UNIT 1 - HTML

Web Programming

- **Web development** is a broad term for the work involved in developing a web site for the Internet (World Wide Web) or an intranet (a private network).
- This can include web design, web content development, client link, client-side/server-side scripting, web server and network security configuration, and e-commerce development.
- “Web development" usually refers to writing markup and coding.

- Web development can range from developing the simplest static single page of plain text to the most complex web-based internet applications, electronic businesses, or social network services.
- All web programming is done with web programming languages.
- These languages can include static technologies or client side scripting languages like HTML, XHTML, CSS, JavaScript, and XML.
- Most web site programming is done using server-side web programming languages to create dynamic websites. This code runs on the server and then gives static information back to the web browser.
- The most popular web programming languages are: PHP, ASP.NET, Ruby on Rails, Perl, ASP classic, Python, and JSP.

HTML - Introduction

- HTML is Hypertext markup language and is not a programming language.
- The **HyperText Markup Language** (HTML) is an SGML application for marking up documents for inclusion in the World Wide Web.
- **Hyper Text:** Hyper Text simply means "Text within Text". A text has a link within it, is a hypertext.
- **Markup language:** A markup language is a programming language that is used make text more interactive and dynamic. It can turn a text into images, tables, links etc.

- **HTML allows you to:**
  - Publish documents to the Internet in a platform independent format
  - Create links to related works from your document
  - Include graphics and multimedia data with your document
  - Link to World Wide Web information resources on the Internet
- HTML comment is like this

  <!-- comment text -->

HTML Tags

- In HTML formatting is done using HTML tags.
- HTML tags are keywords (tag names) surrounded by **angle brackets** like <html>
- HTML tags normally **come in pairs** like and
- The first tag in a pair is the **start tag,** the second tag is the **end tag**
- The end tag is written like the start tag, with a **forward slash** before the tag name
- Start and end tags are also called **opening tags** and **closing tags**
- <tagname>content goes here...</tagname>

- Html tags are not case sensitive. i.e. p and P are same.
- HTML documents contain HTML tags and these documents are called web pages.
- HTML tags are interpreted by the browser itself. No need of any other interpreters for executing and displaying the result of HTML code
- The purpose of a web browser (Chrome, Internet Explorer, Firefox) is to read HTML documents and display them as web pages.
- The browser does not display the HTML tags, but uses the tags to interpret the content of the page:

```
<html>

<head>

<title>Page title</title>

</head>

<body>

<h1>This is a heading</h1>

<p>This is a paragraph.</p>

<p>This is another paragraph.</p>

</body>

</html>
```

Example 1

```
<!DOCTYPE html>
  <html>
      <head>
          <title>Example1</title>
      </head>
      <body>
          <h1>My First Heading</h1>
          <p>My first paragraph.</p>
      </body>
  </html>
```

Example1 – Explanation

- The DOCTYPE declaration defines the document type
- The text between <html> and </html> describes the web page
- The document head or head tag holds the control information to be used by browsers and servers. The head section is mainly used to
  - Insert Style Sheet & Scripting code like Java Script
  - insert title tag (most people use only title tag here).
  - Insert information about the document like metadata

- The text between <body> and </body> is the visible page content
- The text between <h1> and </h1> is displayed as a heading
- The text between and is displayed as a paragraph

HTML elements

- HTML documents are defined by HTML elements.
- Elements are the bits that make up web pages. i.e. everything between start and end tags are called elements.

Eg. '<title>' and '</title>' are *tags*, '<title>Testing</title>' is a title *element*.

- The **element content** is everything between the start and the end tag
- Empty elements are **closed in the start tag (
)**
- Most HTML elements can have **attributes**

HTML Attributes

- HTML tags or elements can contain attributes like <tag attribute="value">
- Attribute value should be in double quotes(" ")
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Commonly used attributes are :
  - class – specifies the class name of an element
  - id – specify unique id of an element
  - style – specifies the inline style of an element
  - title – specifies the title of an element which will be shown as tooltip

- Attributes usually come in name/value pairs like: **name="value"**
- The link address is specified in the href attribute.
- The filename of the image source is specified in the src attribute.
- Images in HTML have a set of size attributes, which specifies the width and height of the image.
- The alt attribute specifies an alternative text to be used, when an image cannot be displayed.
- The style attribute is used to specify the styling of an element, like color, font, size etc.
- A title attribute is added to the <p> element. The value of the title attribute will be displayed as a tooltip when you mouse over the paragraph.
- Double quotes around attribute values are the most common in HTML, but single quotes can also be used.
- In some situations, when the attribute value itself contains double quotes, it is necessary to use single quotes

The Document Body

- HTML documents are structured as blocks of text, each of which can be formatted independently.
- A block has no meaning outside the context of a particular document.
- The two major blocks of text in HTML documents are the paragraph and heading.
- Html processors ignore all white spaces except the spacing between words.

Different Tags in HTML

- <title> </title> - display the title of the web page
- <h1></h1>, <h2></h2>,<h3></h3> - display headings in different sizes. <h1>1 will display in large font size, <h2> in little more small size than h1 and so on.
- <p> - to display the contents in the paragraph form
-
 - newline or line break. It is a self closing tag as it does not have a separate end tag
-   - to specify a space.

- Headings are defined with the <h1> to <h6> tags.
- <h1> defines the most important heading. <h6> defines the least important heading.
- <h1>Heading 1</h1>
- <h2>Heading 2</h2>
- <h3>Heading 3</h3>
- <h4>Heading 4</h4>
- <h5>Heading 5</h5>
- <h6>Heading 6</h6>
- The HTML <pre> element defines preformatted text.
- The text inside a <pre> element is displayed in a fixed-width font (usually Courier), and it preserves both spaces and line breaks.

Escape sequences

- They are used to display characters that HTML uses as control sequences.
-   - to represent a single space ( )
- & - to represent an ampersand symbol (&)
- > - to represent greater than symbol (>)
- < - to represent less than symbol (<)
- © - to represent copyright symbol (©)
- " - to represent quotation marks(“)

Formatting tags

The text in HTML can be altered or formatted using different formatting tags in HTML.

- <b> - contents inside this tag will be displayed in bold
- <i> - to display the contents in italics form
- or it is used to specify a text as "important". Many browsers display as bold and as italics.
- <big> - to show in big size
- <small> - to show in small size
- <sub> - to specify a text as subscript
- <sup> - to specify a text as superscript

- HTML <mark> Element
- The HTML <mark> element defines marked or highlighted text
- The HTML <del> element defines deleted (removed) text

Font

- The tag specifies the font face, font size, and font color of text.
- Color attribute is used to specify color name
- Face attribute is used to specify the font family
- Size attribute is used specify the font size which will be a number.
- Eg : This is some text!

Hyperlinks - Anchor tag

- <a href> - anchor tag. HTML links (Hyperlinks) are defined using this tag.
- A hyperlink (or link) is a word, group of words, or image or video or graphical animations that you can click on, to jump to another document or to navigate through different web pages or websites.
- When you move the cursor over a link in a Web page, the arrow will turn into a little hand.
- The most important attribute of the <a> element is the href attribute, which indicates the link's destination.

- Hyperlinks are used by HTML to link documents together.
- The anchor tag has three sections
  - Opening tag with the address of the referenced document. href is used to mention the reference part which can be links to documents within or outside the website, links to web pages within or outside the current website.
  - A piece of text or image to display as link. The link text can be formatted using any of the text formatting options.
  - Closing tag

- Eg. click here - here "click here" text will be shown as a hyper link and when you click on this link it will get redirected to next.html web page in the same folder.
- If next.html page does not exist then error will be shown. href is the most important attribute of anchor tag which is used to specify the redirection page or destination page.

Important Attributes of anchor tag

- href – Specifies the URL of the page the link goes to
- name – Specifies the name of the anchor tag
- target - Specifies where to open the linked document.

- **The values of this target attribute are**

  - _blank - Opens the linked document in a new window or tab
  - _self - Opens the linked document in the same frame as it was clicked (this is default)
  - _parent - Opens the linked document in the parent frame
  - _top - Opens the linked document in the full body of the window
  - *framename* - Opens the linked document in a named frame

Relative and Absolute paths

- **Absolute path**
  - An absolute path refers to a file on the Internet using its full URL,

    E.g. http://www.depaul.edu.in/temp/index.php
  - Absolute paths are called that because they refer to the very specific location, including the domain name.
  - The absolute path to a web element is also often referred to as the URL.
  - The absolute path is mainly used with the domain to point to Web elements that are on another domain than your own.

- **For example**

  From depaul.edu.in website if I want to refer a page in colleges.com site, then I should use the absolute url as given below

Click here for colleges website

Relative Path

- A relative path assumes that the file is on the current server. E.g /temp/ index.php
- If we are in the homepage, then to relative link to contact page can be given as below

```
<a href="contact.html">Contact Us</a>
```

- To go to index.html in the products folder. The relative path should be like this

```
<a href="products/index.html">click for products</a>
```

- If we are in index.html in the products folder and in this page we want to give link to contact.html in the home page (i.e. in your momcards folder), use the relative path like this
- ** contact**

- ../ is used to go one folder up. To go two folder up use it twice. i.e ../../page name

Figure to refer examples in relative path

Linking specific location using anchor tag

- Anchor tags can be used to refer to a specific location in the same webpage or a specific location in a different web page.

- To refer a location in same web page
- Eg:

** some text here **

some html codes goes here

** click to go the para section**

- when you click on this link, the control will go to the referenced section with the given ID (ie. Here it will go to the starting of the paragraph)

To refer a location in different webpage

P1.html

 some text here

some html codes goes here

P2.html

 click - when you click on this link, the control will go to the referenced section with the given ID in p1.html web page.

Img Tag

- <img> - HTML images are defined using this tag.

  Eg.** - here flower.jpg image will be shown with height and width 100pixel. This flower.jpg image should be in the current directory.

- Important attributes of img tag are
  - **src**- specifies the url of the image
  - **width** - to specify the width of an image
  - **height -** to specify the width of an image
  - **alt** - alternate text if the image does not exist.
  - **title** - to specify the tooltip of each element.
  - **usemap** - Specifies an image as a client-side image-map

- To add a background image on an HTML element, use the CSS property background-image.
- <body style="background-image:url('clouds.jpg')">
- To add a background image on a paragraph, specify the background-image property on the P element.
-

...

Image maps

- With image maps, you can add clickable areas on an image.
- The <map> tag defines an image-map.
- The usemap value starts with a hash tag # followed by the name of the image map, and is used to create a relationship between the image and the image map.
- The <map> element is used to create an image map, and is linked to the image by using the name attribute.
- The name attribute must have the same value as the usemap attribute.

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap">

<map name="workmap">
    <area shape="rect" coords="34,44,270,350" alt="
        Computer" href="computer.htm">
    <area shape="rect" coords="290,172,333,250" al
        t="Phone" href="phone.htm">
    <area shape="circle" coords="337,300,44" alt="C
        offee" href="coffee.htm">
</map>
```

- Then add the clickable areas. A clickable area is defined using an <area> element.
- You must define the shape of the area, and you can choose one of these values:
  - rect - defines a rectangular region
  - circle - defines a circular region
  - poly - defines a polygonal region
  - default - defines the entire region
- You must define some coordinates to be able to place the clickable area onto the image.
- The coordinates come in pairs, one for the x-axis and one for the y-axis.
- The coordinates 34, 44 is located 34 pixels from the left margin and 44 pixels from the top.
- The coordinates 270, 350 is located 270 pixels from the left margin and 350 pixels from the top.

LISTS

- Lists are mainly used to structure a website or its contents.
- Lists provide a piece of information or a direct index to the site
- Lists may contain −
  - **<ul>** − An unordered list. This will list items using plain bullets.
  - **<ol>** − An ordered list. This will use different schemes of numbers to list your items.
  - **<dl>** − A definition list. This arranges your items in the same way as they are arranged in a dictionary.

Ordered List

**<ol [type="1"|"a"|"A"|"I"|"i"] [start="n"] [compact]> **

- An ordered list starts with the <ol> tag.
- Each list item starts with the <li> tag.
- The list items are marked with numbers.
- There are different types of ordered lists which can be specified using the '**type**' attribute of <ol> tag.
- The starting value of the ol is specified using the **start** attribute.
- **Compact** attribute is used minimize the amount of space that a list uses
- By default ordered list items are displayed with numbers.

Example of types of ordered lists

```
<html>
<body>
<h4>Numbered list:</h4>
<ol>
   <li>Apples</li>
   <li>Bananas</li>
</ol>

<h4>Letters list:</h4>
<ol type="A">
   <li>Apples</li>
   <li>Bananas</li>
</ol>
```

```
<h4>Lowercase letters list:</h4>
<ol  type="a">
    <li>Apples</li>
    <li>Bananas</li>
</ol>
<h4>Roman numbers list:</h4>
<ol type="I">
     <li>Apples</li>
    <li>Bananas</li>
</ol>
<h4>Lowercase Roman numbers list:</h4>
<ol type="i">
    <li>Apples</li>
    <li>Bananas</li>
</ol>
</body>
</html>
```

Output

Numbered list:

1. Apples
2. Bananas

Letters list:

A. Apples

B. Bananas

Lowercase letters list:

a. Apples

b. Bananas

Roman numbers list:

I. Apples

II. Bananas

Lowercase Roman numbers list:

i. Apples

ii. Bananas

unordered list

- An unordered list starts with the <ul> tag. Each list item starts with the <li> tag.
- The list items are marked with bullets
- There are different types of ordered lists which can be specified using the 'type' attribute of <ul> tag.
- Compact attribute is used minimize the amount of space that a list uses
- By default unordered lists items are displayed with bullets.

Example for different types of unordered lists

```
<html><body>
<h4>Disc bullets list:</h4>
<ul type="disc">
 <li>Apples</li>
 <li>Bananas</li>
</ul>
<h4>Circle bullets list:</h4>
<ul type="circle">
 <li>Apples</li>
 <li>Bananas</li>
</ul>
<h4>Square bullets list:</h4>
<ul type="square">
 <li>Apples</li>
 <li>Bananas</li>
</ul>
</body></html>
```

Output

Disc bullets list:

- Apples
- Bananas

Circle bullets list:

- Apples
- Bananas

Square bullets list:

- Apples
- Bananas

Definition lists

<dl [compact]>…</dl>

- They do not use list items. The elements in this list are either items being defined or their definitions
- <dt>……[</dt>] - They are Definition terms which mark items whose definition will be provided by next data definition. They can be formatted using any regular text formatting and the closing tag is optional as it is assumed once a <dd> tag is reached
- <dd>…..[</dd>] – definition of terms are enclosed within these tags. It can include any test or block formatting elements

Example for definition lists

```
<html>
   <body>
      <dl>
         <dt>Widget</dt>
                  <dd>some text</dd>
         <dt>Second</dt>
                  <dd>second Text</dd>
      </dl>
   </body>
</html>
```

Output Definition list example

Widget

some text

Second

second Text

Marquee

- The HTML <marquee> tag is used for scrolling piece of text or image displayed either horizontally across or vertically down your web site page depending on the settings.

Eg : <marquee> Text to Scroll </marquee>

or

<marquee></marquee>

No	Attribute & Description
1	**width** This specifies the width of the marquee. This can be a value like 10 or 20% etc.
2	**height** This specifies the height of the marquee. This can be a value like 10 or 20% etc.
3	**direction** This specifies the direction in which marquee should scroll. This can be a value like *up, down, left* or *right*.
4	**behavior** This specifies the type of scrolling of the marquee. This can have a value like *scroll, slide* and *alternate*.
5	**scrolldelay** This specifies how long to delay between each jump. This will have a value like 10 etc.
6	**scrollamount** This specifies the speed of marquee text. This can have a value like 10 etc.
7	**loop** This specifies how many times to loop. The default value is INFINITE, which means that the marquee loops endlessly.
8	**bgcolor** This specifies background color in terms of color name or color hex value.
9	**hspace** This specifies horizontal space around the marquee. This can be a value like 10 or 20% etc.
10	**vspace** This specifies vertical space around the marquee. This can be a value like 10 or 20% etc.

Frame

- The <frameset> tag defines a frameset which is a special type of webpage.
- The <frameset> element holds one or more <frame> elements. Each <frame> element can hold a separate document or webpage
- The frameset page contains a set of references to HTML files, each of which is displayed inside a separate frame.
- The <frameset> element specifies HOW MANY columns or rows there will be in the frameset, and HOW MUCH percentage/pixels of space will occupy each of them.

- <frameset [cols="n%,n%"] [rows="n%,n%"]> … </frameset>

- <frame [name="name"] src="filename" [scrolling="yes"|"auto"|"no"] [frameborder="0"|"1"]>

```
<frameset cols="25%,*,25%">
    <frame src="frame_a.html" />
    <frame src="frame_b.html" />
    <frame src="frame_c.html“ />
</frameset>
```

Iframes (inline frames)

- The HTML iframe element is used by the browser to access documents which are addressable through URLs and then display those documents in a single page.
- URLs can be used to identify many types of resource from images to text pages. These resources can be either static or dynamic contents.
- Iframe has similar purpose to the object element

- <iframe [name="string"] src="URL" [scrolling="yes"|"auto"|"no"] [frameborder="1"|"0"] [height="nn"] [width="nn"] [marginwidth="nn"] [marginheight="nn"]>… </iframe>
- Iframe has many attributes same as non-inlined frame or simple frame.
  - name – name of iframe
  - Src="url of the resource that should come inside this iframe
  - Scrolling – whether to show the scrollbars or not
  - Frameborder – 1 value displays border for the frame and 0 ignores the frame border
  - Height and width – represents the height and width of the iframe
  - Margin width– represents left & right margins of the iframe
  - Margin height – represents top & bottom margins

Iframe - example

```
<html>
   <head>
      <title>Iframe Example Program</title>
   </head>
   <body>
      <iframe src="content.html" scrolling="auto"
      frameborder="1" marginwidth="10“
      marginheight="10" width="60%"
      height="30%"></iframe>
      <p>djfhdifdofid dfjdfi difdop</p>
   </body>
</html>
```

- In the above example
  - The main page is a simple HTML page
  - The iframe page and its contents will be displayed on the top of this html page with given attribute values

Multimedia Objects

- On the web multimedia generally means sound and image data and in other words we can define multimedia as any data which is not plain text or images.
- HTML has an object element which is used to embed multimedia objects directly into the page.
- <object classid="URL" data="URL" [codebase="URL"] type="string" [standby="string"] height="n" width="n" [title="string"]> ….. </object>

- The <object> tag defines an embedded object within an HTML document. Use this element to embed multimedia (like audio, video, Java applets, ActiveX, PDF, and Flash) in your web pages.
- You can also use the <object> tag to embed another webpage into your HTML document.
- At least one of the "data" or "type" attribute MUST be defined.
- An <object> element must appear inside the <body> element. The text between the <object> and </object> is an alternate text, for browsers that do not support this tag
- You can use the <param> tag to pass parameters to plugins that have been embedded with the <object> tag.

- Important attributes of object tag are:
  - Classid – mandatory field which identifies the URL of the object
  - Codebase – optional field which identifies the directory which contains the object
  - Type – mandatory field which is used to specify the MIME type of the object and is used by the browser
  - Height & width – the browser can use the height and width values to allocate screen space
  - Standby – optional field which is used to specify the alternative text while the object is being downloaded from the server

- When an object needs command-line parameters these can be passed in through the param tag
- <param name="string" value="string" type="string" valuetype=["ref"|"object"|"data"]>
  - name – represents the name that the object expects to receive
  - value – specifies the value that will be passed into the object
  - valuetype – is used to tell the browser the format of each parameter
  - type – the type of data is set using this field

```
<html>
   <head>
        <title>multimedia Example Program</title>
   </head>
   <body>
        <object width="400" height="400"
   data="dagobah_dmc.swf"></object>


   </body>
</html>
```

USING HTML TABLES

Defining a Table Structure

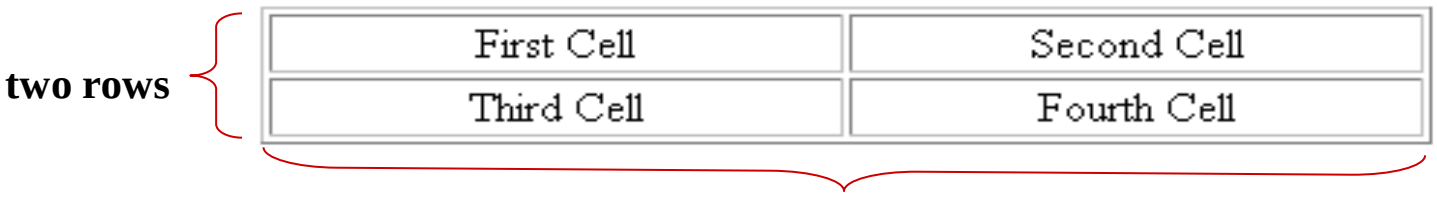
- The first step to creating a table is to specify the table structure:
  - the number of rows and columns
  - the location of column headings
  - the placement of a table caption
- Once the table structure is in place, you can start entering data into the table.

Using the <table>, <tr>, and <td> Tags

- Graphical tables are enclosed within a two-sided **<table>** tag that identifies the start and ending of the table structure.
- Each row of the table is indicated using a two-sided **<tr>** (for table row).
- Within each table row, a two-sided **<td>** (for table data) tag indicates the presence of individual table cells.

The General Table Syntax

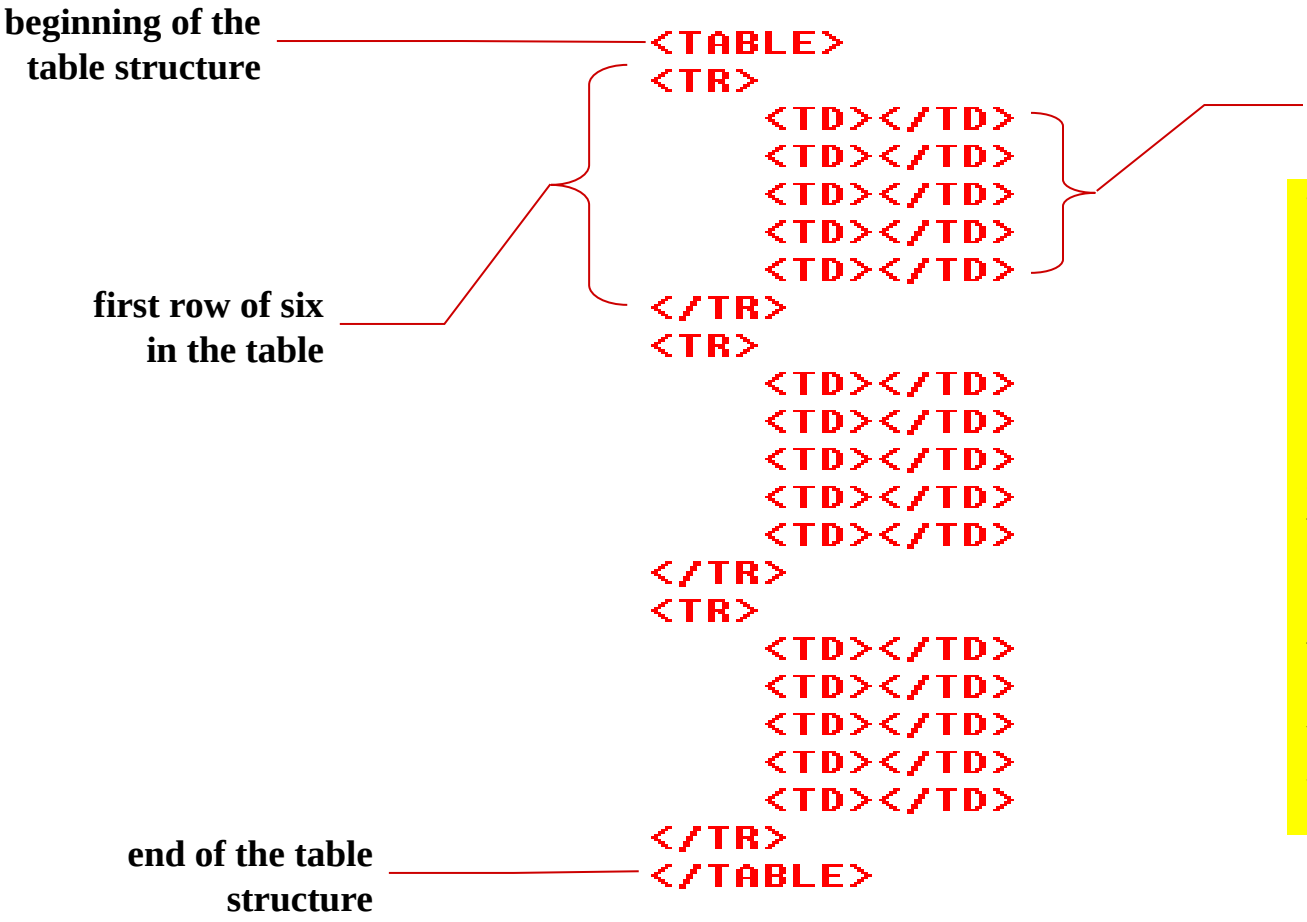
```
<table>
  <tr>
    <td> First Cell </td>
    <td> Second Cell </td>
  </tr>
  <tr>
    <td> Third Cell </td>
    <td> Fourth Cell </td>
  </tr>
</table>
```



Columns within a Table

- HTML does not provide a tag for table columns.
- In the original HTML specifications, the number of columns is determined by how many cells are inserted within each row.
  - for example, if you have four **<td>** tags in each table row, that table has four columns
- Later versions of HTML provide increased support for controlling the appearance of table columns.

HTML Structure of a Table



You do not need to indent the <td> tags or place them on separate lines, but you may find it easier to interpret your code if you do so.

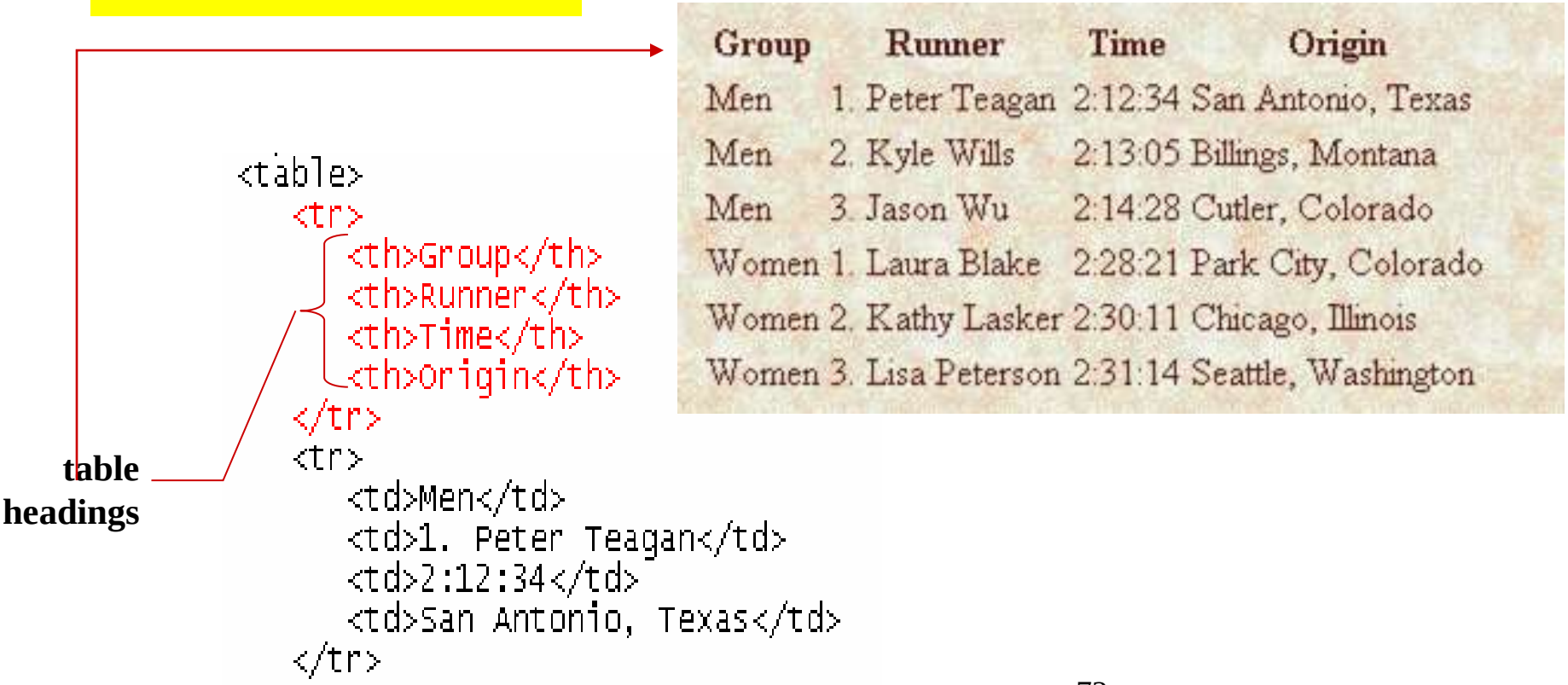
After the table structure is in place, you're ready to add the text for each cell.

Creating Headings with the <th> Tag

- HTML provides the **<th>** tag for table headings.
- Text formatted with the **<th>** tag is centered within the cell and displayed in a boldface font.
- The **<th>** tag is most often used for column headings, but you can use it for any cell that you want to contain centered boldfaced text.

Adding Table Headings to the Table

Text in cells formatted with the `<th>` tag is bold and centered above each table column.



Creating a Table Caption

- HTML allows you to specify a caption for a table.
- The syntax for creating a caption is: **<caption align="alignment">caption text</caption>**
  - ***alignment*** indicates the caption placement
  - a value of ***"bottom"*** centers the caption below the table
  - a value of ***"top"*** or ***"center"*** centers the caption above the table
  - a value of ***"left"*** or ***"right"*** place the caption above the table to the left or right
- The **<caption>** tag works only with tables, the tag must be placed within the table structure.
- Captions are shown as normal text without special formatting.
- Captions can be formatted by embedding the caption text within other HTML tags.
  - for example, place the caption text within a pair of **<b>** and **<i>** tags causes the caption to display as bold and italic

Result of a Table Caption

```
<table>
    <caption align="top"><b>Race Results</b></caption>
    <tr>
        <th>Group</th>
        <th>Runner</th>
        <th>Time</th>
        <th>Origin</th>
    </tr>
```

caption text

caption will be centered above the table

Race Results

Group	Runner	Time	Origin
Men	1. Peter Teagan	2:12:34	San Antonio, Texas
Men	2. Kyle Wills	2:13:05	Billings, Montana
Men	3. Jason Wu	2:14:28	Cutler, Colorado
Women	1. Laura Blake	2:28:21	Park City, Colorado
Women	2. Kathy Lasker	2:30:11	Chicago, Illinois
Women	3. Lisa Peterson	2:31:14	Seattle, Washington

Modifying the Appearance of a Table

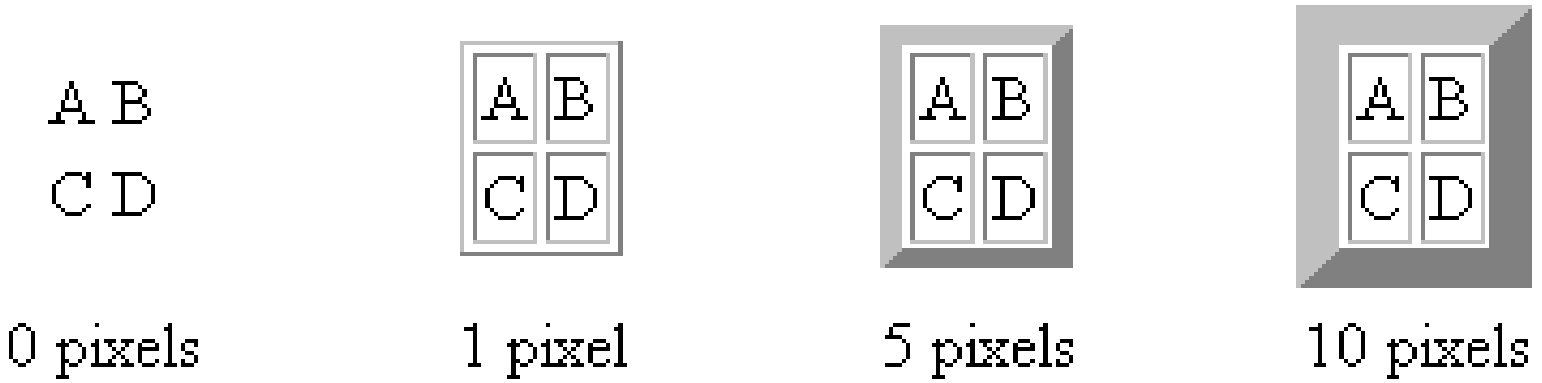
- You can modify the appearance of a table by adding:
  - gridlines
  - borders
  - background color
- HTML also provides tags and attributes to control the placement and size of a table.

Adding a Table Border

- By default, browsers display tables without table borders.
- A table border can be added using the border attribute to the **<table>** tag.
- The syntax for creating a table border is:
  <table border="value">
  - ***value*** is the width of the border in pixels
- The **size** attribute is optional; if you don't specify a size, the browser creates a table border 1 pixel wide.

Tables with Different Borders Values

This figure shows the effect on a table's border when the border size is varied.



Adding a 5-Pixel Border to a Table

```
<table border="5">
   <caption align="top"><b>Race Results</b></caption>
   <tr>
      <th>Group</th>
      <th>Runner</th>
      <th>Time</th>
      <th>Origin</th>
   </tr>
```

Only the outside border is affected by the border attribute; the internal gridlines are not affected.

Race Results

Group	Runner	Time	Origin
Men	1. Peter Teagan	2:12:34	San Antonio, Texas
Men	2. Kyle Wills	2:13:05	Billings, Montana
Men	3. Jason Wu	2:14:28	Cutler, Colorado
Women	1. Laura Blake	2:28:21	Park City, Colorado
Women	2. Kathy Lasker	2:30:11	Chicago, Illinois
Women	3. Lisa Peterson	2:31:14	Seattle, Washington

Controlling Cell Spacing

- The **cellspacing** attribute controls the amount of space inserted between table cells.
- The syntax for specifying the cell space is:

```
<table cellspacing="value">
```

  - ***value*** is the width of the interior borders in pixels
  - the default cell spacing is 2 pixels

- Cell spacing refers to the space between the cells.

Defining Cell Padding

- To control the space between the table text and the cell borders, add the **cellpadding** attribute to the table tag.
- The syntax for this attribute is:

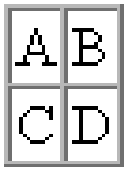
```
<table cellpadding="value">
```

  - ***value*** is the distance from the table text to the cell border, as measured in pixels
  - the default cell padding value is 1 pixel

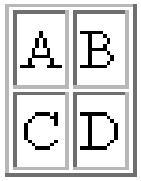
- Cell padding refers to the space within the cells.

Tables with Different Cell Spacing Values

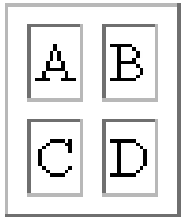
different cell spacing values



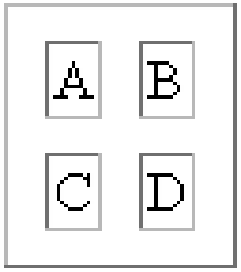
0 pixels



1 pixel

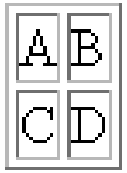


5 pixels

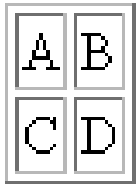


10 pixels

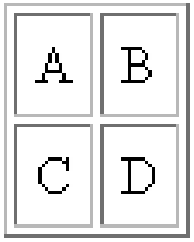
different cell padding values



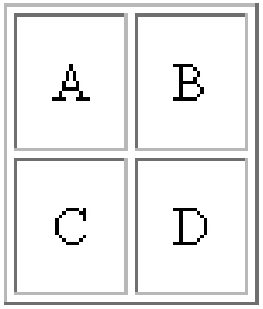
0 pixels



1 pixel



5 pixels



10 pixels

Table Frames and Rules

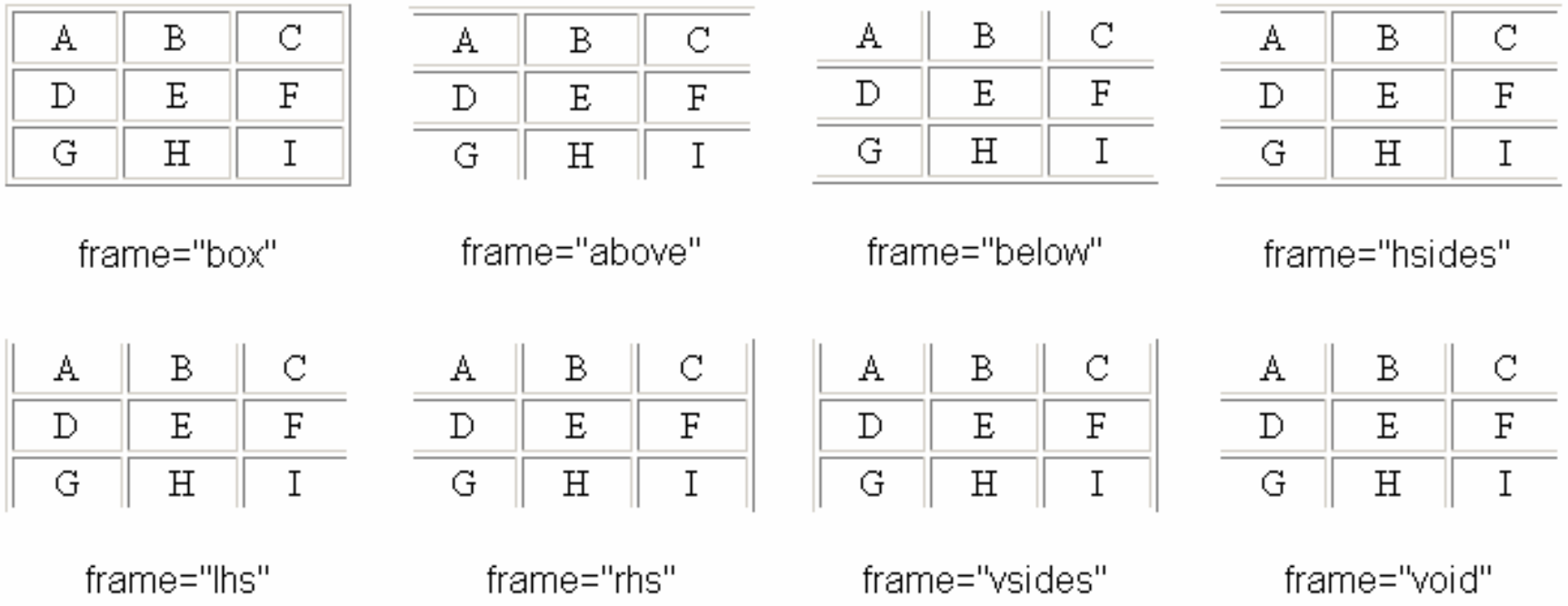
- Two additional table attributes introduced in HTML 4.0 are the **frames** and **rules** attributes.
- With the frame and rule attributes you can control how borders and gridlines are applied to the table.
- The **frames** attribute allows you to determine which sides of the table will have borders.
- The frame attribute syntax is: **`<table frame="type">`**
  - ***type*** is either "box" (the default), "**above**", "**below**", "**hsides**", "**vsides**", "**lhs**", "**rhs**", or "**void**"

Values of the Frame Attribute

FRAME VALUE	DESCRIPTION
BOX	Draws borders around all four sides
ABOVE	Draws only the top border
BELOW	Draws only the bottom border
HSIDES	Draws both the top and bottom borders (the horizontal sides)
LHS	Draws only the left-hand side
RHS	Draws only the right-hand side
VSIDES	Draws both the left and right borders (the vertical sides)
VOID	Does not draw borders on any of the four sides

Effect of Different Frame Values

This figure shows the effect of each of the frame values on the table grid.



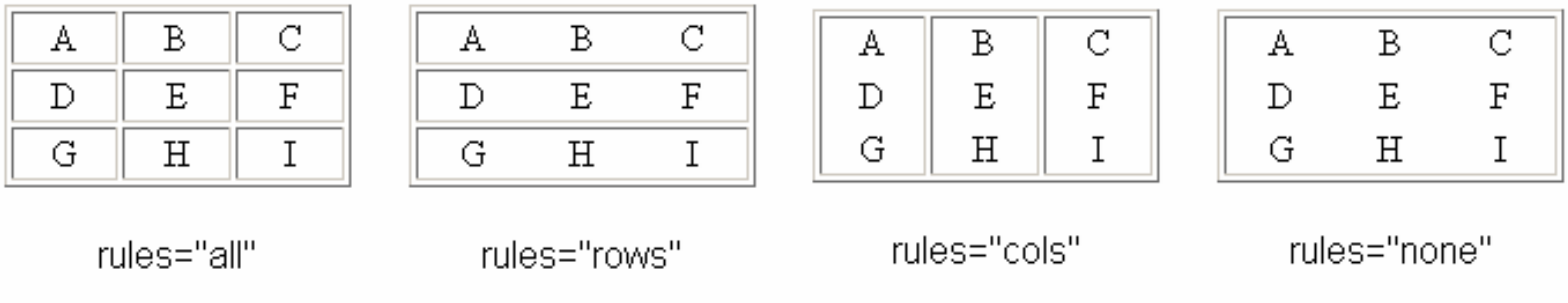
Creating Frames and Rules Continued

- The **rules** attribute lets you control how the table gridlines are drawn (not supported by Netscape)
- The syntax of the rules attribute is:

```
<table rules="type">
```

  - ***type*** is either "**all**", "**rows**", "**cols**", or "**none**"

the effect of each of the rules attribute values on a table



Working with Table and Cell Size

- The size of a table is determined by text it contains in its cells.
- By default, HTML places text on a single line.
- As you add text in a cell, the width of the column and table expands to the edge of the page.
  - once the page edge is reached, the browser reduces the size of the remaining columns to keep the text to a single line
- You can insert line break, paragraph, or other tags within a cell.
- When the browser can no longer increase or decrease the size of the column and table it wraps the text to a second line.
- As more text is added, the height of the table expands to accommodate the additional text.
- But, you can manually define the size of the table and its cells.

Defining the Table Size

- The syntax for specifying the table size is:

```
<table width="size" height="size">
```

  - ***size*** is the width and height of the table as measured in pixels or as a percentage of the display area
- To create a table whose height is equal to the entire height of the display area, enter the attribute height="100%".
- If you specify an absolute size for a table in pixels, its size remains constant, regardless of the browser or monitor settings used.
- Remember that some monitors display Web pages at a resolution of 640 by 480 pixels.

Setting the Width of the Table to 500 Pixels

```
<table border="5" cellspacing="0" cellpadding="4" width="500">
   <caption align="top"><b>Race Results</b></caption>
   <tr>
      <th>Group</th>
      <th>Runner</th>
      <th>Time</th>
      <th>Origin</th>
   </tr>
```

Park City native, **Laura Blake**, won the 27th Front Range Marathon over an elite field of the best long distance runners in the country. Laura's time of 2 hr. 28 min. 21 sec. was only 2 minutes off the women's course record set last year by Sarah Rawlings. Kathy Lasker and Lisa Peterson finished second and third, respectively. Laura's victory came on the heels of her performance at the NCAA Track and Field Championships, in which she placed second running for Colorado State.

Race Results

Group	Runner	Time	Origin
Men	1. Peter Teagan	2:12:34	San Antonio, Texas
Men	2. Kyle Wills	2:13:05	Billings, Montana
Men	3. Jason Wu	2:14:28	Cutler, Colorado
Women	1. Laura Blake	2:28:21	Park City, Colorado
Women	2. Kathy Lasker	2:30:11	Chicago, Illinois
Women	3. Lisa Peterson	2:31:14	Seattle, Washington

Defining Cell and Column Sizes

- To set the width of an individual cell, add the **width** attribute to either the **<td>** or **<th>** tags.
- The syntax is: **width="value"**
  - ***value*** can be expressed in pixels or as a percentage of the table width
  - width value of 30% displays a cell that is 30% of the total width of table.
- The **height** attribute can also be used in the **<td>** or **<th>** tags to set the height of individual cells.
  - The height attribute is expressed either in pixels or as a percentage of the height of the table.
  - If you include more text than can be displayed within that height value you specify, the cell expands to display the additional text.

Defining Cell and Column Sizes

- Specifying a width for an individual cell does not guarantee that the cell will be that width when displayed in the browser.
  - the reason for this is that the cell is part of a column containing other cells.
- Set the width of all the cells in the column to the same value to ensure that the cells do not change in size.

Aligning a Table on the Web Page

- By default, a browser places a table on the left margin of a Web page, with surrounding text placed above and below the table.
- To align a table with the surrounding text, use the **align** attribute as follows: **`align="alignment"`**
  - ***alignment*** equals **"*left*"**, **"*right*"**, or **"*center*"**
  - ***left*** or ***right*** alignment places the table on the margin of the Web page and wraps surrounding text to the side
  - ***center*** alignment places the table in the horizontal center of the page, but does not allow text to wrap around it
- The align attribute is similar to the align attribute used with the **`<img>`** tag.

Results of a Right-Aligned Table

```
<table border="5" cellspacing="0" cellpadding="4" width="500" align="right">
   <caption align="top"><b>Race Results</b></caption>
   <tr>
      <th width="50">Group</th>
      <th>Runner</th>
      <th>Time</th>
      <th>Origin</th>
   </tr>
```

Local Woman Wins Marathon

Park City native, **Laura Blake**, won the 27th Front Range Marathon over an elite field of the best long distance runners in the country. Laura's time of 2 hr. 28 min. 21 sec. was only 2 minutes off the women's course record set last year by Sarah Rawlings. Kathy Lasker and Lisa Peterson finished second and third, respectively. Laura's victory came on the heels of her performance at the NCAA Track and Field Championships, in which she placed second running for Colorado State.

In an exciting race, **Peter Teagan** of San Antonio, Texas, used a finishing kick to win the men's marathon for the second straight year, in a time of 2 hr. 12 min. 34 sec. Ahead for much of the race, Kyle Wills of Billings, Montana, finished second, when he could not match Teagan's finishing pace. Jason Wu of Cutler, Colorado, placed third in a very competitive field.

Race Results

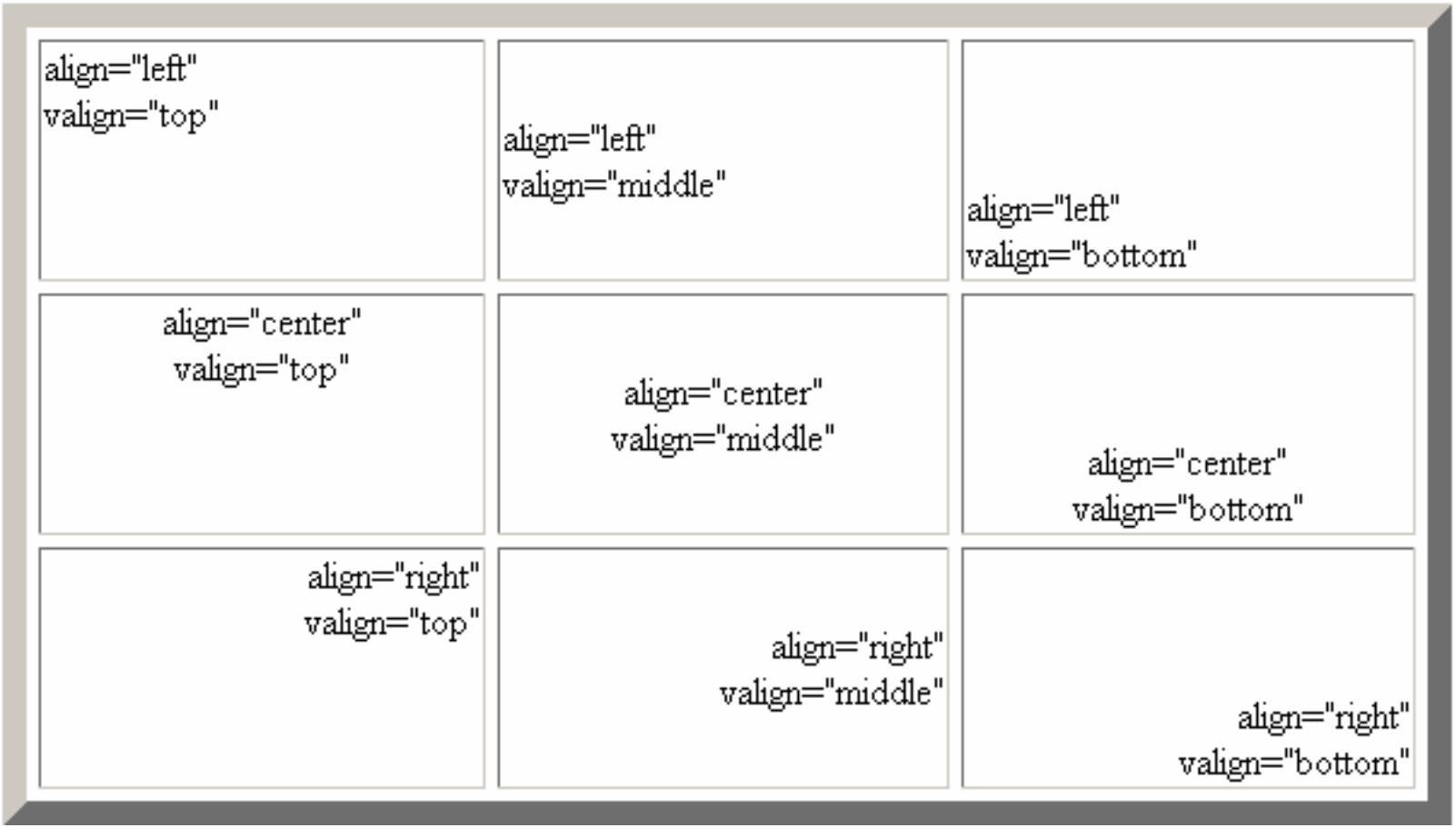
Group	Runner	Time	Origin
Men	1. Peter Teagan	2:12:34	San Antonio, Texas
Men	2. Kyle Wills	2:13:05	Billings, Montana
Men	3. Jason Wu	2:14:28	Cutler, Colorado
Women	1. Laura Blake	2:28:21	Park City, Colorado
Women	2. Kathy Lasker	2:30:11	Chicago, Illinois
Women	3. Lisa Peterson	2:31:14	Seattle, Washington

This year's race through downtown Boulder boasted the largest field in the marathon's history, with over 9500 men and 6700 women competing. Race conditions were perfect with low humidity and temperatures that never exceeded 85°.

Aligning the Contents of a Table

- By default, cell text is placed in the middle of the cell, aligned with the cell's left edge.
- By using the **align** and **valign** attributes, you can specify the text's horizontal and vertical placement.
- To align the text for a single column, you must apply the align attribute to every cell in that column.

Values of the Align and Valign Attributes



Spanning Rows and Columns

- To merge several cells into one, you need to create a **spanning cell**.
- A spanning cell is a cell that occupies more than one row or column in a table.
- Spanning cells are created by inserting the **rowspan** and **colspan** attribute in a **<td>** or **<th>** tag.
- The syntax for these attributes is:
  rowspan="value" colspan="value"
  - ***value*** is the number of rows or columns that the cell spans in the table

Example of Spanning Cells

This cell spans two columns and two rows

this cell spans three columns

This cell spans three rows

Today's Opinion Poll Question		Political Party		
		Democrat	Republican	Independent
"Do you favor or oppose increasing the minimum wage?"	Favor	70%	35%	55%
	Oppose	25%	60%	30%
	Unsure	5%	5%	15%

A Table Structure with a Row-Spanning Cell

four table cells in the first row

only three table cells are required for the second and third rows

```
<table>
   <tr>
      <td rowspan="3">1: This cell spans three rows</td>
      <td>2</td>
      <td>3</td>
      <td>4</td>
   </tr>
   <tr>
      <td>5</td>
      <td>6</td>
      <td>7</td>
   </tr>
   <tr>
      <td>8</td>
      <td>9</td>
      <td>10</td>
   </tr>
</table>
```

HTML code

1: This cell spans three rows	2	3	4
	5	6	7
	8	9	10

resulting table

Adding Spanning Cells to a Table

```
<table border="5" cellspacing="0" cellpadding="4" width="500" align="right">
   <caption align="top"><b>Race Results</b></caption>
   <tr>
      <th colspan="2">Runner</th>
      <th>Time</th>
      <th>Origin</th>
   </tr>
   <tr>
      <td rowspan="3">Men</td>
      <td>1. Peter Teagan</td>
      <td align="right">2:12:34</td>
      <td>San Antonio, Texas</td>
   </tr>
   <tr>
      <td>2. Kyle Wills</td>
      <td align="right">2:13:05</td>
      <td>Billings, Montana</td>
   </tr>
   <tr>
      <td>3. Jason Wu</td>
      <td align="right">2:14:28</td>
      <td>Cutler, Colorado</td>
   </tr>
   <tr>
      <td rowspan="3">Women</td>
      <td>1. Laura Blake</td>
      <td align="right">2:28:21</td>
      <td>Park City, Colorado</td>
   </tr>
   <tr>
      <td>2. Kathy Lasker</td>
      <td align="right">2:30:11</td>
      <td>Chicago, Illinois</td>
   </tr>
   <tr>
      <td>3. Lisa Peterson</td>
      <td align="right">2:31:14</td>
      <td>Seattle, Washington</td>
   </tr>
</table>
```

Race Results

Runner		Time	Origin
Men	1. Peter Teagan	2:12:34	San Antonio, Texas
	2. Kyle Wills	2:13:05	Billings, Montana
	3. Jason Wu	2:14:28	Cutler, Colorado
Women	1. Laura Blake	2:28:21	Park City, Colorado
	2. Kathy Lasker	2:30:11	Chicago, Illinois
	3. Lisa Peterson	2:31:14	Seattle, Washington

Another Example of Spanning Cells

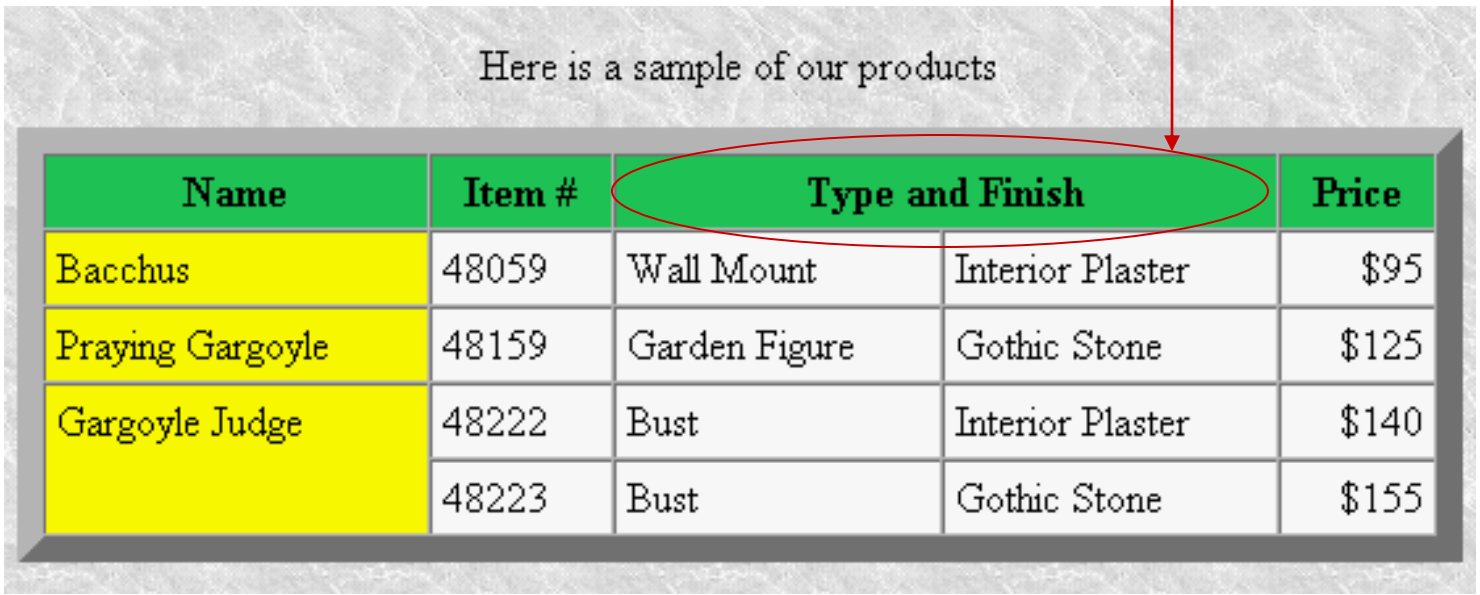
```
<TR>
    <TD BGCOLOR=YELLOW ROWSPAN=2>Gargoyle Judge</TD>
    <TD WIDTH=60>48222</TD>
    <TD>Bust</TD>
    <TD>Interior Plaster</TD>
    <TD ALIGN=RIGHT WIDTH=50>$140</TD>
</TR>
<TR>
    <TD WIDTH=60>48223</TD>
    <TD>Bust</TD>
    <TD>Gothic Stone</TD>
    <TD ALIGN=RIGHT WIDTH=50>$155</TD>
</TR>
</TABLE>
```

Here is a sample of our products

Name	Item #	Type	Finish	Price
Bacchus	48059	Wall Mount	Interior Plaster	$95
Praying Gargoyle	48159	Garden Figure	Gothic Stone	$125
Gargoyle Judge	48222	Bust	Interior Plaster	$140
	48223	Bust	Gothic Stone	$155

Another Example of Spanning Cells

```
<TABLE BORDER=10 CELLSPACING=0 CELLPADDING=4 ALIGN=CENTER WIDTH=550
 BGCOLOR=WHITE>
<CAPTION ALIGN=TOP>Here is a sample of our products</CAPTION>
<TR BGCOLOR="#33CC66">
    <TH>Name</TH>
    <TH WIDTH=60>Item #</TH>
    <TH COLSPAN=2>Type and Finish</TH>
    <TH WIDTH=50>Price</TH>
</TR>
```



Here is a sample of our products

Name	Item #	Type and Finish		Price
Bacchus	48059	Wall Mount	Interior Plaster	$95
Praying Gargoyle	48159	Garden Figure	Gothic Stone	$125
Gargoyle Judge	48222	Bust	Interior Plaster	$140
	48223	Bust	Gothic Stone	$155

Applying a Background Color

- Table elements support the **bgcolor** attribute.
- To specify a background color for all of the cells in a table, all of the cells in a row, or for individual cells, by adding the bgcolor attribute to either the **<table>**, **<tr>**, **<td>**, or **<th>** tags as follows:

```
<table bgcolor="color">
<tr bgcolor="color">
<td bgcolor="color">
<th bgcolor="color">
```

  - ***color*** is either a color name or hexadecimal color value

Specifying Table, Row, and Cell Colors

```
<table border="5" cellspacing="0" cellpadding="4" width="500" align="right" bgcolor="white">
   <caption align="top"><b>Race Results</b></caption>
   <tr bgcolor="yellow">
      <th colspan="2">Runner</th>
      <th>Time</th>
      <th>Origin</th>
   </tr>
   <tr>
      <td rowspan="3" valign="top" bgcolor="lightblue">Men</td>
      <td>1. Peter Teagan</td>
      <td align="right">2:12:34</td>
      <td>San Antonio, Texas</td>
   </tr>
   <tr>
      <td>2. Kyle Wills</td>
      <td align="right">2:13:05</td>
      <td>Billings, Montana</td>
   </tr>
   <tr>
      <td>3. Jason Wu</td>
      <td align="right">2:14:28</td>
      <td>Cutler, Colorado</td>
   </tr>
   <tr>
      <td rowspan="3" valign="top" bgcolor="lightgreen">Women</td>
      <td>1. Laura Blake</td>
      <td align="right">2:28:21</td>
      <td>Park City, Colorado</td>
   </tr>
   <tr>
      <td>2. Kathy Lasker</td>
      <td align="right">2:30:11</td>
      <td>Chicago, Illinois</td>
   </tr>
   <tr>
      <td>3. Lisa Peterson</td>
      <td align="right">2:31:14</td>
      <td>Seattle, Washington</td>
   </tr>
</table>
```

Specifying Table, Row, and Cell Colors

Race Results

Runner		Time	Origin
Men	1. Peter Teagan	2:12:34	San Antonio, Texas
	2. Kyle Wills	2:13:05	Billings, Montana
	3. Jason Wu	2:14:28	Cutler, Colorado
Women	1. Laura Blake	2:28:21	Park City, Colorado
	2. Kathy Lasker	2:30:11	Chicago, Illinois
	3. Lisa Peterson	2:31:14	Seattle, Washington

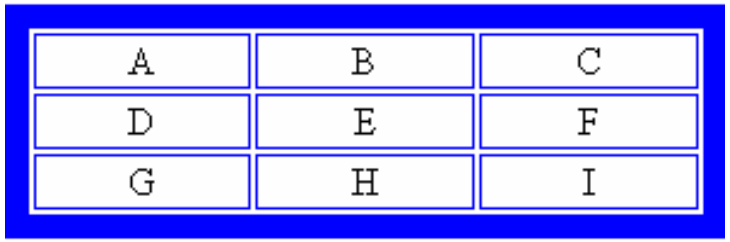
The bordercolor Attribute

- By default, table borders are displayed in two shades of gray that create a three-dimensional effect.
- The syntax for the bordercolor attribute is:

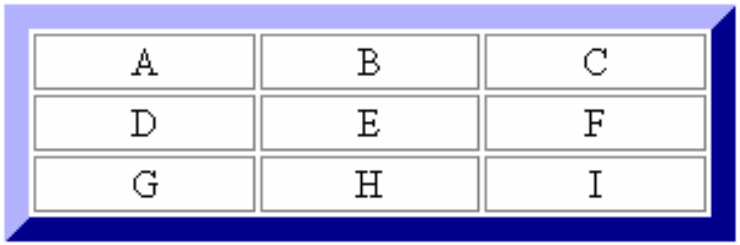
```
<table bordercolor="color">
```

  - ***color*** is an HTML color name or hexadecimal color value
- Internet Explorer and Netscape apply this attribute differently.

```
<table border="10"  bordercolor="blue">
```



Internet Explorer



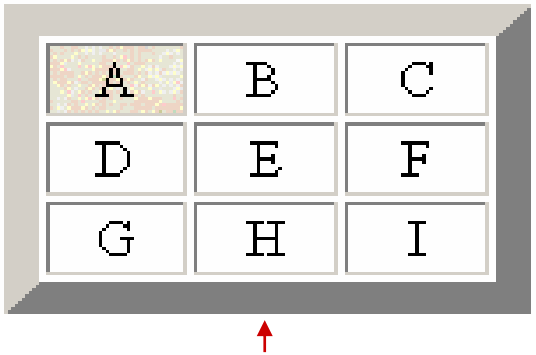
Netscape

Applying a Table Background

- Add a background image to your tables using the **`background`** attribute.
- A background can be applied to the entire table or to a cell.

parch.jpg

`<table background="parch.jpg">`



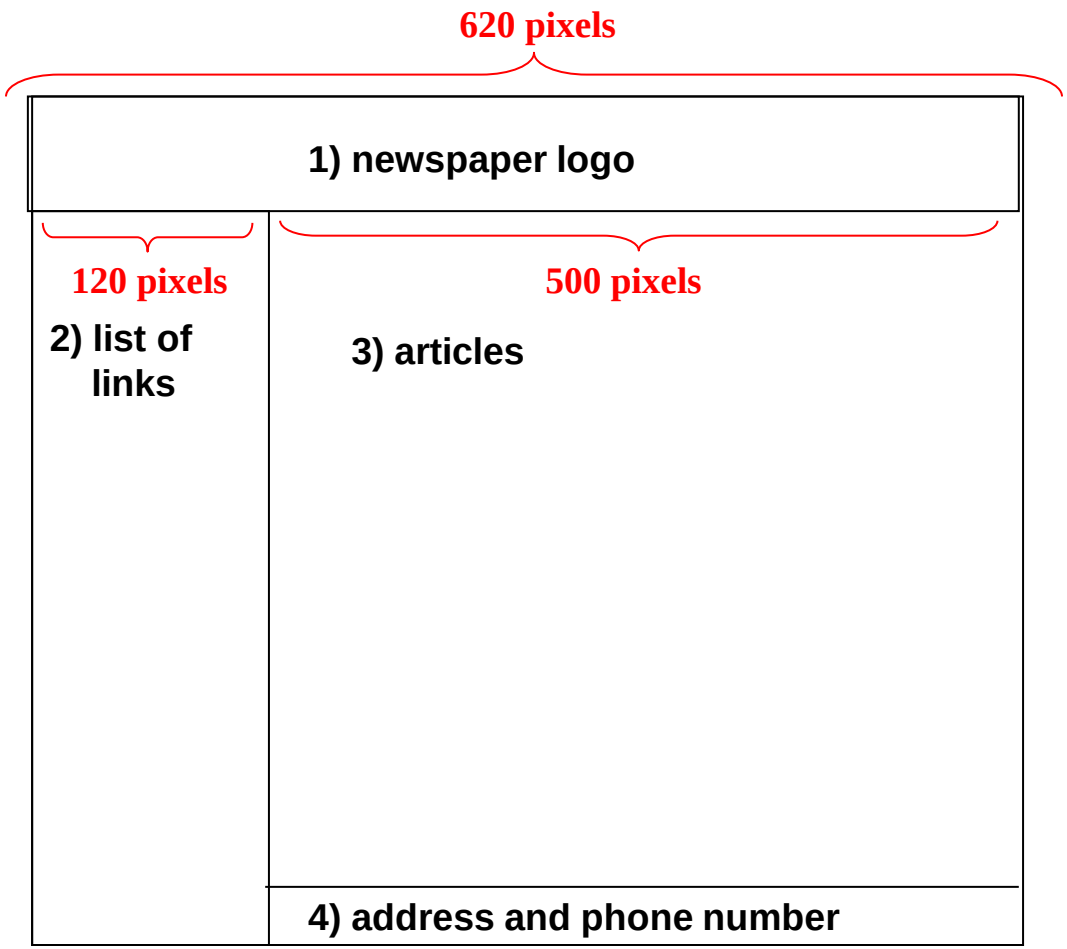
`<td background="parch.jpg">`

Designing a Page Layout with Tables

- HTML tables are most often used to define the layout of an entire Web page.
- If you want to design a page that displays text in newspaper style columns, or separates the page into distinct sections, you'll find tables an essential and useful tool.

Table Layout of a Web Page

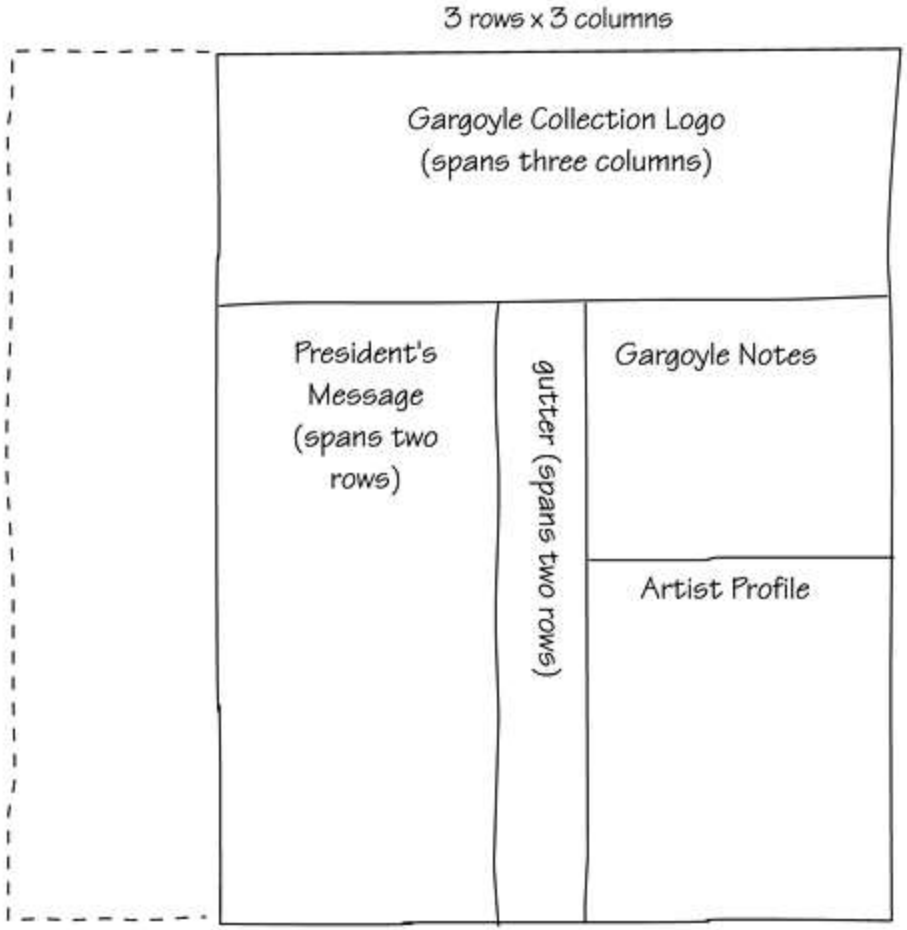
a sample table layout of a Web page.



Using Nested Table

- Tables can be created within another table making the Web page easier to manage.

a sketch of a web page using nested tables



The Result of the Web Page using Nested Tables

a sample web page using nested tables

An Example of the Contents

```
<TR>
   <!--List of Hypertext Links-->
   <TD WIDTH=165 VALIGN=TOP>
       <IMG SRC="MAA2.jpg" WIDTH=144 HEIGHT=25 ALT="Middle Age Arts">
       <H4><FONT COLOR=YELLOW>Middle Age Arts</FONT></H4>
       <FONT COLOR=WHITE>
       <A HREF="Index.htm">Home Page</A><BR>
       <A HREF="Catalog.htm">View the catalog</A><BR>
       <A HREF="Orders.htm">Place an order</A><BR>
       </FONT>
       <H4><FONT COLOR=YELLOW>About Gargoyles</FONT></H4>
       <FONT COLOR=WHITE>
       <A HREF="MAAtable.htm">Gargoyle Products</A><BR>
       <A HREF="MAAtext.htm">Gargoyle Products<BR>(text version)</A><BR>
       </FONT>
       <H4><FONT COLOR=YELLOW>Other Collections</FONT></H4>
       <FONT COLOR=WHITE>
       <A HREF="Vatican.htm">The Vatican Collection</A><BR>
       <A HREF="Rodin.htm">The Rodin Collection</A><BR>
       <A HREF="Masters.htm">Renaissance Masters</A><BR>
       </FONT>
   </TD>
```

An Example of the Gargoyle Notes

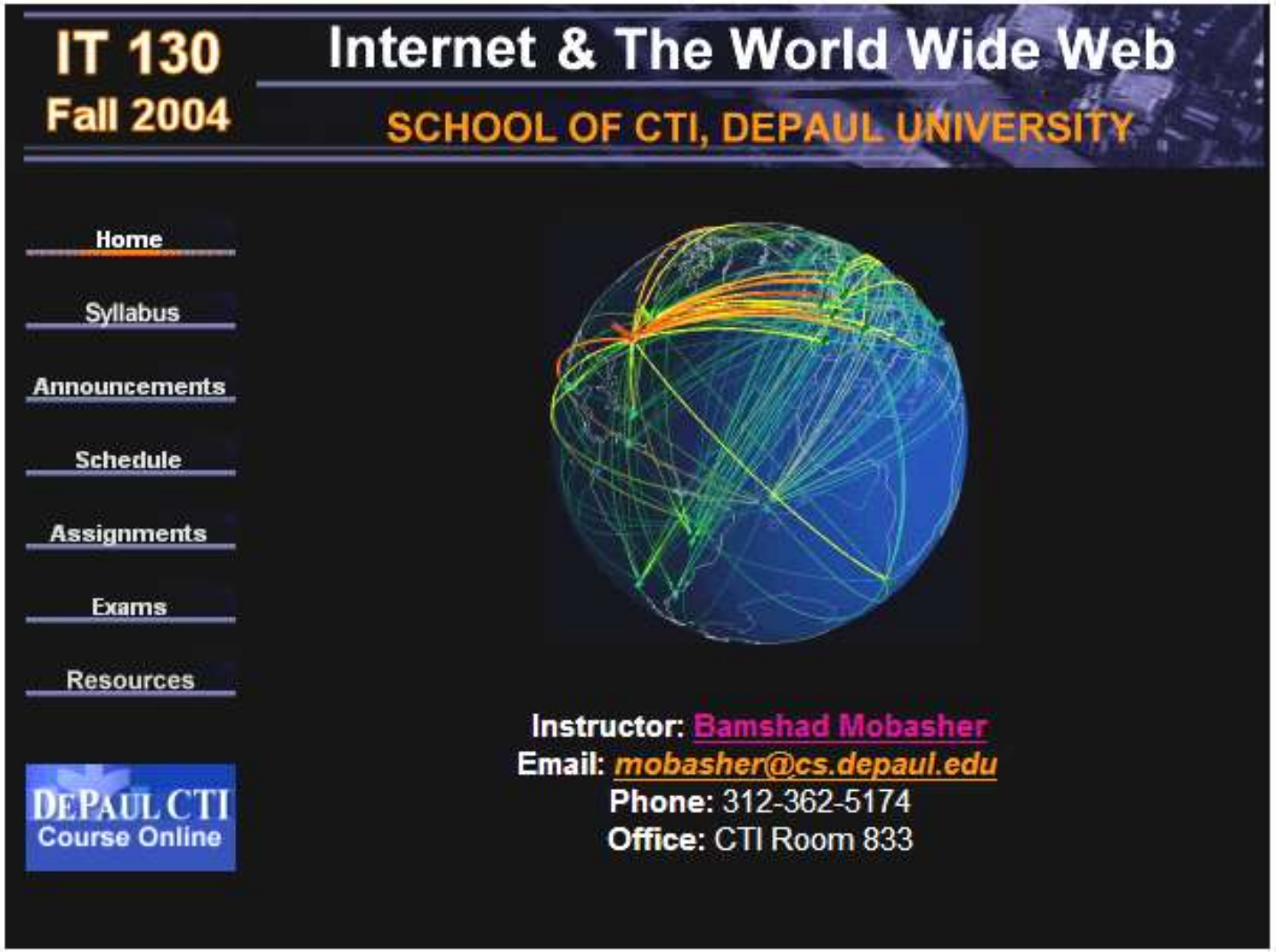
```
   <!--The table gutter-->
   <TD ROWSPAN=2 WIDTH=5> </TD>
   <!--Notes about gargoyles-->
   <TD WIDTH=220 VALIGN=TOP BGCOLOR=YELLOW>
      <FONT COLOR="#800000">
      <H4 ALIGN=CENTER>What do I do with a gargoyle?</H4>
      Don't think you need a gargoyle? Think again. Gargoyles are
      useful as:
      <UL>
        <LI>Bird baths
        <LI>Wind chimes
        <LI>Pen holders
        <LI>Paperweights
        <LI>Bookends
      </UL>
      </FONT>
   </TD>
</TR>
```

What do I do with a gargoyle?

Don't think you need a gargoyle? Think again. Gargoyles are useful as:

- Bird baths
- Wind chimes
- Pen holders
- Paperweights
- Bookends

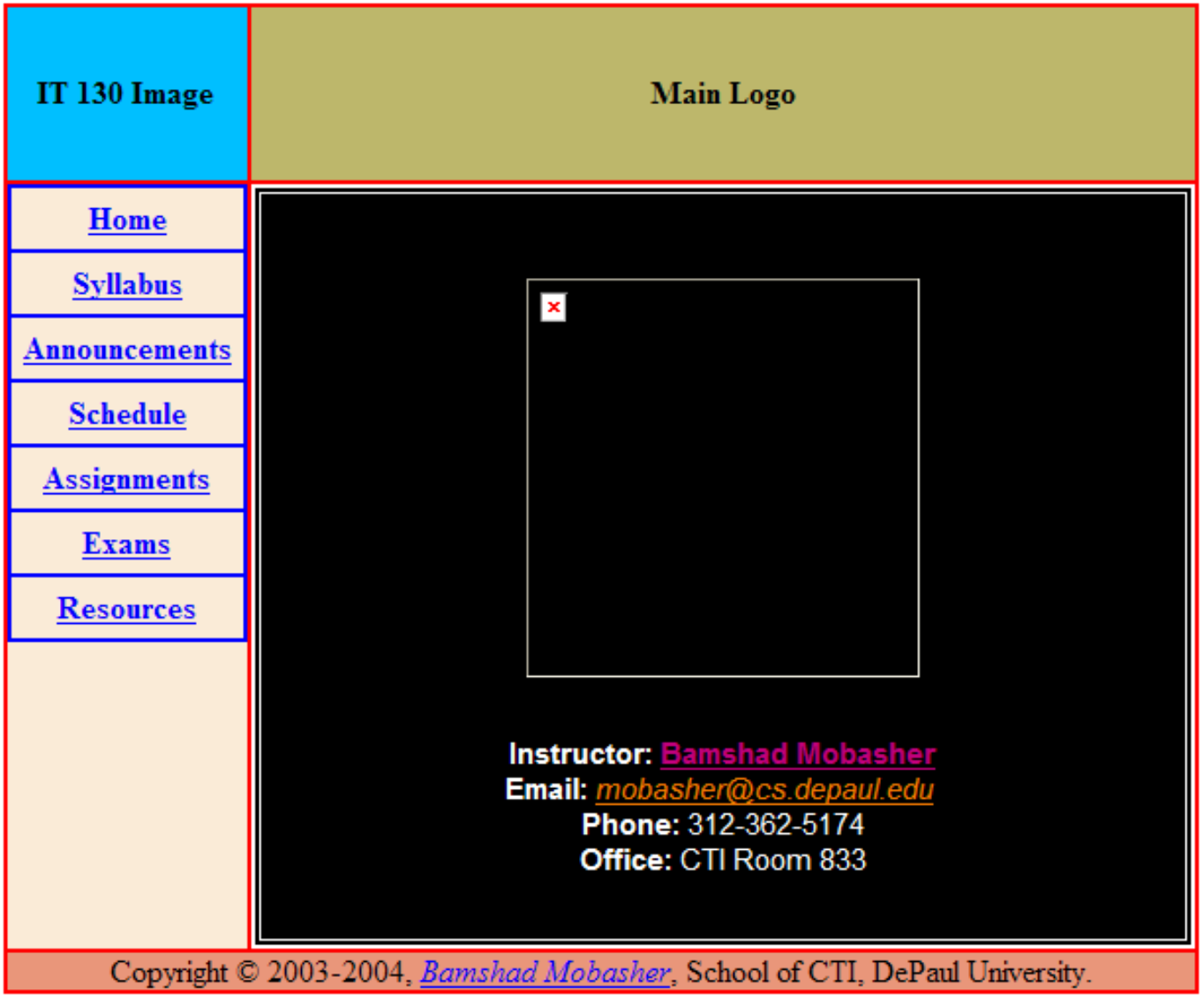
Example from Class Web Site



Sample Template based on Class Web Page

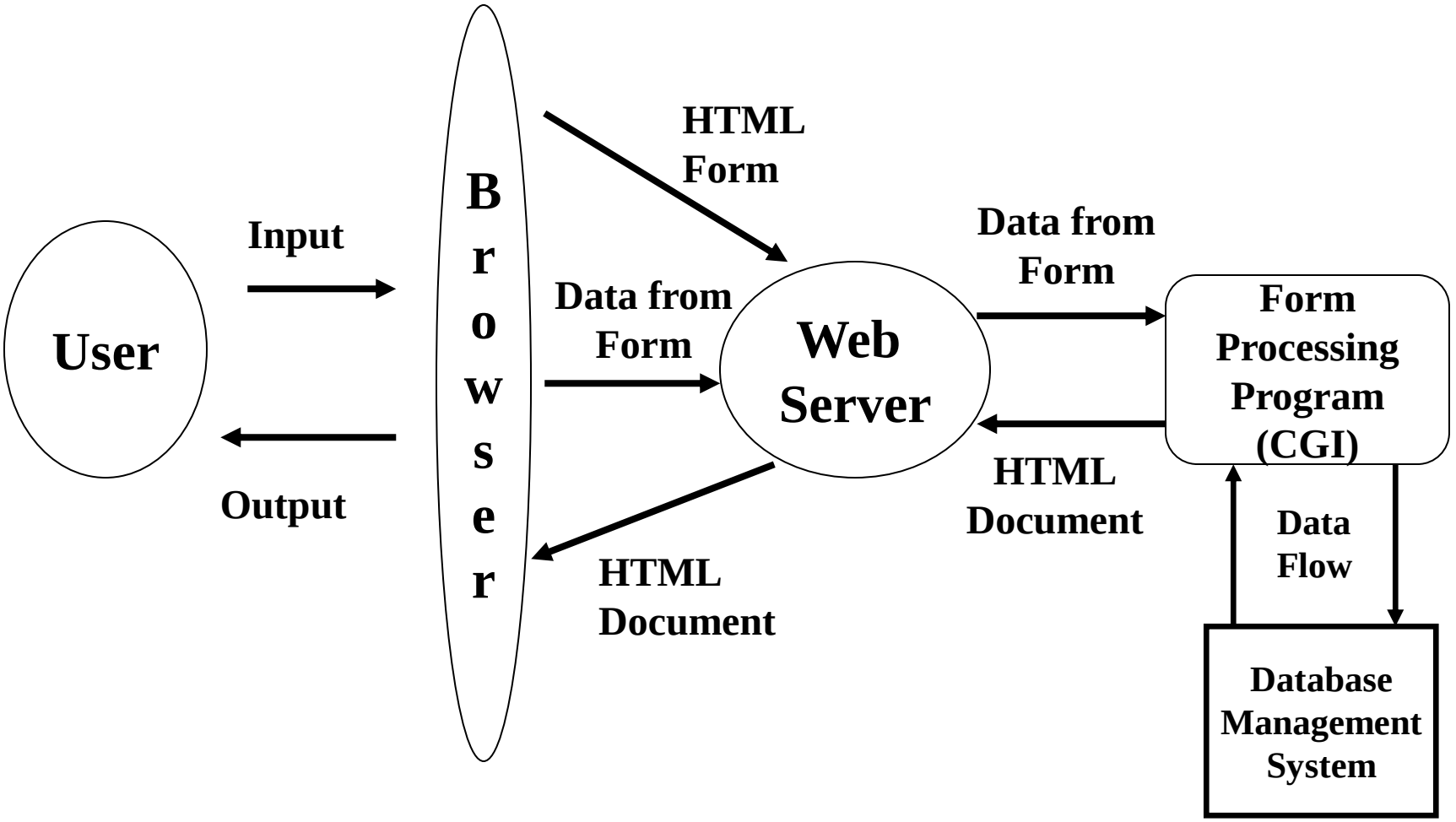
http://maya.cs.depau

HTML code



USER INTERACTIONS: FORMS

Form Processing



Flow of Information for Forms

Forms

- We want to design web-pages that are more interactive.
- Pages that
  - get the data from user (input).
  - Process data.
  - Return the result (output).
- A two way communication.
- The FORM element is used to create a data input form.
- A region using forms is enclosed within the <FORM> </FORM> tags.
- A document can have several forms, but the forms should not be embedded.
- The FORM element has three attributes:
  - ACTION, METHOD, NAME, and ENCTYPE.

- METHOD:
  - Specifies the way in which the data from the user are encoded.
  - The default METHOD is GET.
  - GET: The CGI program receives the encoded form input in the QUERY_STRING variable, which follows the "?" in the URL that calls the script.
  - POST: The CGI script or program receives the encoded form input in its standard input stream. The CONTENT_LENGTH must be used.
  - when GET is used, the submitted form data will be **visible in the page address field:** /action_page.php?firstname=Mickey&lastname=Mouse
  - Appends form-data into the URL in name/value pairs
  - The length of a URL is limited (about 3000 characters)
  - Never use GET to send sensitive data! (will be visible in the URL)
  - GET is better for non-secure data, like query strings in Google

- Always use POST if the form data contains sensitive or personal information. The POST method does not display the submitted form data in the page address field.
- POST has no size limitations, and can be used to send large amounts of data.
- Form submissions with POST cannot be bookmarked
- Name:- Specifies a name used to identify the form (for DOM usage: document.forms.name).

Forms

- **ACTION:**
  - **Specifies the destination URL to which the form should be submitted, once it has been completed by the user.**
  - **If no URL is specified, the URL of the current document containing the form is used.**
  - **MAILTO Action: The data from the form is mailed to the specified E-mail address. Use the POST method.**

Forms

- **ENCTYPE:**
  - **Tell the browser how the data from a form should be encoded when it is returned to the server.**
  - **The default is “application/x-www-form-urlencoded” that converts spaces to “+” and uses “&” to delineated different data fields.**

Form Tag

<Form Action="getemp.asp" Method="post">

(all form elements inside)

</Form>

Forms

- **Form Input: INPUT**
  - **Only used within a FORM element and is denoted by <INPUT>.**
  - **Attributes:**
    - ***NAME*: The name of the particular element.**
    - ***MAXLENGTH*: The maximum number of characters that can be entered by users in a text field.**
    - ***SIZE*: Specifies the size of the field and depends on its type.**
    - ***SRC*: Denote URL for an image.**
    - ***VALUE*: Contain the initial value displayed to users.**
    - ***TYPE*: Defines the type of data used in the field.**
    - ***CHECKED*: Indicates that a checkbox or radio button is selected.**
    - ***DISABLED*: Prevents the field from receiving focus.**
    - ***ALIGN*: Alignment if image is used.**
    - ***READONLY*: Prevents modification of the contents of the field.**

TYPE Attribute

- **Checkbox:**
  - **An object where several values can be selected at the same time.**
  - **The checkbox is submitted as separate name/value pair for each selected value.**
  - **Checkbox that are grouped together should have**
    - **a different *name*.**
    - **a unique *value*.**

TYPE Attribute

- **Checkbox:**
- Checkboxes let a user select ZERO or MORE options of a limited number of choices.

<FORM>

What pets do you own?

<input type="checkbox" name="vehicle1" value="Bike"> I have a bike

<input type="checkbox" name="vehicle2" value="Car"> I have a car

</FORM>

TYPE Attribute

- **Radio Boxes:**
  - An object that defines an item where only one value can be selected from a set of possibilities.
  - A set is defined as the group of radio boxes with the same NAME attribute.
  - Never set more than one radio box at a time with the CHECKED attribute.

```
<P><B>Gender:</B>
<input type="radio" name="gender" value="male" checked> Male<br>
<input type="radio" name="gender" value="female"> Female<br>
<input type="radio" name="gender" value="other"> Other
```

TYPE Attribute

- **HIDDEN Type:**
  - **When this type is chosen, no field is presented to the user.**
  - **Primary use is record keeping for programs that may parse user input from forms.**
  - **May be used for shopping carts.**

TYPE Attribute

- **TEXT type:**
  - **Specifies a single line text entry field.**
  - **Can be used with the MAXLENGTH and SIZE attributes (MAXLENGTH >= SIZE)**

<P><B> First Name:</B> <INPUT NAME="fname" TYPE = text MAXLENGTH=30 SIZE =30></P>

<P><B> Last Name:</B> <INPUT NAME="lname" TYPE = text MAXLENGTH=30 SIZE =30></P>

TYPE Attribute

- **PASSWORD Type:**
  - **Same as text except the text entered by the user is obscured.**
  - **Use the MAXLENGTH and SIZE attributes.**

<P><B> Enter Your Password:</B>

<INPUT NAME="password" TYPE = password MAXLENGTH=30 SIZE =30></P>

TYPE Attribute

- **SUBMIT and RESET Types:**
  - **SUBMIT: Used to submit the form's content, as specified by the ACTION attribute.**
  - **RESET: Set all fields in the form to their initial values.**

<P>INPUT TYPE=SUBMIT>

<INPUT TYPE=RESET>

<P><INPUT TYPE=SUBMIT VALUE = "Place Your Order">

<INPUT TYPE=RESET VALUE = "Start over">

TYPE Attribute

- **BUTTON Input Type:**
  - **Creates a button whose use can be defined through scripting and onClick event.**
  - **Use to create a back button.**
  - **Only useful to browsers that support scripting.**

<FORM><P><INPUT TYPE="button" VALUE="Back to Last Document" onClick="history.back( )"></P></FORM>

TEXTAREA

- Let users enter more than one line of text.
- Uses attributes ROWS and COLS to size.
- WRAP Attribute:
  - OFF: No wrapping
  - VIRTUAL: Display wraps but long lines are sent as one line.
  - PHYSICAL: Word wraps and text is sent with wrap points.

PullDown Menu

- Use SELECT and OPTION to create pulldown menu.
- SELECT:
  - Allows the user to choose one (or possibly more) items from a list.
  - Attributes: MULTIPLE, SIZE, and NAME.
- OPTION:
  - Specifies the list items.
  - Attributes: SELECTED, VALUE, and LABEL

PullDown Menu

- **Example:**

```
<P><B>Pick your favorite baseball team:</B>
<BR><SELECT NAME="team">
<OPTION>Dodgers
<OPTION>Braves
<OPTION>Cardinals
<OPTION>Yankees
</SELECT>
```

Example program

```
<html>
<body>
<form name="input" action="next.html" method="post">
   <label>First name:</label>
   <input type="text" name="fn" value="Smith" /><br />
   <label>Last name:</label>
   <input type="text" name="ln" value="Alfred" /><br />
   <label>Password:</label>
   <input type="password" name="pwd" />
   <label>Gender :</label>
  <input type="radio" name="sex" value="male" /> Male
   <input type="radio" name="sex" value="female" />
  Female<br />
```

```
 <label>Hobbies :</label>
 <input type="checkbox" name="cook" value="cooking"/>Cooking
<input type="checkbox" name="Read" value="Read"/>Reading
 <br /> <label>Address :</label>
 <textarea  rows="10" cols="30"></textarea>
 <label>State :</label>
 <select>
        <option>Kerala</option>
       <option>Karnataka</option>
 </select>
   <input type="submit" value="Submit" />
  <input type="reset" value="Reset"/>
</form>
</body></html>
```