



CSS - selectors

Every CSS style definition has two components: the selector, which defines the tags to which the style will be applied, and the properties, which specify what the style actually does.

CSS text:

CSS

```
h1 {  
    colour:#000033;  
    font-family: 'Verdana', Tahoma, sans-serif;  
}
```

mark up text:

XHTML

```
<h1> content text to go in here </h1>
```

In this example the selector `h1` within the style sheet defines a dark blue colour as font colour and specifies Verdana as chosen font, Tahoma as alternative - and the use of the sans-serif default font in case none of the chosen fonts are found.

To apply this to any section of text the HTML will enclose it within the specified tag, starting tag `<h1>` - closing tag `</h1>`

There are a variety of different selectors you can use to gain control over the layout of the different elements of your page and their spatial relationship to each other.

Tag Selectors

The most basic form of a selector is the shown in the example above. By naming a particular HTML tag, you can apply a style definition to every occurrence of that tag in the document. This is often used to set the basic styles that will appear throughout a website.

For example, the following might be used to set the default font for a website:

CSS text:

CSS

```
body, p, td, th, div, blockquote, dl, ul, ol {  
    font-family: Tahoma, Verdana, Arial, Helvetica, sans-serif;  
    font-size: 1em;  
    color: #000000;  
}
```

This rather long selector is a list of tags, all of which will take on the style definition (font, size, and color). In theory, the `<body>` tag is all that's needed (as all the other tags appear inside the `<body>` tag, and would thus inherit its properties), but many browsers don't properly carry style properties into tables and other elements.





Pseudo-Class Selectors

A text hyperlink, the `<a>` tag in HTML is much more versatile in its formatting options than most other tags. Specifying `link`, `vlink` and `alink` properties inside the `<body>` tag of the HTML document allows you to give the link a different appearance and look for its different states - unvisited, visited and being clicked. Using CSS you can now add a 4th state to your link - a rollover state, ie what the link looks like when the mouse hover over it.

The following example shows 4 style definitions - always to be listed in the order shown below `<memory hint - mnemonic: LoVeHAtE>`:

CSS text:

CSS `a:link { color: #01435F; }
a:visited { color: #000000; }
a:hover { color: #FFFFFF; }
a:active { color: #01435F; }`

<code>a:link</code>		unvisited link, set to { <code>color: #01435F; </code> } light blue
<code>a:visited</code>		visited link, set to { <code>color: #000000; </code> } black
<code>a:hover</code>		mouse over link, set to { <code>color: #FFFFFF; </code> } white (overriding both previous style definitions)
<code>a:active</code>		link clicked, set to { <code>color: #01435F; </code> } dark blue (overriding all 3 previous style definitions)

Class Selectors

Assigning styles to tags does however limit you to the content of chosen tags - consistently within the whole HTML document. To differentiate between identical tags in different places on your page and to distinguish between different elements within a tag - define a class in your CSS.

CSS text:

CSS `p { color: #000000; }
.sidebar { color: #999999; }`

The example above shows the paragraph tag, defining all text within it to appear black. To create a visual difference to define the text within the sidebar - the second line of code uses a class selector (which starts with a full stop) to indicate an application of style to any tag of the `sidebar` class - here setting the sidebar text to grey.

To create a paragraph of text with the sidebar's properties - add the class attribute to the paragraph tag

mark up text:

XHTML `<p class="sidebar"> content to go here </p>`



Specifying any other details within the sidebar then - will also be possible by simply adding more style rule definitions.

For example - if the sidebar contains links that you'd like to appear in bold - you can add a specific style for any links within the side as shown here

CSS text:

```
CSS > p { color: #000000; }  
.sidebar { color: #999999; }  
a.sidebar:link, a.sidebar:visited {font-weight: bold;}
```

The example now will show all main text in black, any text within the sidebar in grey - and links within the sidebar still in grey but with the type set to bold. The links' colour could be changed by adding yet another style rule - as this rule is very specific and listed below the preceding, more general rules - it overrides previous settings.

mark up text:

```
XHTML > <p class="sidebar"> content to go here  
    <a href="samplepage.html" class="sidebar">samples</a>  
</p>
```

Contextual Selectors

As the example above shows - it could become quite time consuming to add the specific class to each and every link within the sidebar. If you used your sidebar purely for links - this would become quite a bloated section of coding.

This is when you use the contextual selectors - which allow you to provide a list of selectors based on the context, ie the tag/s that contain it.

CSS text:

```
CSS > p { color: #000000; }  
.sidebar { color: #999999; }  
p.sidebar a:link, p.sidebar a:visited {  
    font-weight: bold;  
    colour: #CCCCCC}
```

The example now is solved more efficiently: all links (specified with the `a:link` and the `a:visited` selector) within the sidebar paragraph (specified with the `p.sidebar` selector) are now using the set style, of bold font and lighter grey colour.

mark up text:

```
XHTML > <p class="sidebar"> content to go here  
<a href="samplepage.html">samples</a>.</p>
```



ID Selectors

The ID selectors again are used - as the class selectors - to select a particular tag, rather than a group of tags. Instead of the full stop however which defined a class - the ID selected is preceded by the hash #.

CSS text:

CSS

```
#sidebar { color: #999999; }
#sidebar a:link, #sidebar a:visited {
    font-weight: bold;
    colour: #CCCCCC}
```

Above, the previous example changed to use the ID selector instead of the tag and class selectors.

mark up text:

XHTML

```
<p id="sidebar"> content to go here
<a href="link.html">samples</a>.</p>
```