MORE HTML

Overview

- Recap some basic HTML
  - Links
  - Style sheets
- Dynamic web pages
  - Image maps
  - Forms
  - Javascript

A web page with simple links

- How can we create this page?

Here's the HTML for that:

```
<html>
<head>
<title>Some links</title>
</head>
<body>
<hr>
This is an example of using links to make it easy for a
user to go from one web page to another:
<p>
My favorite colors are:
<ul>
<li> <a href="blue.html">blue</a>
<li> <a href="purple.html">purple</a>
<li> <a href="grey.html">grey</a>
</ul>
<hr>
</body>
</html>
```

(you can also find this on the course web-page)

Style sheets

- Two ways of using style sheets.
  1. "Inline"
     - Include the style information in the HTML file
  2. "By reference"
     - In a separate file
- The same information, just formatted in a different way.

- "inline" setting text color and background:

```
<style type="text/css">
  body { color: black; background: white; }
</style>
```

- Using a separate style sheet ("style file")
  - In the html file

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

  - In the "style file":

```
/* style.css - a simple style sheet */
body {
  color: black; background: white;
}
```

Other style sheet things

- Margins, left and right indents:

```
body { margin-left: 10%;
          margin-right: 10%; }
h1 { margin-left: -8%;}
h2,h3,h4,h5,h6 { margin-left: -4%; }
p { text-indent: 2em; margin-top: 0;
                  margin-bottom: 0; }
```

- white space above and below:

```
h2 { margin-top: 8em; margin-bottom: 3em; }
```

- Fonts:
  - Styles:

```
em { font-style: italic;
             font-weight: bold; }
strong { text-transform: uppercase;
             font-weight: bold; }
```

  - text-transform can be: uppercase, lowercase
  - Can select font families: Verdana, Garamond, "Times New Roman", sans-serif,

```
body { font-family: Verdana, sans-serif; }
```

- Link colors:

```
:link { color: rgb(0, 0, 153) }
:visited { color: rgb(153, 0, 153) }
a:active { color: rgb(255, 0, 102) }
a:hover { color: rgb(0, 96, 255) }
```

- `rgb(0, 96, 255)` defines the intensity of the red, green and blue light in the `color`

Divisions

- Way of structuring pages
  - Way to specify borders, colors etc
- Name the divisions using “class”

- In the HTML file use:

```
<div class="box">
The content within this DIV element
will be enclosed in a box with a thin
line around it.
</div>
```

- In the style sheet, use:

```
div.box { border: solid;
          border-width: thin;
                 width: 100% }
```

Making things dynamic

- Giving the user control to navigate pages
  - Links
  - Image maps
- Giving the user control to change page content
  - Forms
  - Javascript

An imagemap example

```
<html>
<head>
<title>Image map</title>
</head>
<body>
<hr>
This is an example of using an image map to make it easy for a
user to go from one web page to another
<p>
My favorite colors are:<br>
<img src="mymap.jpg" usemap="#mymap" height="100" width="300">
<map name="mymap">
<area shape="rect" coords="0,  0,100,100" href="blue.html">
<area shape="rect" coords="100,0,200,100" href="purple.html">
<area shape="rect" coords="200,0,300,100" href="grey.html">
</map>
<hr>
</body>
</html>
```

- The background to the map is the image `mymap.jpg`
- The origin is in the upper left corner.
- The numbers in `rect` then specify:
  - left x coordinate
  - top y coordinate
  - right x coordinate
  - bottom y coordinate
- All in pixels
- Similarly for:
  - circle (center-x, center-y radius)
  - polygon (x1, y1, x2, y2, xn, yn)

Forms

- HTML forms provide a way to create *dynamic* web pages
  - The user has control over what appears
- Forms are linked with some type of *program* that runs either on the *server-side* or the *client-side*.
- We will learn a little bit of Javascript, which runs on the *client* side.

- A form starts with the *form* tag
  - Appears in the body of an HTML file.

```
<form name="form1">
.
.
.
</form>
```

- A form contains one or more *input* tags
  - Allow the user to enter information:

```
<input type="button"
       name="mybutton"
       value="click me"
       onclick="alert('hello')" />

<input type="text"
       name="myname" />
```

Javascript

- Javascript is another computer programming language.
  - Other than HTML
- Tells the computer how to *do* things, rather than how to *display* things.
  - Still needs the browser to run.
- Javascript is *not* the same as Java.
- Javascript *functions* handle user input, when user clicks on a `button`
  - n the example above, the javascript function `alert()` is "called", or "run" or "invoked"

The first form example

```
<html>
<head>
<title>
Simple form
</title>
</head>
<body>
<p>
<form name="form1">
<input type="button"
name="mybutton"
value="press me"
onclick="alert('hello are we done yet?')" />
</form>
</body>
</html>
```

The second form example

```
<html>
<head>
<title>
Another form
</title>
</head>
<body>
<form name="form1">

Please enter your name:

<input type="text"
name="myname" />

<input type="button"
name="mybutton"
value="click me"
onclick="alert('hello '+form1.myname.value)" />
</form>
</body>
</html>
```

Summary

- This lecture recapped some of the HTML we covered before:
  - Links
  - Style sheets
  - Image maps
- It also covered some new topics:
  - Forms
  - Javascript
- We will use these things in the lab on Wednesday