

today: advanced HTML

- images, tables
- style sheets
- making things dynamic
 - image maps
 - forms
 - javascript

resources:

- on-line tutorials:
 - <http://www.w3.org/MarkUp/Guide/Advanced.html>
 - <http://www.w3.org/MarkUp/Guide/Style.html>
- reading: Reed chapter 2

images

- the *image* tag allows you to display an image within a web page
- example:

```

```

- you can specify properties of the image:

- width, height
- border
- alignment

```
<p> This text will be
flowed around the right side of the graphic.</p>
```

```
<p> This text will be
flowed around the left side of the graphic.</p>
```

```
<p> This text will be
flowed around the right side of the graphic.<br clear="all">
This starts a new line below the floated image.</p>
```

- you can include images in tables (see next topic)

tables

- begin a table with `<table>`
- end a table with `</table>`
- begin each row with `<tr>` and end each row with `</tr>`
- begin each column with `<td>` and end each column with `</td>`
- options:
 - borders
 - *cellpadding* (padding within a cell)
 - *cellspacing* (spacing between cells)
 - width and height (in pixels)
- tricks:
 - empty cells — use ` `;
 - *spanning* multiple rows or columns
- coloring cells:

```
<td bgcolor="red">ASDF</td>
```

- aligning cell content:
 - horizontally: left, center, right
 - vertically: top, middle, bottom
- advanced formatting:
 - cell padding (extra space inside the cells)
 - cell spacing (space between the cells)
 - width of table, of cells
 - alignment of cell content
 - empty cells ()
 - multi-row and multi-column cells
 - borders
 - coloring cells
 - headings

style sheets

- text color and background color:


```
<style type="text/css">
  body { color: black; background: white; }
</style>
```
- linking to a separate style sheet (in another file):


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

and the "style file" looks like this:

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

- 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
 - font families: Verdana, Garamond, "Times New Roman", sans-serif, e.g.


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

- divisions:
 - borders
 - colors
 - name the divisions using "class"
 - example:

in the style file:

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

in the html file:

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


```
:link { color: rgb(0, 0, 153) } /* for unvisited links */
:visited { color: rgb(153, 0, 153) } /* for visited links */
```

```
a:active { color: rgb(255, 0, 102) } /* when link is clicked */  
a:hover { color: rgb(0, 96, 255) } /* when mouse is over link */
```

making things dynamic

- giving the user control to navigate pages
 - links
 - image maps
- giving the user control to change page content
 - forms
 - javascript

image maps

- example:

```
  
<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>
```

- origin is upper left corner (pixel)
- rect: left-x, top-y, right-x, bottom-y
- circle: center-x, center-y, radius
- poly: x1,y1, x2,y2, ... xn,yn

forms and javascript

- overview
 - HTML forms provide a way to create *dynamic* web pages—
i.e., 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
- form tag:
 - appears in the body of an HTML file

```
<form name="form1">  
.  
.  
.  
</form>
```
- input tag:
 - contains **input** components that allow the user to enter information

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

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

- **javascript** *functions* handle user input, when user clicks on a button
 - in the example above, the javascript function `alert()` is "called", or "run" or "invoked"