



CSS

Cascading Style Sheets

Mr. K's Technology



Welcome to HTML.net
Free tutorials on HTML and CSS
<http://www.html.net/>

Mr. K's
Technology

Information was obtained for this Unit

CSS
Cascading Style Sheets
Mr. K's Technology



ABOUT
WEB
ELEMENTS

CSS}



CSS}

<http://www.w3schools.com/css/>

Introduction - Free CSS Tutorial



CSS is a style language that defines layout of HTML documents Cascading Style Sheets (CSS) is a fantastic tool to add layout to your websites. It can save you a lot of time and it enables you to design websites in a completely new way. CSS is a must for anyone working with web design.

This tutorial will get you started with CSS in just a few hours. It is easy to understand and it will teach you all the sophisticated techniques.

Learning CSS is fun. As you go along through the tutorial, remember to take enough time to properly experiment with what you learn in each lesson.

Using CSS requires basic experience with HTML. If you are not familiar with HTML, please start with our HTML tutorial before moving on to CSS.

Which software do I need?

Please avoid using software such as FrontPage, DreamWeaver or Word with this tutorial. Advanced software will not help you learn CSS. Instead, it will limit you and slow down your learning curve significantly.

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. Pico for Linux or Simple Text for Macintosh.

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 browser with this tutorial. 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!



<http://www.html.net/>



Will W3Schools always be Completely Free?

Yes. W3Schools will always be free.

Get Your Diploma!

W3Schools' Online Certification Program is the perfect solution for busy professionals who need to balance work, family, and career building.



What is CSS?

-  CSS stands for Cascading Style Sheets
-  Styles define how to display HTML elements
-  Styles are normally stored in Style Sheets
-  Styles were added to HTML 4.0 to solve a problem
-  External Style Sheets can save you a lot of work
-  External Style Sheets are stored in CSS files
-  Multiple style definitions will cascade into one



<http://www.w3schools.com/css/>



Lesson 1: What is CSS?

Maybe you already heard about CSS without really knowing what it is. In this lesson you will learn more about what CSS is and what it can do for you.

CSS is an acronym for Cascading Style Sheets.
What can I do with CSS?

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. Just wait and see!

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 only a few lessons of this tutorial 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.

Okay, it sounds a bit technical and confusing. But please continue reading. It will all make sense to you soon.

Back in the good old days when a guy called Tim Berners Lee invented the World Wide Web, 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 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, separation of the presentation style of documents from the content of documents, makes site maintenance a lot easier.

Which benefits will CSS give me?

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.





Styles Solve a Common Problem

HTML tags were originally designed to define the content of a document. They were supposed to say "This is a header", "This is a paragraph", "This is a table", by using tags like `<h1>`, `<p>`, `<table>`, and so on. The layout of the document was supposed to be taken care of by the browser, without using any formatting tags.

As the two major browsers - Netscape and Internet Explorer - continued to add new HTML tags and attributes (like the `<font>` tag and the color attribute) to the original HTML specification, it became more and more difficult to create Web sites where the content of HTML documents was clearly separated from the document's presentation layout.

To solve this problem, the World Wide Web Consortium (W3C) - the non profit, standard setting consortium, responsible for standardizing HTML - created STYLES in addition to HTML 4.0.

All major browsers support Cascading Style Sheets.



Style Sheets Can Save a Lot of Work

Styles sheets define HOW HTML elements are to be displayed, just like the font tag and the color attribute in HTML 3.2. Styles are normally saved in external .css files. External style sheets enable you to change the appearance and layout of all the pages in your Web document, just by editing one single CSS document!

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 you can define a style for each HTML element and apply it to as many Web pages as you want. To make a global change, simply change the style, and all elements in the Web are updated automatically.



Multiple Styles Will Cascade Into One

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.

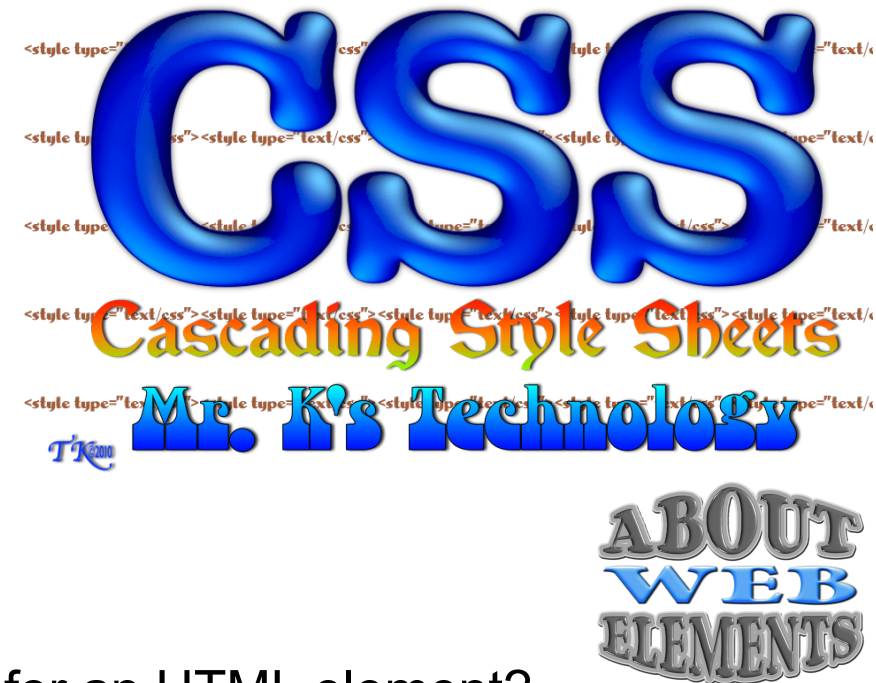
Cascading Order

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 (inside the <head> tag)
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 declared inside the <head> tag, in an external style sheet, or in a browser (a default value).



Lesson 1a: How does CSS work?

In this lesson you 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.

The basic CSS syntax

Let's say we want a nice red color as the background of a webpage:

Using HTML we could have done it like this:

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

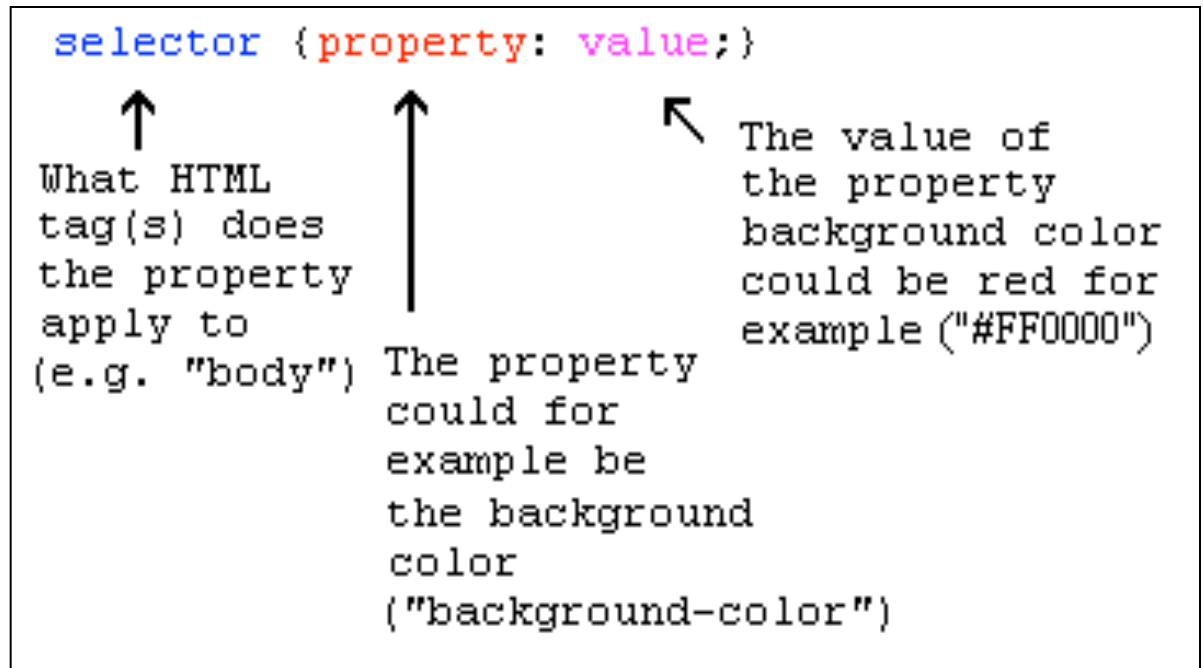
With CSS the same result can be achieved like this:

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



```
<body bgcolor="#FF0000"> HTML  
body {background-color: #FF0000;} CSS
```

As you will note, the codes are more or less identical for HTML and CSS. The previous example also shows you the fundamental CSS model:



Syntax

The CSS syntax is made up of three parts: a selector, a property and a value: selector {property: value}

The selector is normally the HTML element/tag you wish to define, the property is the attribute you wish to change, and each property can take a value. The property and value are separated by a colon, and surrounded by curly braces: The selector is normally the HTML element/tag you wish to define, the property is the attribute you wish to change, and each property can take a value. The property and value are separated by a colon, and surrounded by curly braces:

```
body {color: black}
```

Note: If the value is multiple words, put quotes around the value:

```
p {font-family: "sans serif"}
```

Note: If you wish to specify more than one property, you must separate each property with a semicolon.

The example below shows how to define a center aligned paragraph, with a red text color:

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

To make the style definitions more readable, you can describe one property on each line, like this:

```
p
{
text-align: center;
color: black;
font-family: arial
}
```



Grouping

You can group selectors. Separate each selector with a comma. In the example below we have grouped all the header elements. All header elements will be displayed in green text color:

```
h1,h2,h3,h4,h5,h6  
{  
color: green  
}
```

The class Selector

With the class selector you can define different styles for the same type of HTML element.

Say that you would like to have two types of paragraphs in your document: one right-aligned paragraph, and one center-aligned paragraph. Here is how you can do it with styles:

```
p.right {text-align: right;}  
p.center {text-align: center;}
```

You have to use the class attribute in your HTML document:

```
<p class="right">  
This paragraph will be right-aligned.  
</p>
```

```
<p class="center">  
This paragraph will be center-aligned.  
</p>
```



Note: To apply more than one class per given element, the syntax is:

```
<p class="center bold">  
This is a paragraph.  
</p>
```

The paragraph above will be styled by the class "center" AND the class "bold".

You can also omit the tag name in the selector to define a style that will be used by all HTML elements that have a certain class. In the example below, all HTML elements with class="center" will be center-aligned:

```
.center {text-align: center}
```



NOTE Do NOT start a class name with a number! It will not work in Mozilla/Firefox.



NOTE Do NOT start a class name with a number! It will not work in Mozilla/Firefox.

In the code below both the h1 element and the p element have class="center". This means that both elements will follow the rules in the ".center" selector:

```
<h1 class="center">
```

This heading will be center-aligned

```
</h1>
```

```
<p class="center">
```

This paragraph will also be center-aligned.

```
</p>
```



Add Styles to Elements with Particular Attributes

You can also apply styles to HTML elements with particular attributes.

The style rule below will match all input elements that have a type attribute with a value of "text":

```
input[type="text"] {background-color: blue}
```

The id Selector

You can also define styles for HTML elements with the id selector. The id selector is defined as a #.

The style rule below will match the element that has an id attribute with a value of "green":

```
#green {color: green}
```



The style rule below will match the p element that has an id with a value of "para1":

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

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



CSS Comments

Comments are used to explain your code, and may help you when you edit the source code at a later date. A comment will be 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  
}
```





How to Insert a Style Sheet

When a browser reads a style sheet, it will format the document according to it. There are three ways of inserting a style sheet:





Method 1: In-line (the attribute style)

One way to apply CSS to HTML is by using the HTML attribute style. Building on the above example with the red background color, it can be applied like this:

```
<html>
  <head>
    <title>Example</title>
  </head>
  <body style="background-color: #FF0000;">
    <p>This is a red page</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, such as when a style is to be applied to a single occurrence of an element.

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>
```

Method 2: Internal (the tag style)

Another way is to include the CSS codes using the HTML tag `<style>`. For example like this:

```
<html>
  <head>
    <title>Example</title>
    <style type="text/css">
      body {background-color: #FF0000;}
    </style>
  </head>
  <body>
    <p>This is a red page</p>
  </body>
</html>
```



Internal Style Sheet

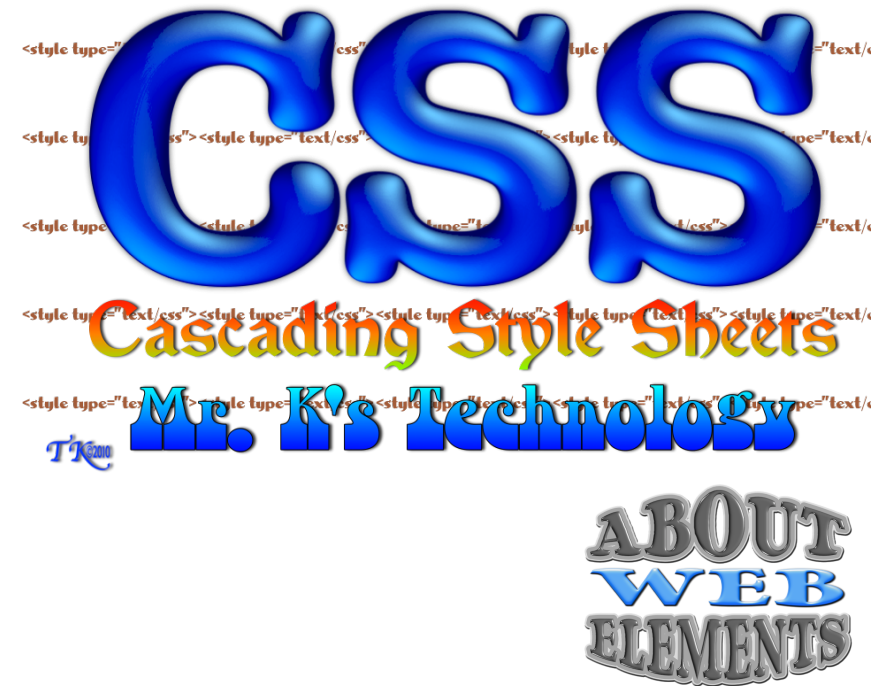
An internal style sheet should be used when a single document has a unique style. You define internal styles in the head section by using the <style> tag, like this:

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

The browser will now read the style definitions, and format the document according to it.

Note: A browser normally ignores unknown tags. This means that an old browser that does not support styles, will ignore the <style> tag, but the content of the <style> tag will be displayed on the page. It is possible to prevent an old browser from displaying the content by hiding it in the HTML comment element:

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



Method 3: External (link to a style sheet)

The recommended method is to link to a so-called external style sheet. Throughout this tutorial we will use this method in all our examples.

An external style sheet is simply a text file with the extension .css. Like any other file, you can place the style sheet on your web server or hard disk.

For example, let's say that your style sheet is named style.css and is located in a folder named style. The situation can be illustrated like this:

The folder "style" containing the file "style.css"

The trick is to create a link from the HTML document (default.htm) to the style sheet (style.css). Such link can be created with one line of HTML code:

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

Notice how the path to our style sheet is indicated using the attribute href.

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>
```

The browser will read the style definitions from the file mystyle.css, and format the document according to it.

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")}
```

NOTE Do NOT leave spaces between the property value and the units! If you use "margin-left: 20 px" instead of "margin-left: 20px" it will only work properly in IE6 but it will not work in Mozilla/Firefox or Netscape.



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.



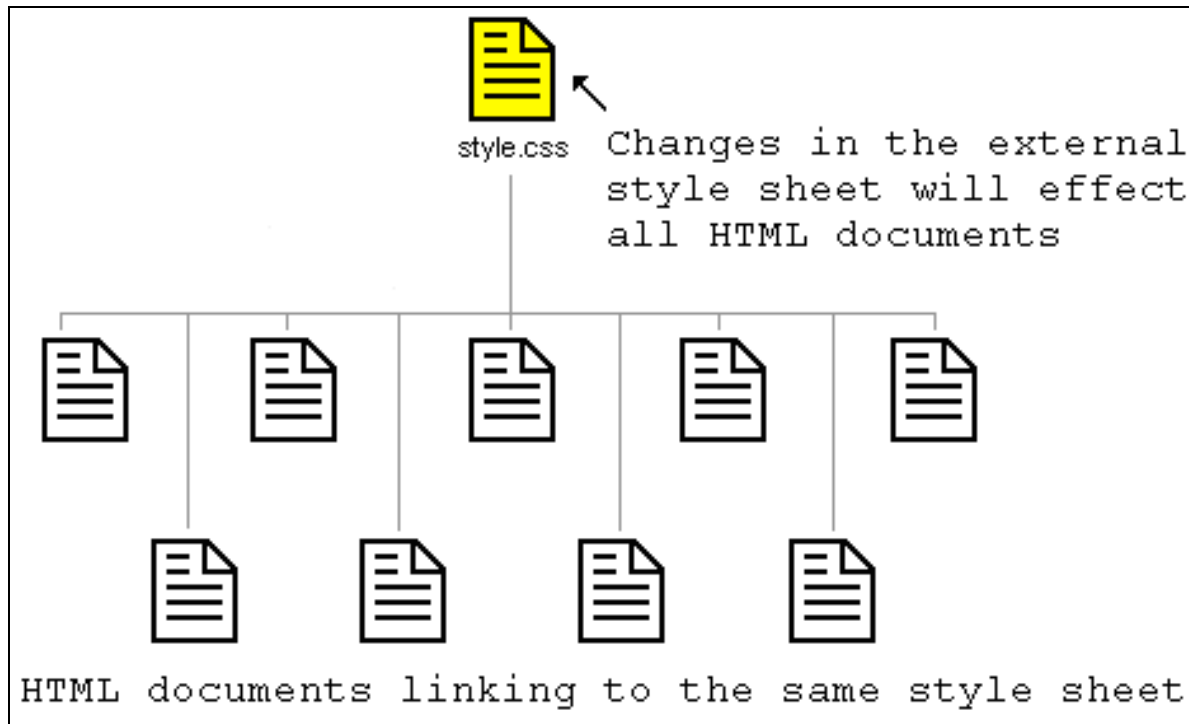
The line of code must be inserted in the header section of the HTML code i.e. between the <head> and </head> tags. Like this:

```
<html>
  <head>
    <title>My document</title>
    <link rel="stylesheet" type="text/css" href="style/style.css" />
  </head>
  <body>
    ...
```

This link tells the browser that it should use the layout from the CSS file when displaying the HTML file.



The really smart thing is that several HTML documents can be linked to the same style sheet. In other words, one CSS file can be used to control the layout of many HTML documents.



This technique can save you a lot of work. If you, for example, would like to change the background color of a website with 100 pages, a style sheet can save you from having to manually change all 100 HTML documents. Using CSS, the change can be made in a few seconds just by changing one code in the central style sheet.

CSS
Cascading Style Sheets
Mr. K's Technology

ABOUT
WEB
ELEMENTS

Try it yourself

Open Notepad (or whatever text editor you use) and create two files - an HTML file and a CSS file - with the following contents:

default.html

```
<html>
<head>
  <title>My document</title>
  <link rel="stylesheet" type="text/css" href="style.css" />
</head>
<body>
  <h1>My first stylesheet</h1>
</body>
</html>
```

style.css

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

Now place the two files in the same folder. Remember to save the files with the right extensions (respectively ".html" and ".css")

Open default.html with your browser and see how the page has a red background. Congratulations! You have made your first style sheet!



Your Project



SAVE YOUR WORK

CLOSE ALL OPEN PROGRAMS

ABOUT
WEB
ELEMENTS

LOG OFF

CSS

Cascading Style Sheets

Mr. K's Technology