Introduction to Cascading Style Sheets

Cascading Style Sheets (CSS) are used to add *styles* to web pages. Styles are such things as color, font type and size, margins, and cell spacing. There are two reasons to use style sheets. One is to separate style information from html markup and content. And the other is to keep consistent styles throughout a web site.

Style sheets are very simple. They consist of a list of html tags followed in curly braces by styles that are to be associated with them. For example, if a web page contains a table, a style sheet might have a listing for styles associated with the table.

```
    table
    {
        border-style: solid;
        border-width: thin;
        margin-left: 1.0cm;
        margin-top: 1.0cm;
    }
```

An Example CSS Document

```
/* Style sheet for an application, named project-styles.css. */
body
{
    background-image: url("images/white.gif");
    text-align: center;
    color: blue;
}
h1, h2, h3, h3, h4, h5, h6 {color: blue}
form {color: seagreen}
table {border-color: blue;}
```

Things to note:

- Comments may be inserted anywhere and come between /* and */.
- Semi-colons are used to separate styles and not to terminate them. They are required between styles but not at the end.
- Colons are used between the attribute and its value, i.e. color: blue.
- White space is ignored (space, tab, line-feed).
- A list of tags may all have the same style attached to them.
- Go to http://www.w3schools.com/ to find a list of styles supported by the W3C consortium.

Why Cascading?

There are four levels for style information.

- Inline styles: <body bgcolor = "cyan">.
- Internal style sheets: styles listed between <style type = "text/css"> … </style>.
- External style sheets; style sheets linked to web pages.
- Default values for the browser: such as font, "Times New Roman".

These styles *cascade* into one style set. An inline style is applied first. If there is a conflict, it has precedence. If there is no inline style set for a tag and there is an internal style sheet, it will be used. If neither of the first two exist for a tag and there is an external style sheet, it comes next. If none of the above exist, browser defaults will be applied.

Linking to an External Style Sheet

To link to an external style sheet, you have to add a line to the top of your web page.

```
	<link rel="stylesheet" type="text/css" href="project-styles.css" />
```

where project-styles.css is the name of the style sheet. If the sheet is not on the same directory as the web page, more detailed information about the path to it would be required.

```
<html>
<head>
	<link rel="stylesheet" type="text/css" href="project-styles.css" />
	<script language = "Javascript" src = "logonscript.js"></script>
	<title>Login Form</title>
</head>
<body>
	<h2>Please Login</h2>
	<form name = "LogonForm" method = "post" action = "../project/clogin">
		Username: <input type = "text" name = "username" /> <br />
		Password: <input type = "password" name = "password" /> <br />
		<input type = "submit" value = "Login" onClick = "return CheckForm (LogonForm)"/>
		<input type = "reset" value = "Reset" />
	</form>
</body>
</html>
```

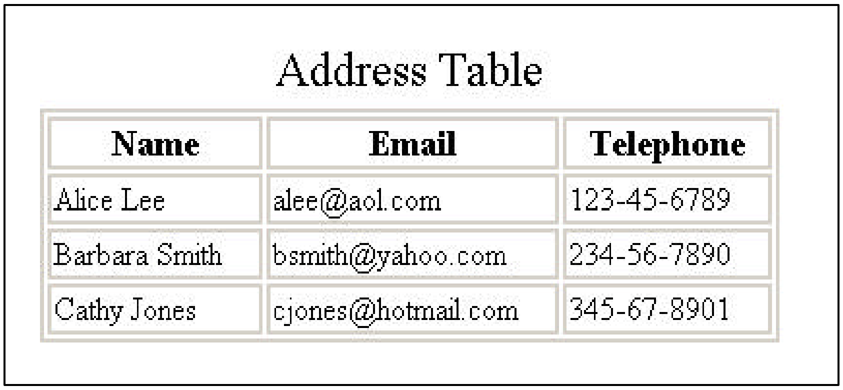
Note that this web page also has a link to a Javascript file, logonscript.js. This particular script is used to prevent the form being sent to the server if one of the boxes is empty.

Internal Style Sheets

Styles can also be included in the web page itself. This separates the styles from the tags, but it does not help with establishing uniformity across a number of pages. However, it is very convenient to use when you are developing your web page and style sheet. You only have one file to make changes in, not two.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
	<title>Table Example</title>
	<style type: "text/css">
	table {
		border: "1";
		border-style: "solid";
		border-width: "thin";
		cellpadding: "10";
	}
	caption {font-size: "large"}
	th {
		border-style: "solid";
		border-width: "thin";
```

```
                font-size: "medium";
        }
        td {
                border-style: "solid";
                border-width: "thin";
                padding-right: "0.5cm";
        }
        </style>
</head>
<body>
<table>
<caption>Address Table</caption>
<tr> <th>Name</th><th>Email</th><th>Telephone</th> </tr>
<tr> <td>Alice Lee</td><td>alee@aol.com</td><td>123-45-6789</td> </tr>
<tr> <td>Barbara Smith</td><td>bsmith@yahoo.com</td><td>234-56-7890</td> </tr>
<tr> <td>Cathy Jones</td><td>cjones@hotmail.com</td><td>345-67-8901</td> </tr>
</table>
</body>
</html>
```



Address Table

Name	Email	Telephone
Alice Lee	alee@aol.com	123-45-6789
Barbara Smith	bsmith@yahoo.com	234-56-7890
Cathy Jones	cjones@hotmail.com	345-67-8901

The class attribute

It is possible to have a style apply to a *class* that you define. A class attribute is added to a tag, and then all the styles defined for that class will be applied to that tag. For example, we can have two heading tags of the same size that have different styles applied to them.

```
        <h3 class="red-heading">This line is red.</h3>
        <h3 class="blue-heading ">This line is blue.</h3>
```

In the style sheet, class definitions begin with periods.

```
        <style type: "text/css">
                .red-heading {color: "red";}
                .blue-heading {color: "blue";}
        </style>
```

This means that you can distinguish between different uses of the same tag.

The entire web page follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
        <title>The class attribute</title>
        <style type: "text/css">
                .red-heading {color: "red";}
                .blue-heading {color: "blue";}
        </style>
</head>

<body>
<h3 class="red-heading">This line is red.</h3>
<h3 class="blue-heading">This line is blue</h3>
</body>
</html>
```

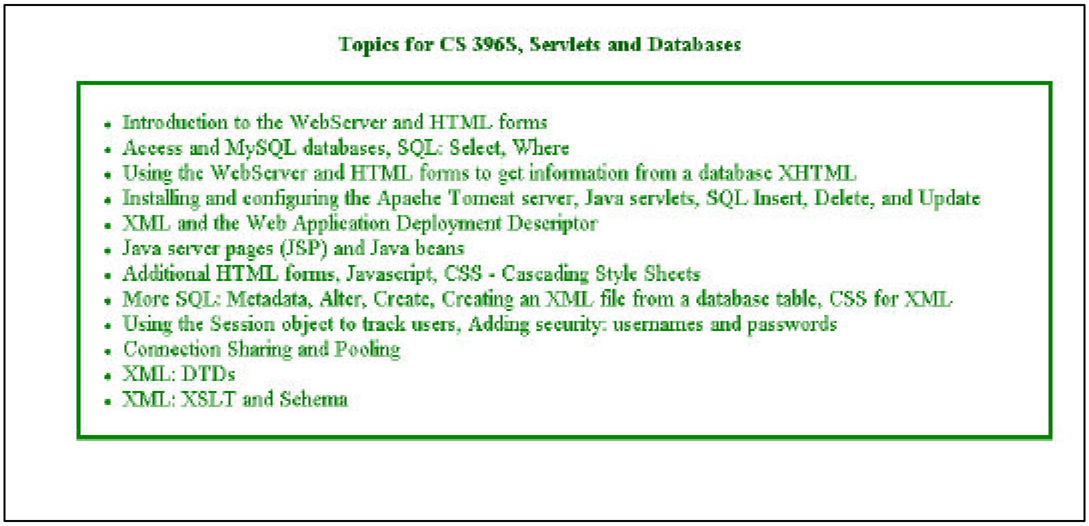
A more extensive example displays a list of topics for the Databases and Servlets course. It has a class attribute for centering a line. This may be useful in a number of places. It also has one for styles to apply to a list.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
<head>
        <title>CS 396S Topics</title>
        <style type: "text/css">
        .center {text-align: "center";}
        .topics
        {
                font-size: "medium";
                border-style: "solid";
                list-style-position: "inside";
                padding: "0.5cm";
        }
        </style>
</head>

<body>
        <h3 class="center">Topics for CS 396S, Servlets and Databases</h3>
        <ul class="topics">
                <li>Introduction to the WebServer and HTML forms</li>
                <li>Access and MySQL databases, SQL: Select, Where</li>
                <li>Using the WebServer and HTML forms to get information from a database XHTML</li>
                <li>Installing and configuring the Apache Tomcat server, Java servlets, SQL Insert, Delete, and
                Update</li>
                <li>XML and the Web Application Deployment Descriptor</li>
                <li>Java server pages (JSP) and Java beans</li>
                <li>Additional HTML forms, Javascript, CSS - Cascading Style Sheets</li>
                <li>More SQL: Metadata, Alter, Create, Creating an XML file from a database table, CSS for
                XML</li>
                <li>Using the Session object to track users, Adding security: usernames and passwords</li>
                <li>Connection Sharing and Pooling</li>
```

```
		<li>XML: DTDs</li>
		<li>XML: XSLT and Schema</li>
	</ul>
</body>
</html>
```

In the browser, this looks like the figure below.



Topics for CS 396S, Servlets and Databases

- Introduction to the WebServer and HTML forms
- Access and MySQL databases, SQL: Select, Where
- Using the WebServer and HTML forms to get information from a database XHTML
- Installing and configuring the Apache Tomcat server, Java servlets, SQL Insert, Delete, and Update
- XML and the Web Application Deployment Descriptor
- Java server pages (JSP) and Java beans
- Additional HTML forms, Javascript, CSS - Cascading Style Sheets
- More SQL: Metadata, Alter, Create, Creating an XML file from a database table, CSS for XML
- Using the Session object to track users, Adding security: usernames and passwords
- Connection Sharing and Pooling
- XML: DTDs
- XML: XSLT and Schema

There is much more that you can do with Cascading Style Sheets. See http://www.w3schools.com for more information about many of the possible styles.