

Web Design continued

Web Design Part 4

We will complete our web design unit by:

1. Creating a second web page that uses tags for tables and lists.
2. Creating an external style sheet for the second web page.
3. Creating at least a third web page and making some links between the pages so they function together as multi-page web site.

The detailed steps are as follows:

1. Create a second web page that can be related to your first web page (without losing the original page) and eventually be part of a web site of at least three pages. To create your new web page, use Notepad as before, and save the page in a new filename (again ending in `.html` and containing no spaces). Your page will have the same structure as before but with different content between the opening and closing tags for `title` and `body`. (I don't care if what you communicate is very similar to your first page as long as you use the new layout features described below.) As before, remember that the title appears in the upper bar of the browser window, and you may want to repeat it in the body if you want it to display in the main part of the page.
2. In your second web page, include at least one table. Like everything in HTML (the language of web pages), a table begins with an opening and closing tag (`<table>` and `</table>` in the most basic form). Within a table, one can place one or more rows of content; each row begins with `<tr>` (for "table row") and ends with `</tr>`. Within a row, one can place several cells of table data, each beginning with `<td>` and ending with `</td>`. One also can choose to instead surround a cell with `<th>` and `</th>`, which generally makes it appear bold and centered so as to constitute a heading for a column or row. Below is a full example of a table, which also incorporates an optional attribute into the opening `<table>` tag to display a border of one pixel around the table:

```
<table border="1">
<tr>
<th>Header 1</th>
<th>Header 2</th>
</tr>
<tr>
<td>row 1, cell 1</td>
<td>row 1, cell 2</td>
</tr>
<tr>
<td>row 2, cell 1</td>
<td>row 2, cell 2</td>
</tr>
</table>
```

Make your own table. If you want to see more examples and variations, you can look at http://www.w3schools.com/html/html_tables.asp.

3. Now you should include at least two lists, but unlike the approach used in your first web page, you will use HTML markup specifically designed for lists instead of just inserting line breaks. There are two kinds of lists: unordered and ordered. An unordered list could look like this:

```

<ul>
  <li>cat food</li>
  <li>bread</li>
  <li>milk</li>
  <li>veggies</li>
  <li>paper towels</li>
</ul>

```

and would generally display with a “bullet” in front of each list item. An ordered list would be created the same way but with `` `` instead of `` ``. Note that the ordered list will display with numbers (or you can choose a different enumeration style) in front of each list item. Include at least one unordered list and at least one ordered list in your page. For extra credit, nest one list inside of another.

- For this web page, you are to provide styling with an external style sheet (like the internal style sheet you did before, but in a separate file). Instead of using a section beginning `<style>` and ending `</style>`, you will include a line just like this in the head of the page (again before or after the entire title element):

```
<link href="style.css" rel="stylesheet" type="text/css">
```

The only thing you might change in the line above is the name of the external style sheet, `style.css`. Whatever name you use, create a file of that name in the same folder as everything else you have been working on; then put your style sheet into that file.

- One last feature you might want in your style sheet is to use `id` and `class` attributes. Actually, it is both the `.html` file and the style sheet that you will need to modify to work with `id` and `class`. The idea of both of these things is that you can mark certain elements in your `.html` file as having a particular `id` or `class`; then the style sheet can provide styling specifications that apply just to elements that have the indicated `id` or `class`. An `id` is usually used for a single unique element, while a `class` may be applied to a number of different elements. An example of HTML using each is as follows:

```
<p id="special-par">This is a special paragraph that is going to have
a styling specification that applies just to this particular
paragraph.</p>
```

```
<p>And here is a table:
```

```
<table>
  <tr class="odd"><td>first row</td>
  <tr class="even"><td>second row</td>
  <tr class="odd"><td>third row</td>
  <tr class="even"><td>fourth row</td>
  <tr class="odd"><td>fifth row</td>
</table>
```

```
</p>
```

Now we can apply a special text color to the paragraph above and apply different background colors to the odd and even rows of the table by putting the following styling directives in the style sheet:

```
#special-par { color: red; }
.odd { background-color: yellow; }
.even { background-color: aqua; }
```

Note that in the style sheet, we use `#` before an `id` and `.` before a `class`.

- The last step in the web design unit is to make at least a third web page that uses the same external style sheet as the second web page. Make some links between your pages so that they function together as a multi-page web site. You can continue to work on your web pages to make the coolest possible site; time permitting, we will look at each other’s sites.
- Totally optional is to put your web site in a publicly accessible place. If you want to do this, ask your parents for permission and search for “free web host” in a search engine or try one of the top-rated choices at <http://www.free-webhosts.com/webhosting-01.php> . Then following the instructions there to register for an account and upload to the web host.