INT106 Lab – CSS (Credit: w3schools.com)

**What is CSS?**

* CSS stands for Cascading Style Sheets
* CSS defines how HTML elements are to be displayed
* Styles were added to HTML 4.0 to solve a problem
* CSS saves a lot of work
* External Style Sheets are stored in CSS files
* The CSS is a separate language with its own syntax.

|  |
| --- |
| One page, different CSS.  http://www.w3schools.com/css/demo\_default.htm |

**CSS Solved a Big Problem**

* HTML was NEVER intended to contain tags for formatting a document.
* HTML was intended to define the content of a document, like:

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

* When tags like <font>, and color attributes were added to the HTML 3.2 specification, it started a nightmare for web developers. Development of large web sites, where fonts and color information were added to every single page, became a long and expensive process.
* To solve this problem, the World Wide Web Consortium (W3C) created CSS.
* In HTML 4.0, all formatting could (and should!) be removed from the HTML document, and stored in a separate CSS file.

|  |
| --- |
| **The style definitions are normally saved in external .css files.**  With an external style sheet file, you can change the look of an entire Web site by changing just one file! |

**CSS Syntax**



Selector is the HTML element you want to style.

For example:

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

Or you may write the CSS to be easier to read as the following:

|  |
| --- |
| p {  Color: red;  text-align:center;}  } |

**Note**

CSS comments start with /\* and end with \*/. Yes, it is not like Java.

There are 3 types of the selector.

1. **The element selector** is used to style the HTML elements.

2. **The id selector** uses the id attribute of an HTML element to select a specific element. For example:

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

Use # to indicate that it is the ID.

The ID selector will then be referred to using the ID attribute. For example:

|  |
| --- |
| <h1 id = “para1”> |

**Note**

Do NOT start an ID name with a number.

3. **The class selector** selects elements with a specific class attribute. For example:

|  |
| --- |
| .center {  text-align: center;  color: red;  } |

All elements that use the class center will receive the style.

But if you alter the code a bit as the following:

|  |
| --- |
| p.center {  text-align: center;  color: red;  } |

All the **p elements** with the **center class** will receive the style.

**Grouping Selectors**

If you have elements with the same style definitions, like this:

|  |
| --- |
| h1 {     text-align: center;     color: red; }  h2 {     text-align: center;     color: red; }  p {     text-align: center;     color: red; }  You may write like the following instead:  h1, h2, p {     text-align: center;     color: red; } |

**3 ways to insert CSS.**

* External style sheet
* Internal style sheet
* Inline style sheet

**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 just one file.

Each page must include a link to the style sheet with the <link> tag. The <link> tag goes inside the head section:

|  |
| --- |
| <head>  <link rel="stylesheet" type="text/css" href="mystyle.css">  </head> |

**Note**

The CSS file should not contain any html tags. The style sheet file must be saved with a .css extension.

**NO** space between the property value and the unit (such as margin-left: **20 px**;). The correct way is: margin-left: **20px**;

**Internal style sheet** should be used when a single document has a unique style. You define internal styles in the head section of an HTML page, inside the <style> tag, like this:

|  |
| --- |
| <head>  <style>  body {  background-color: linen;  }  h1 {  color: maroon;  margin-left: 40px;  }  </style>  </head> |

**Inline style** loses many of the advantages of a style sheet (by mixing content with presentation). Use this method sparingly!

To use inline styles, add the style attribute to 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 h1 element:

|  |
| --- |
| <h1 style="color:blue;margin-left:30px;">This is a heading.</h1> |

**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, assume that an external style sheet has the following properties for the <h1> element:

|  |
| --- |
| h1 {  color: navy;  margin-left: 20px;  } |

Then, assume that an internal style sheet also has the following property for the <h1> element:

|  |
| --- |
| h1 {  color: orange;  } |

If the page with the internal style sheet also links to the external style sheet the properties for the <h1> element will be:

|  |
| --- |
| color: orange;  margin-left: 20px; |

The left margin is inherited from the external style sheet and the color is replaced by the internal style sheet.

**Multiple Styles Will Cascade into One**

Styles can be specified:

* inside an HTML element (Inline)
* inside the <head> section of an HTML page (Internal)
* in an external CSS file

Tip: Even **multiple external style sheets** can be referenced inside a single HTML document.

**Cascading order**

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 (in the head section)

4. Inline style (inside an HTML element) (Highest priority)

\*The higher priority overrides the lower ones.

**Note**

**BUT** if the link to the external style sheet is placed after the internal style sheet in HTML <head>, the external style sheet will override the internal style sheet!

**Background**

With CSS, a color is most often specified by:

a HEX value - like "#ff0000"

an RGB value - like "rgb(255,0,0)"

a color name - like "red"

See here: <http://www.w3schools.com/cssref/css_colors_legal.asp>  
for complete list of the colors.

**Background Image**

The background-image property specifies an image to use as the background of an element.

By default, the image is repeated so it covers the entire element.

The background image for a page can be set like this:

|  |
| --- |
| body {  background-image: url("paper.gif");  } |

**Background Image - Repeat Horizontally or Vertically**

By default, the background-image property repeats an image both horizontally and vertically.

Some images should be repeated only horizontally or vertically, or they will look strange, like this:

|  |
| --- |
| http://www.w3schools.com/css/tryit.asp?filename=trycss\_background-image\_gradient1 |

If the image is repeated only horizontally (repeat-x), the background will look better:

|  |
| --- |
| <http://www.w3schools.com/css/tryit.asp?filename=trycss_background-image_gradient2>  body {  background-image: url("gradient\_bg.png");  background-repeat: repeat-x;  } |

**Background Image - Set position and no-repeat**

|  |
| --- |
| <http://www.w3schools.com/css/tryit.asp?filename=trycss_background-image_norepeat>  body {  background-image: url("img\_tree.png");  background-repeat: no-repeat;  } |

In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

The position of the image is specified by the background-position property:

|  |
| --- |
| <http://www.w3schools.com/css/tryit.asp?filename=trycss_background-image_position>  body {  background-image: url("img\_tree.png");  background-repeat: no-repeat;  background-position: right top;  } |

**Background - Shorthand property**

As you can see from the examples above, there are many properties to consider when dealing with backgrounds.

To shorten the code, it is also possible to specify all the properties in one single property. This is called a shorthand property.

The shorthand property for background is simply "background":

|  |
| --- |
| body {  background: #ffffff url("img\_tree.png") no-repeat right top;  } |

When using the shorthand property the order of the property values is:

* background-color
* background-image
* background-repeat
* background-attachment
* background-position

\*It does not matter if one of the property values is missing, as long as the ones that are present are in this order.

**Background – Fixed**

With this property, the background will not move together with the text when scrolling.

|  |
| --- |
| <http://www.w3schools.com/css/tryit.asp?filename=trycss_background-attachment>  body {  background-image: url("w3css.gif");  background-repeat: no-repeat;  background-attachment: fixed;  } |

**Text**

**Text Alignment**

The text-align property is used to set the horizontal alignment of a text.

Text can be centered, or aligned to the left or right, or justified.

When text-align is set to "justify", each line is stretched so that every line has equal width, and the left and right margins are straight (like in magazines and newspapers).

|  |
| --- |
| h1 {  text-align: center;  }  p.date {  text-align: right;  }  p.main {  text-align: justify;  } |

**Text Decoration**

The text-decoration property is used to set or remove decorations from text.

The text-decoration property is mostly used to remove underlines from links for design purposes:

|  |
| --- |
| a {  text-decoration: none;  }  h1 {  text-decoration: overline;  }  h2 {  text-decoration: line-through;  }  h3 {  text-decoration: underline;  } |

**Text Transformation**

The text-transform property is used to specify uppercase and lowercase letters in a text.

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

|  |
| --- |
| p.uppercase {  text-transform: uppercase;  }  p.lowercase {  text-transform: lowercase;  }  p.capitalize {  text-transform: capitalize;  } |

**Text Indentation**

The text-indent property is used to specify the indentation of the first line of a text.

|  |
| --- |
| p {  text-indent: 50px;  } |

**Text Direction**

|  |
| --- |
| <http://www.w3schools.com/css/tryit.asp?filename=trycss_text_direction>  div.ex1 {  direction: rtl;  unicode-bidi: bidi-override;  } |

|  |  |  |
| --- | --- | --- |
| **Practice**  Open twenties.html in a text editor. Insert your code into the style element.  1. Make the h1 element “red”  2. Add a new rule that makes the h2 elements red as well.  3. Add a 100-pixel left margin to paragraph (p) elements using  this declaration:  margin-left: 100px;  4. Add a 100-pixel left margin to the h2 headings as well.  5. Add a red, 1-pixel border to the bottom of the h1 element using this declaration:   |  | | --- | | border-bottom: 1px solid red; |   6. Move the image to the right margin, and allow text to flow around it with the float property. The shorthand margin property shown in this rule adds zero pixels of space on the top and bottom of the image and 12 pixels of space on the left and right of the image (the values are mirrored in a manner explained in Chapter 14, Thinking Inside the Box).   |  | | --- | | img {  float: right;  margin: 0 12px;  } |   **7. Move all of your style into a .css file and use it as an external style sheet.** |

**Styling Links**

Links can be styled with any CSS property (e.g. color, font-family, background, etc.).

In addition, links can be styled differently depending on what state they are in.

The four links states are:

* a:link - a normal, unvisited link
* a:visited - a link the user has visited
* a:hover - a link when the user mouse over it
* a:active - a link the moment it is clicked

|  |
| --- |
| <http://www.w3schools.com/css/tryit.asp?filename=trycss_link>  /\* unvisited link \*/  a:link {  color: #FF0000;  }  /\* visited link \*/  a:visited {  color: #00FF00;  }  /\* mouse over link \*/  a:hover {  color: #FF00FF;  }  /\* selected link \*/  a:active {  color: #0000FF;  } |

\* When setting the style for several link states, there are some order rules:

a:hover MUST come after a:link and a:visited

a:active MUST come after a:hover

**Common Link Styles**

In the example above the link changes color depending on what state it is in.

Lets go through some of the other common ways to style links:

**The text-decoration property** is mostly used to remove underlines from links:

|  |
| --- |
| a:link {  text-decoration: none;  }  a:visited {  text-decoration: none;  }  a:hover {  text-decoration: underline;  }  a:active {  text-decoration: underline;  } |

**The background-color property** specifies the background color for links:

|  |
| --- |
| a:link {  background-color: #B2FF99;  }  a:visited {  background-color: #FFFF85;  }  a:hover {  background-color: #FF704D;  }  a:active {  background-color: #FF704D;  } |

More styles for the links.

<http://www.w3schools.com/css/tryit.asp?filename=trycss_link2>