

Creating a simple style file

Cascading Style Sheets or **CSS** files provide a way to control the look and feel of your web page that is more convenient, more flexible and more comprehensive than adding parameters to HTML tags (like `<td bgcolor="blue">`). Style sheets let you define a wide range of settings such as colors, font types, font sizes, boxes around regions on your web page and spacing between various types of elements on the page. Style sheets are especially useful because you can create one CSS file that defines the look and feel of multiple web pages.

1 Getting started

A CSS file is a separate file from an HTML file. You create the CSS file in a text editor.

First you need to add a tag to the HTML file to tell it that it should use the CSS file. There are several ways to do this. One common way is shown below, using the **link** tag.

```
<head>
<title>prof sklar's sample home page</title>
<link rel="stylesheet" href="sklar-style.css" type="text/css">
</head>
```

Note that the `<link...>` tag is contained inside the `<head>...</head>` tags.

Add the link tag (as above) to your HTML file.

Next (below) you will create a separate CSS file that will contain the style sheet **rules**. The name of the CSS file should end in **.css** and the full filename is the value of the `href` parameter in the `link` tag. Above, my CSS file is named: **sklar-style.css**.

2 Setting body properties

Open a new file in your text editor.

Inside the file, enter a **body** rule to set the properties of the body element in the HTML file. The example below sets the background and text colors.

Note that these are the same properties that we set earlier, by using the `<body>` tag and its parameters `bgcolor` and `text`. You can set more properties using a style sheet. Note that the property settings in the style sheet take precedence over any set directly in the HTML tags.

```
body {
  background: red;
  color: brown;
}
```

Note the use of punctuation:

- the open-curly-bracket `{` at the beginning of the rule,
- the close-curly-bracket `}` at the end of the rule,
- the colon `:` after the name of each property being set by the rule, and
- the semi-colon `;` after the value of each property.;

Enter the rule, above, using your text editor. Make sure that you specify different colors here than the ones in your HTML `<body>` tag, so that you can be sure that the CSS file is being picked up by the browser.

When you are ready to save the file, name it with your last name followed by **-style.css**. For example, my file is called: **sklar-style.css**

Remember, this file name should match the value of the href parameter in the <link> tag that you entered in the previous step.

After you save both files (HTML and CSS), reload the HTML page in your browser. You should see the new colors for the background and text that you specified in your CSS file.

3 Setting table properties

You can use a style sheet to set the properties of a table. For example, the syntax for changing the colors of all the table cells and text is:

```
table {
  background: magenta;
  color: pink;
}
```

Table properties that can be set include:

name	description	possible values
color	color of text in table cells	any valid color name or hex value
background	background color for table cells	any valid color name or hex value
font-family	font specification for text in table cells	examples: serif, sans-serif, monospace, courier, verdana
font-size	size of font in table cells	relative value (e.g., 200%) or absolute (e.g., 8pt) values

Look at the on-line references listed at the end of this document for a complete list of table properties (see CSS Full Specification).

Try setting the color properties of your table by adding a **table** rule to your CSS file. Place it in the file after the **body** rule.

Actually, it does not matter which order rules are placed in a CSS file. The important thing is to keep the rule together, starting with the name of the rule, followed by the open-curly-bracket, the list of property settings and then the close-curly-bracket.

4 Setting list properties

You can use a style sheet to set the properties of a list. For example, the syntax for changing the background color of the whole unordered list is:

```
ul {
  background: black;
}
```

Try adding a **ul** rule to your CSS file, as above. Save the file and reload your HTML file in the browser. See how it looks.

You can also set the properties of list items.

```
li {
  font-size: 16pt;
  border: solid;
  border-width: thin;
}
```

This example shows some new properties: **font-size**, **border** and **border-width**.

Try adding this **li** rule to your CSS file. Save the file and reload your HTML file in the browser. See how it looks.

5 Dividing up your web page

One of the most powerful aspects of style sheets is the ability to divide your web page into sections, or “divisions”. You can then set the properties within each division, regardless of whether you have lists, tables or whatever inside the division.

Using divisions requires two things:

(1) Specifying the extent of the division in the HTML file using `<div class=...NAME...> ... </div>` tags.

(2) Specifying properties for the division in your CSS file.

Bring up your HTML file in the text editor. Try adding a division containing some text, like the example below:

```
<div class=fun>
I'm having so much fun learning HTML!
</div>
```

Save the HTML file. Note that if you reload it now, you will not see any difference in how it displays.

Now bring your CSS file into the text editor. Add a rule that sets the properties inside your “fun” class, for example:

```
div.fun {
  font-size: 24pt;
  background: green;
}
```

Now save the CSS file and reload the HTML in the browser. Now you should see a difference!

A few more things about divisions:

- You can name the divisions whatever you want, but the names should not contain any spaces. Use the underscore character (`_`) if you want to have a multi-word name, e.g., `my_fun_division`.
- Notice that in the CSS file, the rule begins with **div.** followed by the name you specified as the value of the `class` parameter in the `<div>` tag.
- You can “nest” divisions, by defining new divisions inside other ones. For example:

```
<div class=main>
  <div class=part_one>
    this is the first part of the page.
  </div>
  <div class=part_two>
    this is the second part of the page.
  </div>
</div>
```

In this example, the divisions `part_one` and `part_two` are *nested* inside the division `main`.

On-line references

- Introduction to CSS:
 - <http://www.w3.org/Style/Examples/011/firstcss>
 - <http://www.w3.org/Style/LieBos2e/enter/>
 - <http://www.w3.org/MarkUp/Guide/Style.html>
 - <http://htmldog.com/guides/cssbeginner/>
- Advanced CSS:
 - <http://htmldog.com/guides/cssintermediate/>
 - <http://htmldog.com/guides/cssadvanced/>
 - <http://www.w3.org/Style/Examples/007/>
- CSS Reference:
 - <http://htmldog.com/reference/cssproperties/>
- CSS Full Specification:
 - <http://www.w3.org/TR/css21/> (CSS version 2.1)