



# Cascading Style Sheets (CSS)

## Webworks – A Workshop Series in Web Design (Session Five)

---

Table of Contents:

1. Introduction
2. Introduction to Syntax
3. Implementation
4. Main Tags

## 1. Introduction

### What is CSS?

CSS stands for Cascading Style Sheets and is used to achieve a level of uniformity easily throughout your web pages. You can apply a set of styles to as many html files as you want, allowing you to control the style of many web pages all at once. For example, we can specify that all the background colors be mauve, or that all text be written in Arial font. CSS is written in a language a little different from HTML.

### Why do we need CSS?

When you are working on a large website project that spans a number of HTML pages, CSS saves time and ensures consistency throughout your site. If you suddenly decide to change all the headings on your website to a different font, instead of changing each file and hoping you haven't missed any, you can just change one CSS style and have it apply across all pages. Also, the separation of style code from content code makes reading source code much easier and cleaner. This seems trivial for simple pages but becomes a major problem with complex web pages with hundreds of lines of code.

## 2. Introduction to Syntax

CSS syntax follows a three step process. You first select the element you want to style, and then pick a property of the element you want to style and finally the value of this property.

The basic syntax of a style is as follows:

```
selector { property: value; }
```

An example would be:

```
BODY { background-color: yellow; }
```

**Selector:** In this example, the *selector* is "body." It specifies that this is the element you wish to apply the style to. The selector could either be an HTML tag such as <P>, <LI>, or <H1>, or an element you define yourself (discussed at the end).

Note on Selectors: Selectors do not have the "<>" brackets. If an HTML element is selected, such as LI or P, then all the P's and LI's in your page will be affected. To create flexibility in the elements you affect with CSS, use internal, inline and custom selectors, discussed later.

**Properties and Values:** “background-color” is the *property* within the selector that will be modified. “yellow” is the *value* that the property is changed to. It is possible to include more than one property within each selector. In this case, a semicolon is used to separate the properties. An example of this is shown below.

```
H1      {background-color: transparent; border-style: solid}
selector {property:          value;          property:          value}
```

Like in HTML, white space in CSS, including new lines, do not affect the display or results.

### 3. Implementation: External, Internal, and Inline

Now you may be wondering, where do I stick all this CSS code? Well there are 3 ways to use CSS.

#### External

This is the most common implementation of styles. The CSS code is in a separate file with a “.css” extension (**NOTE: You must have the .css extension**). A snippet is put in the <HEAD> section of the HTML file specifying where the style sheet is. For example, if you had a style sheet called main.css in your styles folder under your web directory, your HTML will be:

```
<HEAD>
<LINK REL="stylesheet" TYPE="text/css" HREF="../styles/main.css" />
</HEAD>
```

The CSS file would then just contain your CSS code **AND NOTHING ELSE**. For example if all your style sheet did was set the background color of your pages to yellow, the CSS file would look like this:

```
BODY {background-color: yellow}
```

Using external style sheets makes it easy to apply the style sheet to multiple pages. Even better, any changes you make to the source style sheet *cascades* and updates the styling of all your pages.

#### Internal/Embedded

Say you apply an external style sheet to your page, but then want just one page to have a blue background. Then you can include the page specific CSS code within the <HEAD> section of your page. While other styles of your external style sheet come through, the background color style of the external sheet will be overridden by the internal stylesheet in the page. Now the CSS code needs to be wrapped with special <STYLE> tags in the HTML:

```
<head>
<style type="text/css">
<!--BODY { background-color: blue;}-->
</style>
</head>
```

## Inline

Inline uses of CSS is generally not recommended and is slowly being faded out. Inline CSS is where you stick the style directly inside a HTML tag. For example:

```
<P STYLE="color:green">
The text in this paragraph would then be green.
</P>
```

The only time you would use Inline CSS is if you need one instance of CSS, say highlighting a sentence or something that would be difficult to do with other HTML methods.

You can use more than one of these implementations. When they conflict, the order of precedence is as follows:

1. **Inline styles**
2. **Internal styles**
3. **External styles**

## Custom Selectors

Besides selecting HTML elements to apply styles to, you can also create your own custom element names to apply to any element. Custom styles take two forms, CLASS and ID. CLASS styles can be attached to multiple elements while ID styles can only be attached to one element. The syntax for both is as follows:

### CLASS

```
.somename { color: green; }
```

### ID

```
#someothername {color: red; }
```

CLASS must be preceded by a period. ID must be preceded by a hash "#". Now in my page I can do this:

```
<table class="somename">
  <tr>
    <td id="someothername"> This text would be red. </td>
    <td> This text would be green.</td>
  </tr>
</table>
```

The CLASS "somename" sets all the text in the table to green. But the ID "someothername" in the first TD sets the text in that TD to red, overriding the green.

## 4. Main Tags

Here are just a few listings of common style tags you'll be using.

For more information and styles: [http://www.w3schools.com/css/css\\_syntax.asp](http://www.w3schools.com/css/css_syntax.asp)

### Background

Property	Values	Description
Background-color	Red, blue, FFFFFFFF, transparent, etc	Sets the background color
Background-image	url	Sets the background image

### Border

Property	Values	Description
Border-color	Red, blue, FFFFFFFF, etc	Sets the color of the border
Border-style	Hidden, dotted, dashed, solid, double, groove, ridge, inset, outset	Sets the style of the border
Border-width	Thin, medium, thick, pixels	Sets the width of the border
Border-bottom-color	See border-color	Sets the color of the bottom border (bottom can be replaced by left, right, or top)
Border-bottom-style	See border-style	Sets the style of the bottom border (bottom can be replaced by left, right, or top)
Border-bottom-width	See border-width	Sets the style of the bottom width (bottom can be replaced by left, right, or top)

### Classification

Property	Values	Description
Cursor	Auto, crosshair, default, pointer, move, text, wait, help	Sets the type of cursor to display
Display	None, inline, block, list-item, run-in, compact, marker, table, inline-table, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-row, table-column-group, table-column, table-cell, table-caption	Sets how an item is displayed

Float	Left, right, none	Sets where an item will appear within another
Position	Static, relative, absolute, fixed	Sets where to put the item
Visibility	Visible, hidden, collapse	Sets the visibility mode of the item

**Dimension**

Property	Values	Description
Height	Auto, pixels, 30%	Sets the height of an item
Line-height	Normal, #, pixels, 30%	Sets the distance between lines
Max-height	None, length, %	Sets the maximum height of an item
Min-height	None, length, %	Sets the minimum height of an item
Max-width	Length, %	Sets the maximum width of an item
Min-width	Length, %	Sets the minimum width of an item
width	Auto, length, %	Sets the width of an item

**Font**

Property	Values	Description
Font-family	Family name (Arial) or a generic name (serif)	Sets the font family (Allows a list by priority)
Font-size	xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, length, %	Sets the font size
Font-stretch	Normal, wider, narrower, ultra-condensed, extra-condensed, condensed, semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded	Stretches or condenses the font
Font-style	Normal, italic, oblique	Sets the style of the font
Font-weight	Normal, bold, bolder, lighter, 100-900	Sets the weight of the font

**List**

Property	Values	Description
List-style-type	None, disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha	Sets the type of the list marker
List-style-position	Inside, outsider	Sets where the marker is placed
List-style-image	None, url	Sets an image for the list marker
Marker-offset	Auto, length	Specifies by how much the marker is to be offset

**Margin**

Property	Values	Description
Margin-bottom	Auto, length, %	Sets the bottom margin of the element
Margin-left	Auto, length, %	Sets the left margin of the element
Margin-right	Auto, length, %	Sets the right margin of the element
Margin-top	Auto, length, %	Sets the top margin of the element

**Outline**

Property	Values	Description
Outline-color	Color, invert	Sets the color of the outline around an item
Outline-style	None, dotted, dashed, solid, double, groove, ridge, inset, outset	Sets the style of the outline around an item
Outline-width	Thin, medium, thick, length	Sets the width of the outline around an item

**Padding**

Property	Values	Description
Padding-bottom	Length, %	Sets the padding on the bottom of an item
Padding-left	Length, %	Sets the padding on the left of an item
Padding-right	Length, %	Sets the padding on the right of an item
Padding-top	Length, %	Sets the padding on the top of an item

**Position**

Property	Values	Description
Bottom	Auto, %, length	Sets how far from the bottom of the parent item the current item is
Left	Auto, %, length	Sets how far from the left of the left

		of the parent item the current item is
Right	Auto, %, length	Sets how far from the right of the right of the parent item the current item is.
Top	Auto, %, length	Sets how far from the top of the parent item the current item is
Clip	Shape, auto	Clips the item into a specific shape
Overflow	Visible, hidden, scroll, auto	Sets what is to happen if the item overflows its given area
Vertical-align	Baseline, sub, super, top, text-top, middle, bottom, text-bottom, length, %	Sets the vertical alignment of an item
z-index	Auto, #	Sets the stack order of an item

**Table**

Property	Values	Description
Border-collapse	Collapse, separate	Sets the border of a table to collapse or separate
Border-spacing	Length	Sets the distance between borders of two cells
Empty-cells	Top, bottom, left, right	Sets whether empty cells should have a border
Table-layout	Auto, fixed	Sets how the table is to be laid out

**Text**

Property	Values	Description
Color	Blue, green, FFFFFFFF, etc	Sets the color of the text
Direction	Ltr, rtl	Sets the direction of the text
Letter-spacing	Normal, length	Changes the space between characters
Text-align	Left, right, center, justify	Aligns the text
Text-decoration	None, underline, overline, line-through, blink	Decorates the text
Text-indent	Length, %	Indents the first line of text
Text-shadow	None, color, length	Shadows the text
Text-transform	None, capitalize, uppercase, lowercase	Transforms the text
White-space	Normal, pre, nowrap	Decides how white space is handled
Word-spacing	Normal, length	Changes the space between words