

Cascading Style Sheets Tutorial

style sheets tutorial

part 1: [style sheet syntax](#)

how style sheets are 'written out'

part 2: [applying style sheets](#)

attaching style sheets to html documents

part 3: [class and ID](#)

giving one html tag multiple styles

part 4: [style sheet problems](#)

avoiding the pitfalls of style sheets

part 5: [tags for style sheets](#)

about div and span

style sheets examples

[style sheets and cursors](#)

changing the cursor with css (IE)

[style sheets and forms](#)

change the appearance of forms with css

[style sheets and links](#)

multi colors, remove the underline, more

[style sheets and borders](#)

catch the eye with colorful borders

[style sheets and html lists](#)

use an image, not a disc character



Cascading Style Sheets Syntax

Style sheets work by controlling the value(s) of the property(s) of a selector

css example of syntax

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

almost any html tag can be used as a selector

in the example below the <TT> tag is our selector, the property is color and the value is red

```
TT {color: red;}
```

<TT>This is typewriter text with style</TT>

for clarity style sheets are usually written out like this...

```
TT {
  COLOR: red;
  FONT-FAMILY: arial;
}
H1 {
  COLOR: red;
  FONT-FAMILY: tahoma, arial;
}
```

in the style sheet example above the value of the H1 font-family property is set to 'tahoma' with 'arial' as the back-up font. If 'tahoma' is unavailable on the users computer then the selector (H1) will have arial as it's default font

Applying Style Sheets

this tutorial shows four main methods of applying style sheets to a html document, these are

1. [embedding style sheets in the <HEAD></HEAD> part of a webpage](#)
2. [inline style sheets in a html tag](#)
3. [linking to an external style sheet](#)
4. [importing style sheets](#)

1. Embedding cascading style sheets in the <HEAD>

in this example we change the color of a [link](#)
we can suggest the color of links, visited (vlink) and active links (alink) in the body tag

```
<BODY link="blue" vlink="purple" alink="red">
```

adding the style sheet below to the head of a html document will cause [links](#) to be green when the mouse cursor hovers over them

```
<HEAD>
<STYLE>
<!--
A:HOVER {
    color: green;
}
-->
</STYLE>
</HEAD>
```

but style sheets can do much more than that, if you are using internet explorer you may have noticed that the links not only change color but the underline is also removed.

here's an example of how to achieve this

```
A:HOVER {
    color: green;
    text-decoration: none;
}
```

we can use the style sheet to control the font size, font color, background color and cursor (in Internet Explorer)

```
A:HOVER {
    color: #FF0066;
    background-color: transparent;
    text-decoration: none;
    cursor: hand;
    font-size: 12pt;
}
```

hold your cursor over this link [Yahoo](#)

2. inline style sheets in a html tag

style **properties** can easily be included in individual html tags like this...

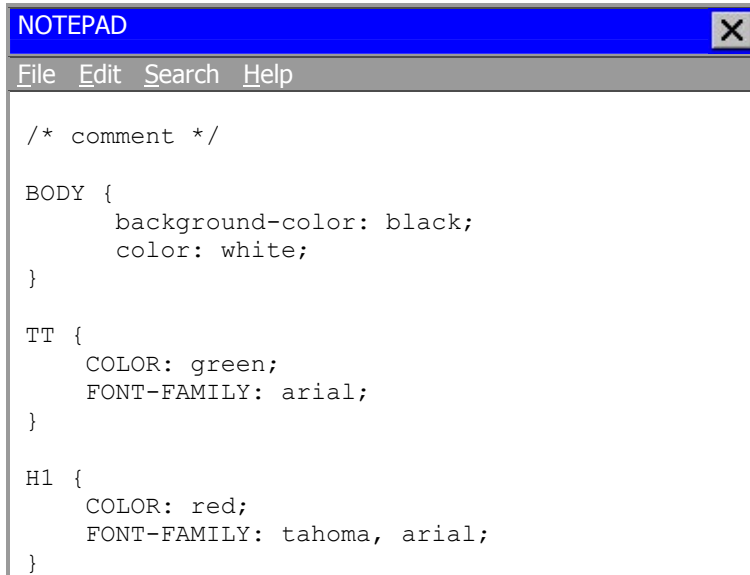
```
<A HREF="Yahoo" STYLE="color: red; text-decoration: underline;">Yahoo</A>
```

```
<TT STYLE="color: yellow; font-family: arial; background-color: purple;">
This is typewriter text with an inline stylesheet
</TT>
```

3. external style sheets

the ability to use an external style sheet adds a tremendous amount of flexibility to html documents, because any amount of pages can link to the same .css document a uniform look can be applied to an entire website
the benefits of an external style sheet are that file size is reduced because individual html documents do not require a style sheet to be embedded as well as the obvious saving in time

an external style sheet can be written with most simple text editors and should be saved as whatever-you-want.css but for the sake of argument lets call it style.css



```
NOTEPAD
File Edit Search Help

/* comment */

BODY {
    background-color: black;
    color: white;
}

TT {
    COLOR: green;
    FONT-FAMILY: arial;
}

H1 {
    COLOR: red;
    FONT-FAMILY: tahoma, arial;
}
```

notice that comments can be added to the style sheet between the `/*slash|star*/` but html tags should not be included

```
/* comments go in here */
```

below is an example of the link that would be used for style.css and is placed in the HEAD part of a html document

```
<HEAD>
<LINK REL="STYLESHEET" TYPE="text/css" HREF="style.css">
</HEAD>
```

4. importing style sheets

An example of the import statement is included here for the sake of completeness, unless you are a hardcore coder you may prefer to skip this part and go straight to the links at the bottom of this page.

A style sheet may be imported with the **@import** statement.

The **@import** statement may be used in a .css file or inside the <style> element

```
<STYLE TYPE="text/css">
<!--
@import url(http://www.somewhere.com/style.css);
@import url(/somefolder/another.css);
TT { color: red; }
H1 { color: lime; }
-->
</STYLE>
```

The style sheet example above shows that css rules may be included but the **@import** statement must occur at the start of the style sheet.

Rules specified in the style sheet override rules in the imported style sheets, even if one of the imported style sheets contained

```
TT { color: blue;}
```

typewriter text would still be rendered in red

The order in which the style sheets are imported determines how they cascade. In the example above, if the first imported style sheet (style.css) specifies that PRE elements be shown in red and the second (another.css) style sheet specifies that PRE elements be shown in purple, in this case the second imported style sheet (another.css) overrides the first style sheet (style.css) and PRE elements would be rendered in purple.

Cascading Style Sheets Class and ID

how to give one html tag more than one appearance with the use of class and id

1. cascading style sheets and class

previously we looked at applying css values to tag (selector) properties

```
selector {property: value}  
TT {color: red;}
```

this page describes how style sheets can alter those values

the easiest way to explain class is to think of it as making-up a descriptive word to describe the effect we want to create

the example below forces any text between the <TT></TT> tags to be red in color and 8pt in size

```
TT {color: red; font-size: 8pt;}
```

<TT>This is typewriter text with style</TT>

however if the <TT></TT> tags are required to be a color other than red we can do this with a 'class'

(a descriptive word preceded by a dot or period)

the example below makes text between the <TT></TT> purple and 12pt in size

```
.purple {color: purple; font-size: 12pt;}
```

<TT>This is typewriter text with style</TT>

<TT class="purple">This is typewriter text with class</TT>

other tags can share a class

<B class="purple">This is bold text with class

2. cascading style sheets and ID

style sheet ID and class are almost the same thing except where a class is declared by being preceded by a dot (period) an ID is preceded by a # hash mark

```
#goblue {color: blue; font-size: 12pt;}
```

<TT>This is typewriter text with style</TT>

<TT class="purple">This is typewriter text with class</TT>

<TT ID="goblue">This is typewriter text with ID</TT>

<B ID="goblue">This is bold text with ID

The ability to reuse the same class or id in multiple html tags can be a great timesaver

Cascading Style Sheets Browser Compatibility

the trouble with style sheets

how to avoid problems with cascading style sheets

the rule when using css is to avoid depending on a style sheet completely, and use html to mimic the style sheet as closely as possible.

1. cascading style sheet compatibility

Although style sheets (css1) have been supported since netscape 4.0 and microsoft internet explorer 4 (partial css1 support from IE3) users do have the option of simply 'turning off' style sheets and this should be considered when using style sheets

It should also be noted that if a netscape user disables "javascript" (and some of them do!) css seems to be disabled automatically

2. netscape and style sheets

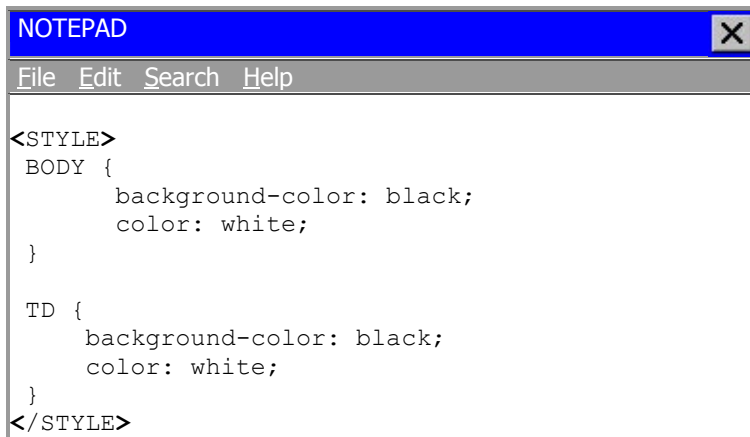
This is where it all falls down!

It is absolutely essential to check any html documents that use a style sheet, with a netscape browser, this cannot be emphasised enough, as effects can be rendered very differently than they would appear using a microsoft internet explorer browser particularly when style sheet elements are used in table cells.

Netscape seems to 'see' table cells as independent documents and renders the contents of a table cell as a default appearance, for instance although the "arial" font may be specified in the body tag any text in a table cell will be given a default appearance, which means whatever font the user has set as their default.

to overcome netscape problems with tables and style sheets there are one or two things we can do

one of which is use the style sheet to give the <TD> tag the same properties as the <BODY> tag

A screenshot of a Notepad window. The title bar is blue and says "NOTEPAD" with a close button on the right. Below the title bar is a menu bar with "File", "Edit", "Search", and "Help". The main text area contains the following CSS code:

```
<STYLE>
BODY {
    background-color: black;
    color: white;
}

TD {
    background-color: black;
    color: white;
}
</STYLE>
```

html tags that use a [class](#) seem to work well, and another simple solution is to use the <DIV> tag. find out about DIV and more in the next module.

3. a little advice

a style sheet can be used to affect almost any html tag and can have a great deal of power over the appearance of a html document, and the first temptation is to add css values to the most commonly used html tags, however it is strongly advised that style sheet properties are not given to the font tag as keeping this tag free gives a great deal of flexibility in text formatting

CSS, SPAN & DIV

consider div and span as html style sheet tags

style sheets and span

both div and span have no real meaning as html tags and only serve any use when a style sheet is applied

 is an inline element, which means it can start on the same line (like the font tag) and the <div> is a block level element, which means it must start on a new line (like a html table)

an example of span...

```
<STYLE>
SPAN {
    font-family: comic sans ms, garamond, arial;
    font-size: 14pt;
    font-weight: bold;
    color: #0000FF;
}
</STYLE>
```

this is span

div and style sheets

div, short for division, is an excellent way to very quickly add blocks of color to a html document

div can contain other block level elements, including tables

div seems to work best with a [class](#)

an example of div

```
<STYLE>
DIV {
    border-style: ridge;
    border-color: #FF0000;
    border-width: 5px;
    padding-left: 10px;
    padding-right: 10px;
    background: #009999;
    color: #00FF00;
}
</STYLE>
```

Style Sheets & Cursors

it is fairly simple to change the cursor in a html document with css, once the cursor names are known
there are 17 cursors to choose from

unfortunately this is another of those neat things that style sheets can achieve that netscape browsers just cant 'see' in a html document
internet explorer users will see the new cursor by hovering the cursor over the description and link cells

name	description	
crosshair	displays as a cross	link
auto	user's default setting or (windows settings)	link
pointer	same as auto	link
default	the normal arrow	link
hand	hand that appears when over a link	link
help	displays an arrow and question mark	link
move	a cross with arrows on the tips	link
text	like a cross section of an I-Beam	link
wait	an hourglass	link
n-resize	arrow pointing north (up)	link
ne-resize	arrow pointing North-East	link
e-resize	arrow pointing East (right)	link
se-resize	arrow pointing South-East	link
s-resize	arrow pointing South (down)	link
sw-resize	arrow pointing South-West	link
w-resize	arrow pointing West (left)	link
nw-resize	arrow pointing North-West	link

the cursor could be changed by changing the properties of the 'A:hover' selector in an embedded style sheet

```
<STYLE>
  A:hover {
    cursor: crosshair;
  }
</STYLE>
```

here is an inline style alternative

```
<A HREF="index.html" STYLE="cursor: move;">css tutorial</A>
```

Cascading Style Sheets & HTML Forms

with a cascading style sheet used to alter the value of any html form tag even the buttons
although not as effective in netscape, html forms can look spectacular with style sheets

The image shows a screenshot of an HTML form with a teal background and lime-colored text. The form contains the following elements:

- A text input field with the value "http://".
- A text input field.
- A text input field.
- A dropdown menu labeled "OPTIONS" with a downward arrow.
- Four checkboxes: the first is unchecked, the second is checked, and the third and fourth are unchecked.
- A text area containing the text "Hey !", "w hat do you think of the form?", and "cool huh?".
- Two buttons: "Send" and "Reset".

when viewed with an internet explorer the form above has lime colored text on a teal colored background

below are the style commands that were used to achieve these effects

```
<STYLE>
TEXTAREA {
    background-color: #009999;
    color: #00FF00;
    font-family: garamond, tahoma, arial;
    font-weight: bold;
}

.field {
    background-color: #009999;
    color: #00FF00;
    font-family: garamond, tahoma, arial;
    font-weight: bold;
}

.formbutton {
    color: #009999;
    text-transform: capitalize;
    cursor: hand;
}

SELECT {
    background-color: #009999;
    color: #00FF00;
    font-family: garamond, tahoma, arial;
    font-weight: bold;
}
</STYLE>
```

'TEXTAREA' obviously alters the text area, 'SELECT' alters the dropdown menu and the class '.field' alters the 'text input'

```
<INPUT type=text class="field">
```

if style commands were added to the 'INPUT' tag the form buttons would also appear to have lime text on a teal background, so the class 'formbutton' is used

```
<INPUT type=submit value=Send class="formbutton">
```

this gives the form buttons teal coloured text and capitalizes the first letter of each word and for users with Internet Explorer the cursor changes to a hand when the mouse is over a button



here is an image of the form for the benefit of users unable to see the effect

Style Sheets & Alternative Links

It can become necessary to add alternative links in addition to normal links, for instance a document with a 'navigation panel' which has a different background color may require links to be a lighter color than the links in the main document, or have a completely different style, removing the underline from links etc... the example below illustrates how to set up the main document links. whilst the link colors are still set in the body tag for the benefit of users that have browsers that are incompatible with style sheets or simply have css disabled, the style commands in this example describe some of the effects that can be achieved when applying style sheets to links

```
<HEAD>
<STYLE>
  A:link {
    color: #0000FF;
    text-decoration: underline;
    background: transparent;
  }

  A:active {
    color: #FF0000;
    text-decoration: none;
    background: #FFFF00;
  }

  A:visited {
    color: #9900FF;
    text-decoration: line through;
    background: transparent;
  }

  A:hover {
    color: #FF0000;
    text-decoration: overline;
    background: transparent;
  }

</STYLE>
</HEAD>
<BODY link=#0000FF alink=#FF0000 vlink=#9900FF>
```

both the body tag and the embedded style sheet will make links blue, active links red and visited links purple.

The additional properties declared in the style sheet give active links a yellow background, a strong contrast to their red color, the underline is also removed with the 'text-decoration: none;' statement. The style sheet makes visited links purple with a **line-through** and links change to red in color and are **overlined** as the cursor is passed over them.

To make alternative links, a class is added to each of the link attributes.

```
<HEAD>
<STYLE>
A:link {
    color: #0000FF;
}

A:active {
    color: #FF0000;
}

A:visited {
    color: #9900FF;
}

A:hover {
    color: #FF0000;
}

A.newlink:link {
    color: #FFFFFF;
}

A.newlink:active {
    color: #FF3300;
}

A.newlink:visited {
    color: #00FF00;
}

A.newlink:hover {
    color: #FFFF00;
}

</STYLE>
</HEAD>
<BODY link=#0000FF alink=#FF0000 vlink=#9900FF>
```

In the example above the class 'newlink' is used to specify alternative links, for the sake of clarity only the colors have been specified but of course text-decoration, background colors or any valid css [properties](#) could be added

Cascading Style Sheets & Borders

This example looks at border-width, border-color and border-style using a style sheet with the <P> tag

```
<STYLE>
P {
  border-style: double;
  border-color: #FF0000;
  border-width: 10px;
}
</STYLE>
```

this is a double border

the border-width is set to 10px (10 pixels)

the border-width can also be set to thin, medium or thick (border-width: medium)

below is a simple way to write out a border for the <P> tag

```
<P STYLE="border: double #FF0000 10px">
```

notice that just the 'border' command is used (not border-style, border-color or border-width)

the examples below demonstrate more border styles

border-style: ridge

border-style: solid

border-style: inset

border-style: outset

for more information on border and box commands see style sheet [properties](#)

Cascading Style Sheets & HTML Lists

html list properties can be altered in a number of ways with style sheets in the example below the value of white-space is set to normal which makes multiple spaces collapse into just one, other values that could be attributed to white-space are 'pre' which does not collapse multiple spaces and 'nowrap' which does not allow line wrapping without a
 tag.

```
LI {
  white-space: normal;
  list-style: square;
}
```

the style sheet also affects the value of the list-style, giving each list item a square style

- each list item
- starts with a square

other list-styles include disc, circle, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha and none

- circle (this for example)
- disc (this for example)
- decimal (numbers, like an ordered list, 1 2 3 4 5)
- lower-roman (lower case roman numerals, i ii iii iv v)
- upper-roman (upper case roman numerals, I II III IV V)
- lower-alpha (lower case, a b c d e)
- upper-alpha (upper case, A B C D E)
- none, it's gone

list-style is shorthand for list-style-type (discussed above), list-style-position and list-style-image


list-style-position, which determines where the next line of text starts if there is a line break in a list item, can be set to 'inside' or 'outside'

Inside	outside
○ first list item second line of item	○ first list item second line of item

list-style is shorthand for list-style-image. An image can easily be used for the list style instead of a circle, disc, square etc...

LI { list-style-image:	 first list item
	 second list item

```
url(disc.gif)
}
```

 third list item

since list-style is shorthand, the properties of list-style-type, list-style-position and list-style-image can be written out in one line

```
LI { list-style: square outside url(disc.gif)}
```

the list image seems to work in netscape occasionally and in general style sheets are more reliable in netscape when applied to a tag rather than an tag

another css value for html lists is 'display', which has four values and is included here for the sake of completeness but it is recommended that 'display' is not used

- **block (a line break before and after the element)**
- **inline (no line break before and after the element)**
- **list-item (same as block except a list-item marker is added)**
- **none (no display)**

```
LI { display: block;}
```