div>` for grouping content like `<div id="header">` you can directly using the semantic element `<header>`.

This will also be positively noticed by the search engines.

Examples :

```
<header>
  <p>here comes the header</p>
</header>
```

```
<nav>
  <p>here comes the navigation menu</p>
</nav>
```

```
<main>
  <p>here comes the main content</p>
</main>
```

```
<footer>
  <p>here comes the footer</p>
</footer>
```


Miscellaneous

@import and multiple CSS files

Multiple CSS files helps you to structure the styles for your Website. For example you have all your general style properties in one file, your typography properties in a second, your colors in a third and your layout grids in a fourth one.

CSS code of the main.css file :

```
@import url("basics.css");  
@import url("typography.css");  
@import url("colors.css");  
@import url("layout.css");
```

Your **@import** must come before all other content in your CSS. Even putting comments before the **@import** rule will cause your imports to fail. So be sure to do your imports before you do anything else.

HTML code :

```
<head>  
<link rel="stylesheet" type="text/css" href="master.css" title="default"/>  
</head>
```

Validate the HTML and CSS code

Your CSS code will work properly if the HTML code is valid, so check them both!

[HTML checker](#)

[CSS checker](#)

Check for browser compatibility

Use different browsers (on different OS) to check your work. There are also [online tools](#).