Lab 2

Lists

Objective

After studying this lab student should be able to:

1. Know different types of lists
2. Know how to create list items

1. Types of Lists

There are 3 different types of list elements

1. … specifies an ordered list (using numbers or letters to label each list item)
    a. identifies each list item
    b. can set type of ordering, start index
2. … specifies unordered list (using a bullet for each)
    a. identifies each list item
3. <dl>…</dl> specifies a definition list <dt> identifies each term
    a. <dd> identifies its definition

2. UNORDERED LISTS

<ul>, **<li>** Make bulleted lists.

Try The Following code

```
<html>
<head>
<title>Unordered Lists</title>
</head>

<body>
<ul>
<li>The first item in the list.</li>
<li>The second item in the list.
   This line is still part
   of the second item. </li>
<li>The third item in the list. </li>
</ul>

<ul>
<li>text</li>
<li>text</li>
<li>text</li>
</ul>
<ul type="disc">
<ul type="circle">
<ul type="square">

</body>
```

```
</html>
```

Use **
** ... **
** tag for each item in the list.

Try This
Make a list of lists. That is, make a list where each list item is a list.

3. ORDERED LISTS

<ol>, **<li>** Make numbered lists.

Try The Following code

```
<html>
<head>
<title>Ordered Lists</title>
</head>
<body>
<ol>
<li>The first item in the list.</li>
<li>The second item in the list.
   This line is still part
   of the second item.</li>
<li>The third item in the list.</li>
</ol>

<ol>
<li>text</li>
<li>text</li>
<li>text</li>
</ol>
<ol start="5">
<ol type="A">
<ol type="a">
<ol type="I">
<ol type="i">
<ol type="1">
<ol type="I" start="7">

</body>
</html>
```

Use **
** ... **
** tag for each item in the list.

Try This

- Make an ordered list that contains unordered lists as list items.
- Try different combinations of list of lists to see how they nest.

4. DEFINITION LISTS

<dl>, **<dt>**, **<dd>** Make a list of terms and definitions.

```
<html>
<head>
<title>Definition Lists</title>
</head>
<body>
<dl>

<dt>The first term.</dt>
<dd>The definition of the first term
in the list.</dd>

<dt>The second term.</dt>
<dd>The definition of the second
   term in the list.
   This definition is a big one
   that might span a
   couple of lines.</dd>
<dd>The second term has a two part
   definition.</dd>

<dt>The third term.</dt>
<dd>
The definition of the third term
in the list.
</dd>

</dl>
</body>
</html>
```

Each item in a **definiton list**, **<dl>**, is made up of a **definition term**, **<dt>**, followed by zero or more **definition descriptions**, **<dd>**.

Lab 3

Using Graphics and Links

Objective

After studying this lab student should be able to:

1. Display Images on a Web Page
2. Add hyperlinks to a Page
3. Create Email Links
4. Create Anchors on a Page
5. Working with Client-Side Image Maps

1. LINKS TO DIFFERENT PAGES

<a>, **href** Set up an **anchor** with a **hyperlink**.

Try this code
<html> **<head>** **<title>A Sample Link** **</title>** **</head>** **<body>** **Here is** **a link** **to a test page.** **</body>** **</html>**

Click the link in the right frame. The link is probably blue and underlined, but it does not have to be. The mouse cursor should change to a hand with a pointing finger.

Notes

- Use this kind of link to jump to any page on the WWW.
- Clicking the link brings the page from the URL listed after the **href** attribute to your browser.
- The URL to another Web page usually starts with http://, but it can be any protocol understood by your browser, such as ftp://, file://, mailto: and others. It can also be a bare file name of a Web page, an image file or a plain text file.

Try This

Make a link to a Web page that is in the same folder. It should have the form

<a href="yourfile.html">

2. LINKS TO ANCHORS ON THE SAME PAGE

<a>, **name** Making a named anchor.

Try this code
<html> **<head>** **<title>Named Anchors</title>** **</head>** **<body>** **Jump down.** ** ** **Testing ** **One ** **Two ** **Three** ** ** ** ** **Testing ** **One ** **Two ** **Three** ** ** ** ** **Testing ** **One ** **Two ** **Three** ** ** ** ** **Testing ** **One ** **Two ** **Three** ** ** **You jumped here!******* ** ** **Testing ** **One **

```
Two<br>
Three
</p>

<p>
Testing<br>
One<br>
Two<br>
Three
</p>

<p>
Testing<br>
One<br>
Two<br>
Three
</p>

</body>
</html>
```

Click the link in the right frame. Click the output button, above right, to restore the right side.

Notes

- Most of text in this sample is just to fill up the screen so there is someplace to jump.
- There are two anchors in this example. The anchor near the top is a hyperlink, or link, to the named anchor near the bottom.
- Note that the sharp sign, #, should only appear with the **href** attribute. It means the link is to a *named* spot in the current document. A spot is named using the **name** attribute in an anchor.
- You can use as many **name** attributes as you like in a Web page, but each name has to be different. You can reuse names in other Web pages. You can use the name in many **href**'s.
- Use named anchors to make a 'live' table of contents for a Web page.

Try This

Rewrite the previous so that you can jump back and forth between *Jump down* and *You jumped here.*

3. LINKS TO ANCHORS ON DIFFERENT PAGES

<a>, **href** Linking to named places on other pages.

Try this code
<html> <head> <title>A Sample Link</title> </head> <body> Jump to the anchor on a different page. </body> </html>

Sets up a link to a named place in a different page. Click the link in the right frame. Note that when the page loads it is not at the top.

Notes

- The sharp sign, #, following the URL means the link is to a *named* spot in some other document. The **name** attribute had to be used in the other document. Usually you can only use this feature with your own pages.

4. LINKS: MAILTO

<a>, **href:** making an email link.

Try this code
**<html> <head> <title>A mailto Link </title> </head> <body> **

```
Comments? Email  of Ahmed at
<a href="mailto:ahmed@yahoo.com">Ahmed@yahoo.com</a>

</body>
</html>
```

Click the link in the right frame. Your email program should run.

Notes

- There are no slashes, //, in a **mailto** URL.
- A **mailto** link runs an email program.
- Since not all browsers support this feature, it's probably a good idea to use the email address as the link text, as in the sample. You can use any other text you want.

5. INLINE IMAGES

<img>, **src** Displays images mixed in with text.

Try this code
<html> **<head>** **<title>Inline Images</title>** **</head>** **<body>** ** ** **This is a inline image.** **Notice it appears** **<img src="anypicturename.gif" width="59" height="42" alt="** 88 **">** **right in with your text.** ** ** **</body>** **</html>**

Inline images display right inline with your text. They act like big words. Note the big space between the lines of text. Inline images do not start new line or paragraphs

Notes

- The **src** attribute is required. If should be the file name of a gif, jpg, or, for newer browsers, png image file.
- Use **width** and **height** even though they're not required. This information allows your browser to lay out the page before the images start downloading and then fill in the images as they arrive. Your pages will load faster and more smoothly.
- You can get the width and height of your images from a graphics program. Some HTML editors insert them automatically.
- Don't use **width** and **height** to change the size of an image unless you're doing it for effect. Doing so distorts the image. Use a graphics program.
- The **alt** attribute should contain some descriptive text to be displayed if the image can't.
- Setting **border** to zero removes any border that might appear around an image.

6. INLINE IMAGES AS LINKS

<img>, **border** Outlining images.

Try This Code

```
<html>
<head>
<title>Inline Images as Links</title>
</head>
<body>
<p>
<a href="http://www.ms88.com/testpage.html">
<img src="htmle16.gif" width="174" height="34">
a link.
</a>
<p>
<a href="http://www.ms88.com/testpage.html">
<img src="htmle16.gif" border="0" width="174" height="34">
a link.
</a>
<p>
<img src="htmle16.gif" border="3" width="174" height="34">
not a link.
</body>
</html>
```

Click on each of images in the right frame. The first two are links to a test page. Click the Output in the upper right frame to restore the original output. Use an image that you have **on your computer** to get the sample to work correctly.

Notes

- The browser highlights the image by drawing a blue border around it. Once you click the link, the border changes to purple. (These are usually the colors; browsers and Web page authors can change them.) You can include some text with the image.
- To turn off the border set the **border** attribute to zero.
- The third image in the sample is not a link and has its **border** attribute set to three. If you're using Netscape Navigator or newer versions of Internet Explorer, you see a

three pixel wide border around the image. Other browsers may ignore **border** outside of a link.

Try This

Change **border** to different values to see what happens.

7. INLINE IMAGES 2

<img>, **align**, **top**, **middle**, **bottom** Changing how the text and images line up.

Try This Code

```
<html>
<head>
<title>Inline Images 2</title>
</head>
<body>

<p>
<img src="image.gif" align="top" width="59" height="42" alt="">
ONCE upon a time there
lived a rich man, who had a wife,
and one daughter, a very sweet and pretty
girl. The wife fell sick and died
</p>
<p>
<img src="image.gif" align="middle" width="59" height="42" alt="">
ONCE upon a time there
lived a rich man, who had a wife,
and one daughter, a very sweet and pretty
girl. The wife fell sick and died
</p>

<p>
<img src="image.gif" align="bottom" width="59" height="42" alt="">
ONCE upon a time there
lived a rich man, who had a wife,
and one daughter, a very sweet and pretty
girl. The wife fell sick and died
</p>
</body>

</html>
```

The blue rectangle takes up the entire image.

Notes

- If text wraps around it moves to the line below the image.
- Compare the position of the text with the image for each of **top**, **middle** and **bottom**.
- The default alignment is **bottom**.

- **top**, **middle** and **bottom** are useful for small images that you want mixed in with your text. See the next topic for flowing text around larger images.

8. INLINE IMAGES 5

Faking a centered floating image.

Try This Code

```
<html>
<head>
<title>Inline Images 5</title>
</head>
<body>

<p align="center">
<img src="image.gif" align="left" width="40" height="155" alt="">
<img src="image.gif" align="right" width="40" height="155" alt="">
<img src="image.gif" width="40" height="30" alt="">
Title<br>A truly thrilling story!
</p>
<p align="right">
The Hero
<br>The Heroine
</p>

<br clear="all">
<p>
The amazing story of ...
</p>

</body>
</html>
```

The three images keep their positions (left - center - right) when the window is resized.

Notes

- The first two images use **left** and **right**. The third is a regular inline image. **The order of the image tags matters.**
- **<p align="center">** centers the third image and the text, but has no effect on the floating images.
- **<p align="right">** right justifies **The Hero** and **The Heroine**. Note the position of the **
**. It matters for some browsers.

- **<br clear="all">** breaks out of the text wrap that's occurring between the two floating images. In this sample, **left** or **right** would work just as well because the images are the same height.

Lab 4

Tables

Objective

After studying this lab student should be able to:

1. Add Tables to a Page
2. Work with <table>, <tr>, <td>, <th> and <caption>Elements
3. know Table Attributes
4. Create Nested Tables

1. TABLES

<table>, **<tr>**, **<td>** A simple table with three rows and two columns.

Try This Code

```
<html>
<head>
<title>Tables 1</title>
</head>
<body>

<table border>
 <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>
 <tr>
  <td>Row 3, cell 1</td>
  <td>Row 3, cell 2</td>
 </tr>
</table>

</body>
</html>
```

The **border** attribute makes the outline of the table visible. It's not required. Each row has the same number of columns.

Notes

- Plan your table. At first you might want to draw it on paper.
- An HTML table is made of rows that contain cells. The important idea is that the cells are set up in the order they appear in your HTML.
- To Set Up a Table
    - Use a table tag to create the table.

<TABLE>

</TABLE>

- For each row include a table row tag. Here's a table with two rows.

```
<TABLE>
 <TR>

 </TR>
 <TR>

 </TR>
</TABLE>
```

- For each cell include a table data tag.

```
<TABLE>
 <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>
```

This table has two rows. Each row has two cells. I arranged the tags to make it easier to keep track of the data. You can arrange them anyway that's helpful. (Of course, the *order* of the tags is important.)

Think about a table as growing from left to right and top to bottom. Cells are added to a row from left to right until the row ends, and so on until there are no more rows.

Try This

Set up tables with different numbers of rows and columns until you get the hang of it. Pay attention to how the size of the cells changes with the amount of data you put in them. Save these tables for experimenting with in the next couple of topics.

2. TABLES SPANNING COLUMNS

<td>, **colspan** Spanning columns.

Try This Code
<html> **<head>** **<title>Tables - colspan</title>** **</head>** **<body>**

```
<table border>
 <tr>
  <td colspan="2">Row 1, cell 1 + 2 spanned</td>
  <td>Row 1, cell 3</td>
 </tr>
 <tr>
  <td>Row 2, cell 1</td>
  <td>Row 2, cell 2</td>
  <td>Row 2, cell 3</td>
 </tr>
 <tr>
  <td>Row 3, cell 1</td>
  <td>Row 3, cell 2</td>
  <td>Row 3, cell 3</td>
 </tr>
</table>

</body>
</html>
```

In the table on the right, the first row has one fewer column than the others.

Notes

- To span columns add a **colspan** attribute to a table data tag.
- You can span columns anyplace in a table.
- You have to pay attention the number of cells in the row when spanning columns. To span two columns you use one fewer **<td>** in that row, to span three columns you use two fewer **<td>**'s in that row, and so on.

Try This

Set up tables with different numbers of rows and columns (or use the ones from the previous topic) and practice spanning columns until you get the hang of it.

3. TABLES SPANNING ROWS

<td>, **rowspan** Spanning rows.

Try This Code

```
<html>
<head>
<title>Tables - rowspan</title>
</head>
<body>
```

```
<table border>
 <tr>
  <td rowspan="2">Row 1+2, cell 1<br> spanned</td>
  <td>Row 1, cell 2</td>
  <td>Row 1, cell 3</td>
 </tr>
 <tr>
  <td>Row 2, cell 2</td>
  <td>Row 2, cell 3</td>
 </tr>
 <tr>
  <td>Row 3, cell 1</td>
  <td>Row 3, cell 2</td>
  <td>Row 3, cell 3</td>
 </tr>
</table>

</body>
</html>
```

In the table on the right, the first column has one fewer row than the others.

Notes

- To span rows add a **rowspan** attribute to a table data tag.
- You can span rows any place in a table.
- You have to pay attention the number of **<td>**'s in the **<tr>**'s following the one with the spanned rows. To span two rows you use one fewer **<td>** in the next row; to span three rows you use one fewer **<td>** in *each* of the next two rows, and so on.

Try This

Set up tables with different numbers of rows and columns (or use the ones from the previous topics) and practice spanning rows until you get the hang of it.