Introduction to HTML

Brief History of the Internet

- 1969: ARPANET created by the U.S. Department of Defense
- 1983: TCP/IP became the standard protocol
- 1991: Tim Berners-Lee developed the World Wide Web
- 1993: Mosaic browser released
- 1995-Present: Growth of the Internet with cloud computing, social media, and streaming services

How Does the Internet Work?

- Uses TCP/IP for communication
- Clients (browsers) send requests to servers
- Servers respond with requested content
- Routers and ISPs route data packets
- DNS translates domain names to IP addresses

Internet Protocol & HTTP

- Internet Protocol (IP) assigns unique addresses to devices
- TCP ensures reliable data transfer
- HTTP (Hyper Text Transfer Protocol) facilitates web communication
- Stateless request-response model

Domain Names & DNS

- Domain names (e.g., google.com) are user-friendly addresses
- DNS (Domain Name System) maps domain names to IP addresses
- Uses hierarchical structure: Root, TLD (.com, .org), and subdomains

HTTP Protocols

- HTTP/1.0: Opens a new connection for each request
- HTTP/1.1: Introduced persistent connections and chunked transfer
- HTTP/2: Supports multiplexing, header compression
- HTTP/3: Uses QUIC for lower latency

HTTP Methods

- GET: Retrieve data
- POST: Submit data
- PUT: Update/create resource
- DELETE: Remove resource
- HEAD: Retrieve headers without body

HTTP Status Codes

- 1xx: Informational (100 Continue)
- 2xx: Success (200 OK)
- 3xx: Redirection (301 Moved Permanently)
- 4xx: Client Errors (404 Not Found)
- 5xx: Server Errors (500 Internal Server Error)

Stateless Nature of HTTP & HTTPS

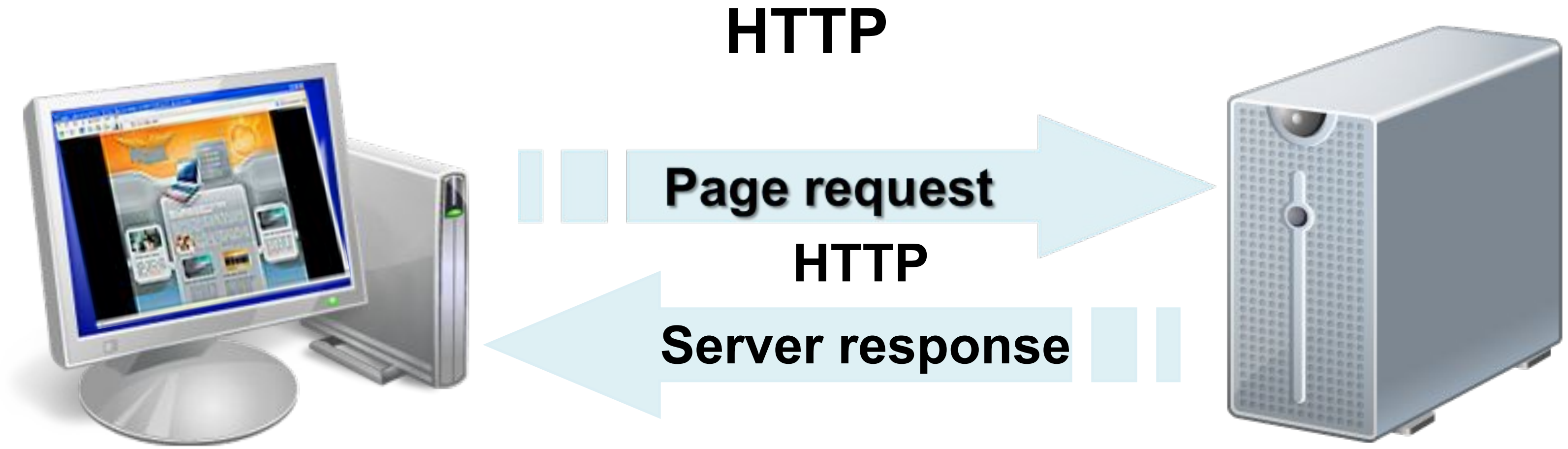
- HTTP is stateless: Each request is independent
- Uses cookies and session storage for maintaining state
- HTTPS encrypts data using SSL/TLS for security

Architecture of the Web

- Client (browser) requests resources
- Web server processes requests
- Database stores and retrieves data
- APIs facilitate data exchange

How the Web Works?

- WWW use classical client / server architecture
  - HTTP is text-based request-response protocol



HTTP

Page request

HTTP

Server response

Client running a Web Browser

Server running Web Server Software (IIS, Apache, etc.)

Definitions

- W W W – World Wide Web.
- HTML – **Hyper Text Markup Language** – The Language of Web Pages on the World Wide Web.HTML is a text formatting language.
- URL – Uniform Resource Locator.
- Browser – A software program which is used to show web pages.

Tags

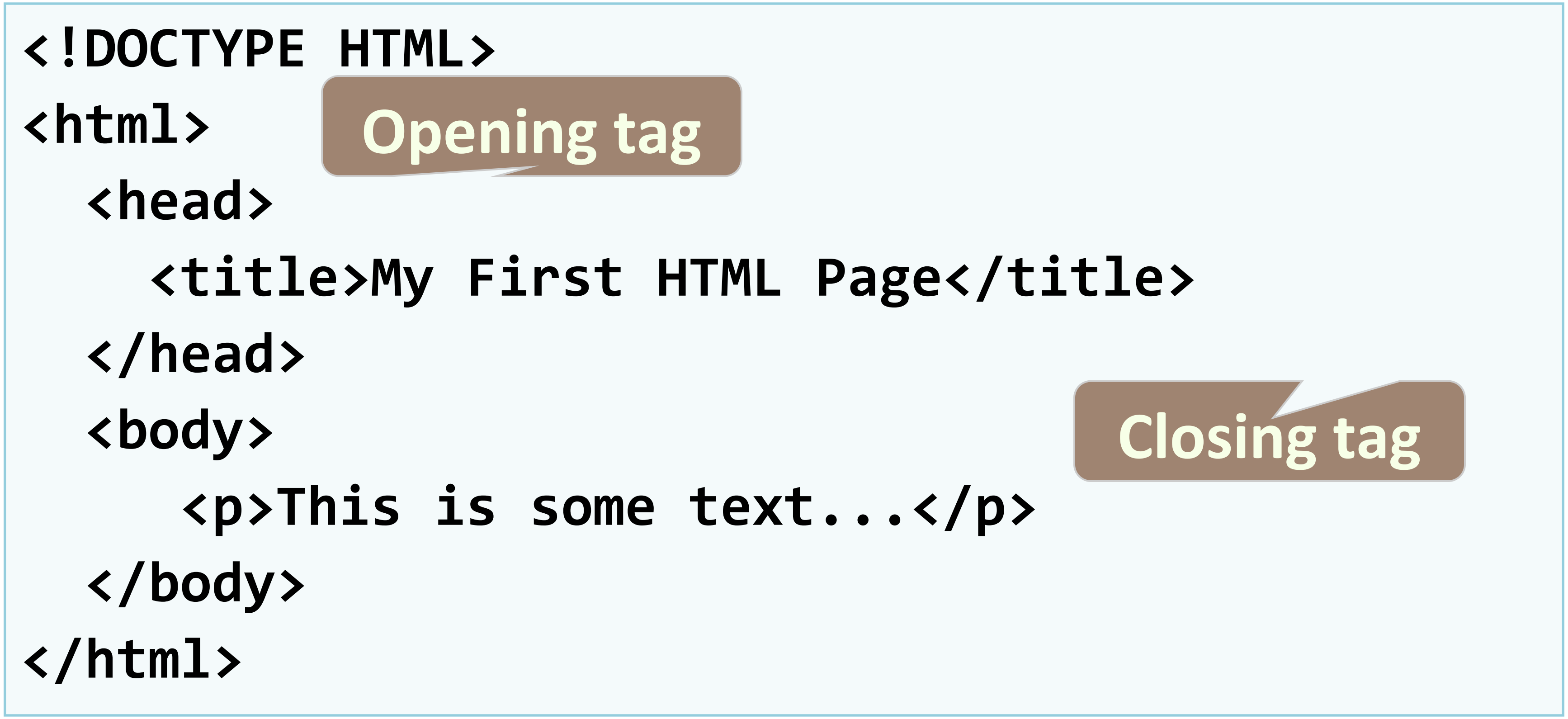
- Codes enclosed in brackets
- Usually paired

  <TITLE>My Web Page</TITLE>

- ***Not*** case sensitive

  <TITLE> = <title> = <TITLE>

First HTML Page: Tags



```
<!DOCTYPE HTML>
<html>
  <head>
    <title>My First HTML Page</title>
  </head>
  <body>
     <p>This is some text...</p>
  </body>
</html>
```

An HTML element consists of an opening tag, a closing tag and the content inside.

Creating a Basic Starting Document

The HEAD of your document point to above window part. The TITLE of your document appears in the very top line of the user’s browser. If the user chooses to “Bookmark” your page or save as a “Favorite”; it is the TITLE that is added to the list.

The text in your TITLE should be as descriptive as possible because this is what many search engines, on the internet, use for indexing your site.

The Body Element

- The BODY element of a web page is an important element in regards to the page's appearance. The following example will display a document with a background color and text color as red:

- **<BODY BGCOLOR="#00FF00" TEXT="#FF0000"></BODY>**
- The default color for text is black.

Background Color and Text color

The following example will display a document with a background color and text color as red:

```
<BODY BGCOLOR=“#00FF00
   TEXT=“#FF0000”></BODY>
```

The default color for text is black.

Headings, <Hx> </Hx>

Inside the **BODY** element, heading elements **H1** through **H6** are generally used for major divisions of the document. Headings are permitted to appear in any order, but you will obtain the best results when your documents are displayed in a browser if you follow these guidelines:

H1: should be used as the highest level of heading, **H2** as the next highest, and so forth.

You should not skip heading levels: e.g., an **H3** should not appear after an **H1**, unless there is an **H2** between them.

Headings, <Hx> </Hx>

```
<HTML>
<HEAD>
<TITLE> Example Page</TITLE>
</HEAD>
<BODY>
<H1> Heading 1 </H1>
<H2> Heading 2 </H2>
<H3> Heading 3 </H3>
<H4> Heading 4 </H4>
<H5> Heading 5 </H5>
<H6> Heading 6 </H6>
</BODY>
</HTML>
```

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Paragraphs, <P> </P>

Paragraphs allow you to add text to a document in such a way that it will automatically adjust the end of line to suite the window size of the browser in which it is being displayed. Each line of text will stretch the entire length of the window.

Paragraphs, <P> </P>

```
<HTML><HEAD>
<TITLE> Example Page</TITLE>
</HEAD>
<BODY></H1> Heading 1 </H1>
<P> Paragraph 1, ….</P>
<H2> Heading 2 </H2>
<P> Paragraph 2, ….</P>
<H3> Heading 3 </H3>
<P> Paragraph 3, ….</P>
<H4> Heading 4 </H4>
<P> Paragraph 4, ….</P>
<H5> Heading 5 </H5>
<P> Paragraph 5, ….</P>
<H6> Heading 6</H6>
<P> Paragraph 6, ….</P>
</BODY></HTML>
```

Heading 1

Paragraph 1,….

Heading 2

Paragraph 2,….

Heading 3

Paragraph 3,….

Heading 4

Paragraph 4,….

Heading 5

Paragraph 5,….

Heading 6

Paragraph 6,….

Break, <BR>

Line breaks allow you to decide where the text will break on a line or continue to the end of the window.

A <BR> is an empty Element, meaning that it may contain attributes but it does not contain content.

The <BR> element does not have a closing tag.

Break, <BR>

```
<HTML>
<HEAD>
<TITLE> Example Page</TITLE>
</HEAD>
<BODY>
<H1> Heading 1 </H1>
<P>Paragraph 1, <BR>
Line 2 <BR> Line 3 <BR>….
</P>
</BODY>
</HTML>
```

Heading 1

Paragraph 1,….

Line 2

Line 3

….

Horizontal Rule, <HR>

The <HR> element causes the browser to display a horizontal line (rule) in your document.

<HR> does not use a closing tag, </HR>.

Lists

In this chapter you will learn how to create a variety of lists.

Objectives

Upon completing this section, you should be able to

- Create an unordered list.
- Create an ordered list.
- Create a defined list.
- Nest Lists.

Lists

There are three types of lists:

UL : Unordered List. Items in this list start with a list mark such as a bullet. Browsers will usually change the list mark in nested lists.

```
<UL>
<LI> List item …</LI>
<LI> List item …</LI>
</UL>
```

List item …

List item …

List Elements

You have the choice of three bullet types: **disc(default), circle, square.**

These are controlled by the “TYPE” attribute for the <UL> element.

```
<UL TYPE=“square”>
<LI> List item …</LI>
<LI> List item …</LI>
<LI> List item …</LI>
</UL>
```

List item …

List item …

List item …

List Elements

OL: Ordered List. Items in this list are numbered automatically by the browser.

```
<OL>
<LI> List item …</LI>
<LI> List item …</LI>
<LI> List item …</LI>
</OL>
```

List item …

List item …

List item

You have the choice of setting the TYPE Attribute to one of five numbering styles.

List Elements

TYPE	Numbering Styles	
1	**Arabic numbers**	**1,2,3, ……**
a	**Lower alpha**	**a, b, c, ……**
A	**Upper alpha**	**A, B, C, ……**
i	**Lower roman**	**i, ii, iii, ……**
I	**Upper roman**	**I, II, III, ……**

List Elements

You can specify a starting number for an ordered list.

```
<OL TYPE =“i”>
<LI> List item …</LI>
<LI> List item …</LI>
</OL>
<P> text ….</P>
<OL TYPE=“i” START=“3”>
<LI> List item …</LI>
</OL>
```

List Elements

List item …

List item …

Text ….

List item …

List Elements

DL: Definition List. This kind of list is different from the others. Each item in a DL consists of one or more **Definition Terms (DT elements),** followed by one or more **Definition Description (DD elements).**

```
<DL>
<DT> HTML </DT>
<DD> Hyper Text Markup Language </DD>
<DT> DOG </DT>
<DD> A human’s best friend!</DD>
</DL>
```

HTML

Hyper Text Markup Language

DOG

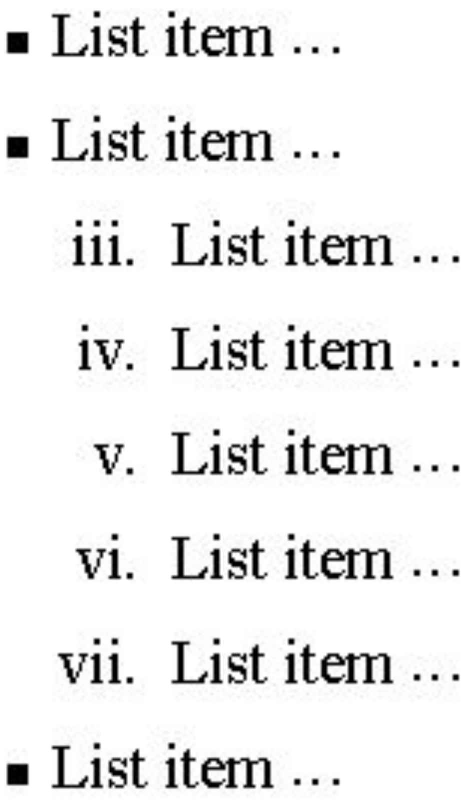
A human’s best friend!

Nesting Lists

You can nest lists by inserting a UL, OL, etc., inside a list item (LI).

Example

```
<UL TYPE = “square”>
<LI> List item …</LI>
<LI> List item …
<OL TYPE=“i” START=“3”>
<LI> List item …</LI>
<LI> List item …</LI>
<LI> List item …</LI>
<LI> List item …</LI>
<LI> List item …</LI>
</OL>
</LI>
<LI> List item …</LI>
</UL>
```



▪ List item …

▪ List item …

iii. List item …

iv. List item …

v. List item …

vi. List item …

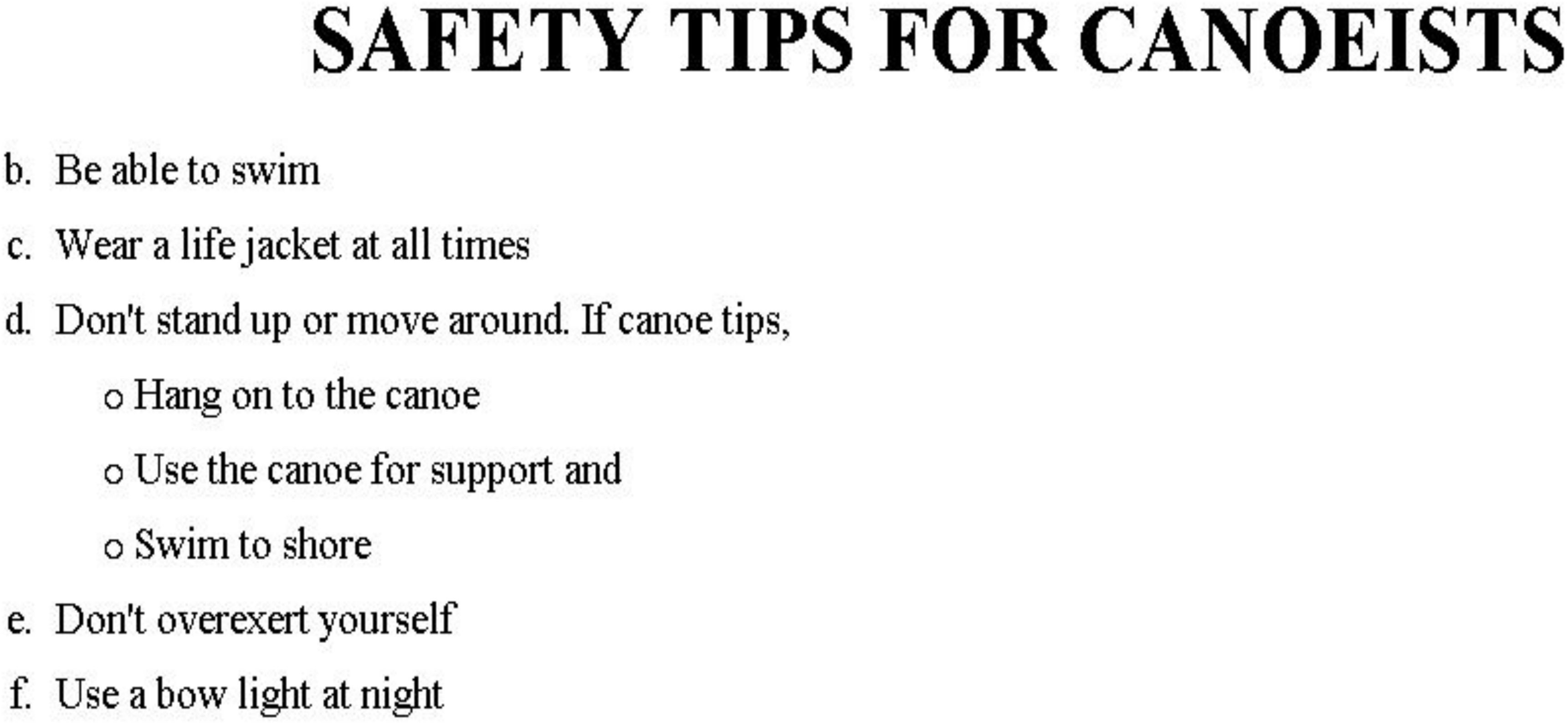
vii. List item …

▪ List item …

What will be the output?

```
<H1 ALIGN="CENTER">SAFETY TIPS FOR CANOEISTS</H1>
<OL TYPE=“a” START=“2”>
<LI>Be able to swim </LI>
<LI>Wear a life jacket at all times </LI>
<LI>Don't stand up or move around. If canoe tips,
    <UL>
    <LI>Hang on to the canoe </LI>
    <LI>Use the canoe for support and </LI>
     <LI>Swim to shore
    </UL> </LI>
<LI>Don't overexert yourself </LI>
<LI>Use a bow light at night </LI>
</OL>
```

The output….



SAFETY TIPS FOR CANOEISTS

b. Be able to swim

c. Wear a life jacket at all times

d. Don't stand up or move around. If canoe tips,

   - Hang on to the canoe
   - Use the canoe for support and
   - Swim to shore

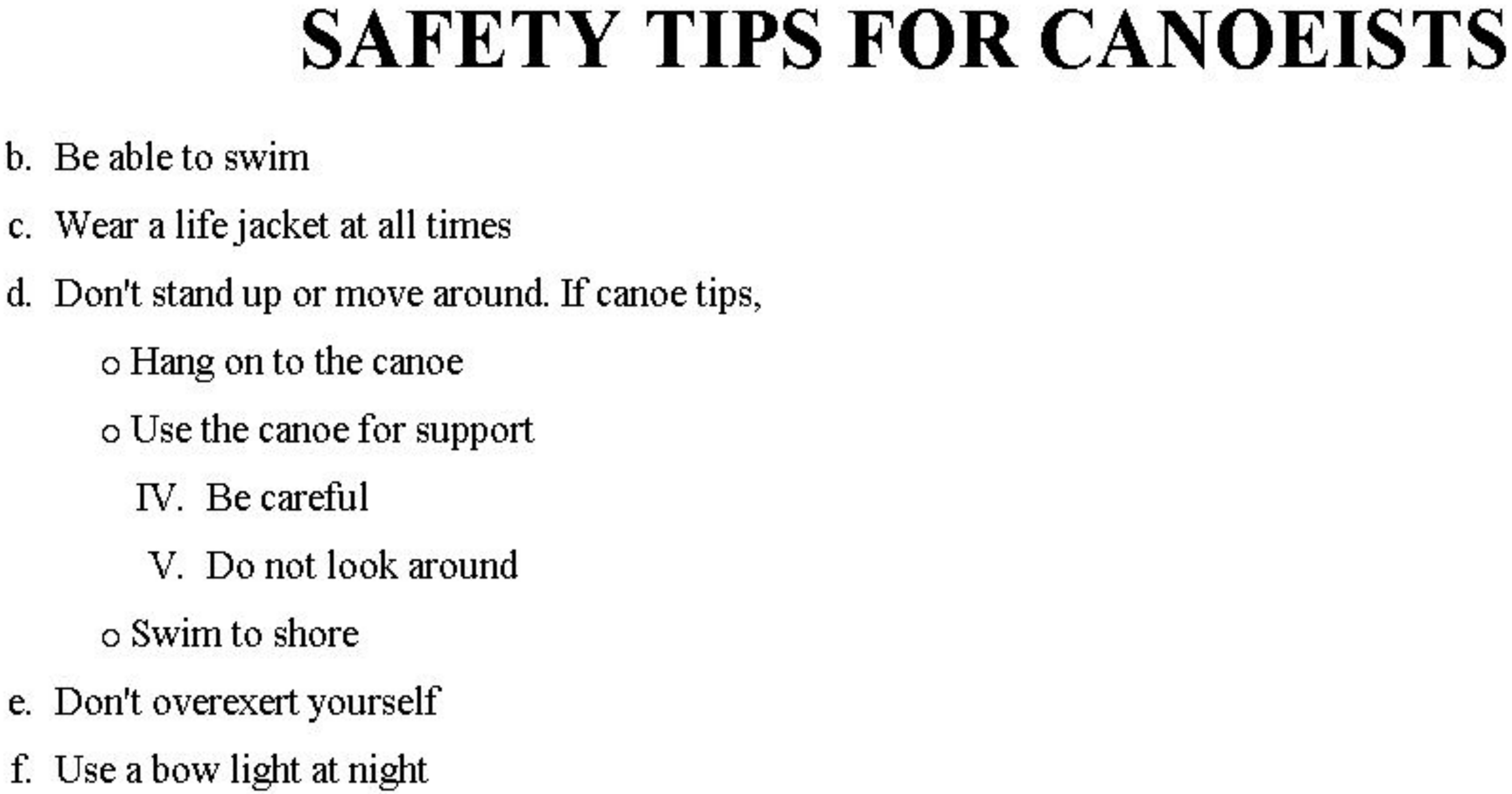
e. Don't overexert yourself

f. Use a bow light at night

```
<H1 ALIGN="CENTER">SAFETY TIPS FOR CANOEISTS</H1>
<OL TYPE="a" START="2">
<LI>Be able to swim </LI>
<LI>Wear a life jacket at all times </LI>
<LI>Don't stand up or move around. If canoe tips,
<UL>
<LI>Hang on to the canoe </LI>
<LI>Use the canoe for support
<OL type="I" start="4">
<LI> Be careful </LI>
<LI> Do not look around</LI>
</LI> </OL>
 <LI>Swim to shore
</UL> </LI>
<LI>Don't overexert yourself </LI>
<LI>Use a bow light at night </LI>
</OL>
```

What will be the output?

The output….



SAFETY TIPS FOR CANOEISTS

b. Be able to swim

c. Wear a life jacket at all times

d. Don't stand up or move around. If canoe tips,

  o Hang on to the canoe

  o Use the canoe for support

   IV. Be careful

   V. Do not look around

  o Swim to shore

e. Don't overexert yourself

f. Use a bow light at night

Images

<IMG>This element defines a graphic image on the page.

Image File (SRC:source): This value will be a URL (location of the image) E.g. http://www.domain.com/dir/file.ext or /dir/file.txt.

Alternate Text (ALT): This is a text field that describes an image or acts as a label. It is displayed when they position the cursor over a graphic image.

Alignment (ALIGN): This allows you to align the image on your page.

Images

Width (WIDTH): is the width of the image in pixels.

Height (HEIGHT): is the height of the image in pixels.

Border (BORDER): is for a border around the image, specified in pixels.

```
1) <IMG SRC=“jordan.jpg“ border=4>
2) <IMG SRC=" jordan.jpg" width="60" height="60">
3) <IMG SRC=“jordan.jpg" ALT="This is a text that goes with the image">
4) <IMG SRC=" jordan.jpg “border=20>
5) < IMG SRC =" jordan.jpg“ align="left">
```

blast blast blast blast blast

HOW TO MAKE A LINK

1) **The tags used to produce links are the <A> and </A>. The <A> tells where the link should start and the </A> indicates where the link ends. Everything between these two will work as a link.**

2) **The example below shows how to make the word Here work as a link to yahoo.**

Click <A HREF="http://www.yahoo.com">here</A> to go to yahoo.

Tables

In this chapter you will learn that tables have many uses in HTML.

Objectives:

Upon completing this section, you should be able to:

- Insert a table.
- Explain a table’s attributes.
- Edit a table.
- Add a table header.

Tables

The <TABLE></TABLE> element has four sub-elements:

Table Row<TR></TR>.

Table Header <TH></TH>.

Table Data <TD></TD>.

Caption <CAPTION></CAPTION>.

The table row elements usually contain table header elements or table data elements.

Tables

```
<table border=“1”>
<tr>
<th> Column 1 header </th>
<th> Column 2 header </th>
</tr>
<tr>
<td> Row1, Col1 </td>
<td> Row1, Col2 </td>
</tr>
<tr>
<td> Row2, Col1 </td>
<td> Row2, Col2 </td>
</tr>
</table>
```

Tables

Column 1 Header	Column 2 Header
Row1, Col1	Row1, Col2
Row2, Col1	Row2, Col2

Tables Attributes

BGColor: Some browsers support background colors in a table.

Width: you can specify the table width as an absolute number of pixels or a percentage of the document width. You can set the width for the table cells as well.

Border: You can choose a numerical value for the border width, which specifies the border in pixels.

CellSpacing: Cell Spacing represents the space between cells and is specified in pixels.

- **Cell Padding**: Cell Padding is the space between the cell border and the cell contents and is specified in pixels.
- **Align**: tables can have left, right, or center alignment.
- **Background**: Background Image

Table Caption

A table caption allows you to specify a line of text that will appear centered above or bellow the table.

```
<TABLE BORDER=1 CELLPADDING=2>
<CAPTION ALIGN="BOTTOM"> Label For My Table </CAPTION>
```

The Caption element has one attribute ALIGN that can be either TOP (Above the table) or BOTTOM (below the table).

Table Header

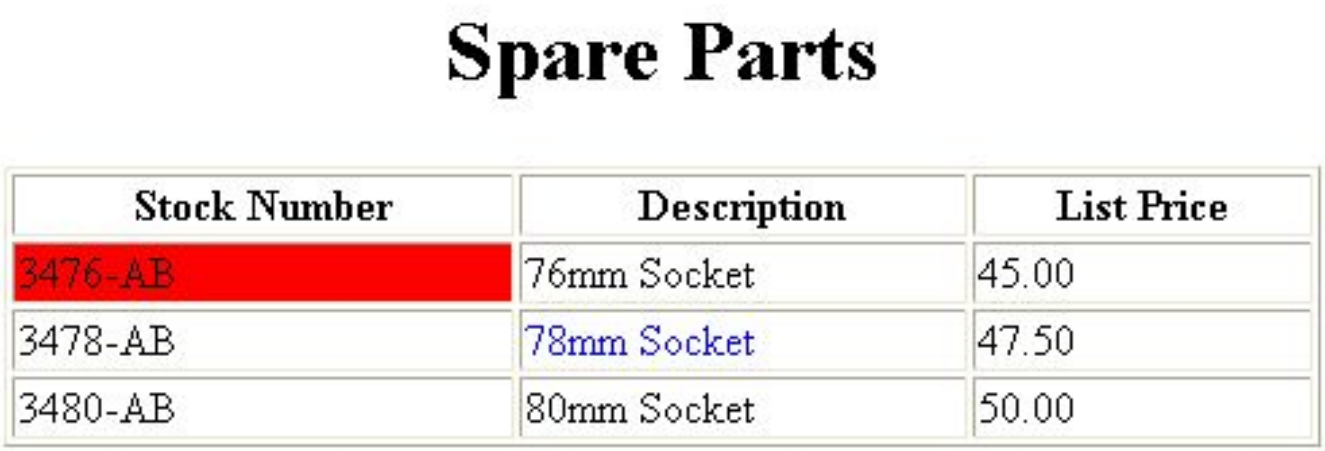
- Table Data cells are represented by the TD element. Cells can also be TH (Table Header) elements which results in the contents of the table header cells appearing centered and in bold text.

Table Data and Table Header Attributes

- **Colspan:** Specifies how many cell columns of the table this cell should span.
- **Rowspan*:*** Specifies how many cell rows of the table this cell should span.
- **Align*:*** cell data can have left, right, or center alignment.
- **Valign*:*** cell data can have top, middle, or bottom alignment.
- **Width*:*** you can specify the width as an absolute number of pixels or a percentage of the document width.
- **Height*:*** You can specify the height as an absolute number of pixels or a percentage of the document height.

Basic Table Code

```
<TABLE BORDER=1 width=50%>
<CAPTION ALIGN="TOP">  <h1>Spare Parts <h1> </Caption>
<TR><TH>Stock Number</TH><TH>Description</TH><TH>List Price</TH></TR>
<TR><TD bgcolor="red">3476-AB</TD><TD>76mm
     Socket</TD><TD>45.00</TD></TR>
<TR><TD >3478-AB</TD><TD><font color=blue>78mm Socket</font>
     </TD><TD>47.50</TD></TR>
<TR><TD>3480-AB</TD><TD>80mm Socket</TD><TD>50.00</TD></TR>
</TABLE>
```



Spare Parts

Stock Number	Description	List Price
3476-AB	76mm Socket	45.00
3478-AB	78mm Socket	47.50
3480-AB	80mm Socket	50.00

Table Data and Table Header Attributes

```
<Table border=“1” cellpadding =“2”>
<tr> <th> Column 1 Header</th> <th> Column 2 Header</th> </tr>
<tr> <td colspan=“2”> Row 1 Col 1</td> </tr>
<tr> <td rowspan=“2”>Row 2 Col 1</td>
<td> Row 2 Col2</td> </tr>
<tr> <td> Row 3 Col2</td> </tr>
</table>
```

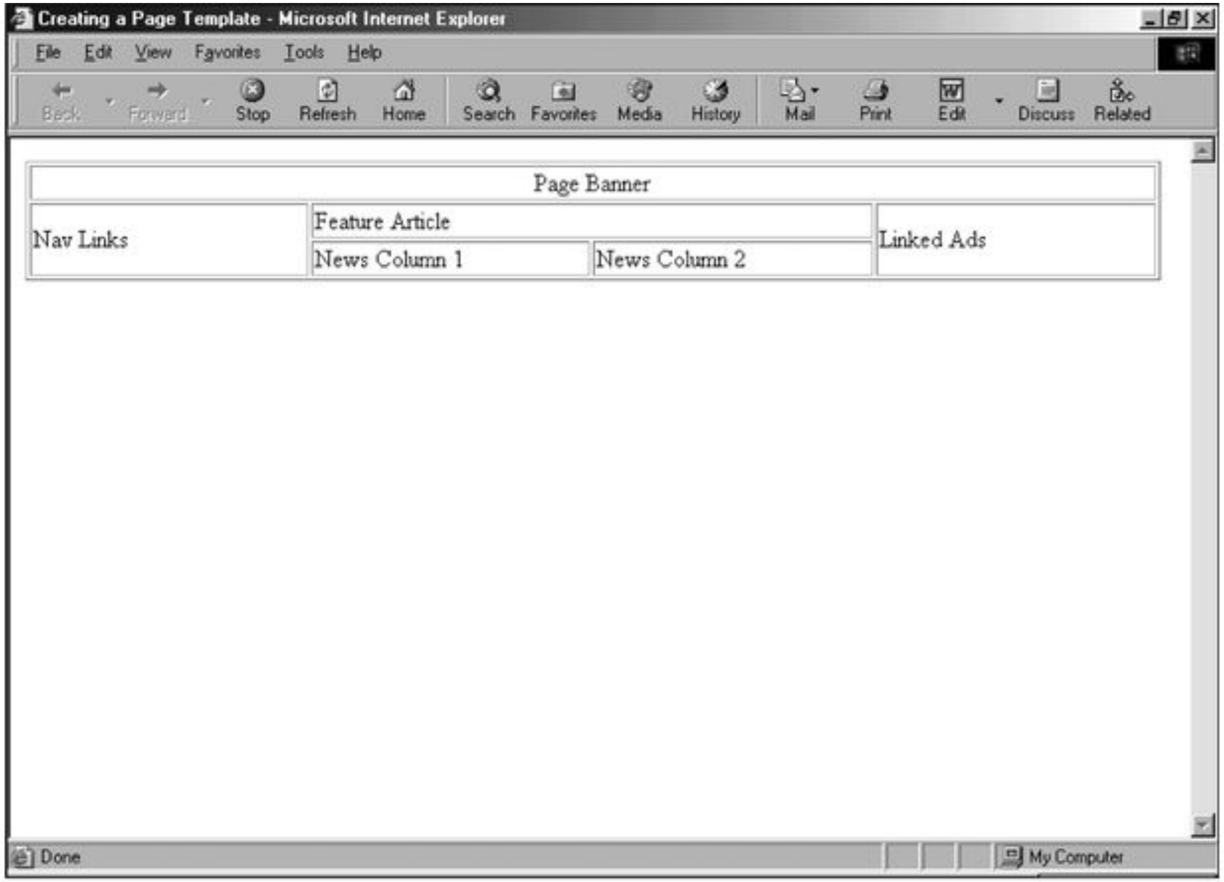
Table Data and Table Header Attributes

<table>
<tr><th>Column 1 Header</th><th>Column 2 Header</th></tr>
<tr><td colspan="2">Row 1 Col 1</td></tr>
<tr><td rowspan="2">Row 2 Col 1</td><td>Row 2 Col 2</td></tr>
<tr><td>Row 3 Col 2</td></tr>
</table>

What will be the output?

```
<TABLE BORDER width=“750”>
<TR> <TD colspan=“4” align=“center”>Page Banner</TD></TR>

<TR> <TD rowspan=“2” width=“25%”>Nav Links</TD><TD
    colspan=“2”>Feature Article</TD> <TD rowspan=“2”
    width=“25%”>Linked Ads</TD></TR>

    <TR><TD width=“25%”>News Column 1 </TD> <TD width=“25%”><News
    Column 2 </TD></TR>
</TABLE>
```

The Output



<table>
<tr><td colspan="4">Page Banner</td></tr>
<tr><td rowspan="2">Nav Links</td><td colspan="2">Feature Article</td><td rowspan="2">Linked Ads</td></tr>
<tr><td>News Column 1</td><td>News Column 2</td></tr>
</table>

Figure 5-26 Column widths set to 25%

Forms

- Forms add the ability to web pages to not only provide the person viewing the document with dynamic information but also to obtain information from the person viewing it, and process the information.

Forms

- To insert a form we use the <FORM></FORM> tags. The rest of the form elements must be inserted in between the form tags.

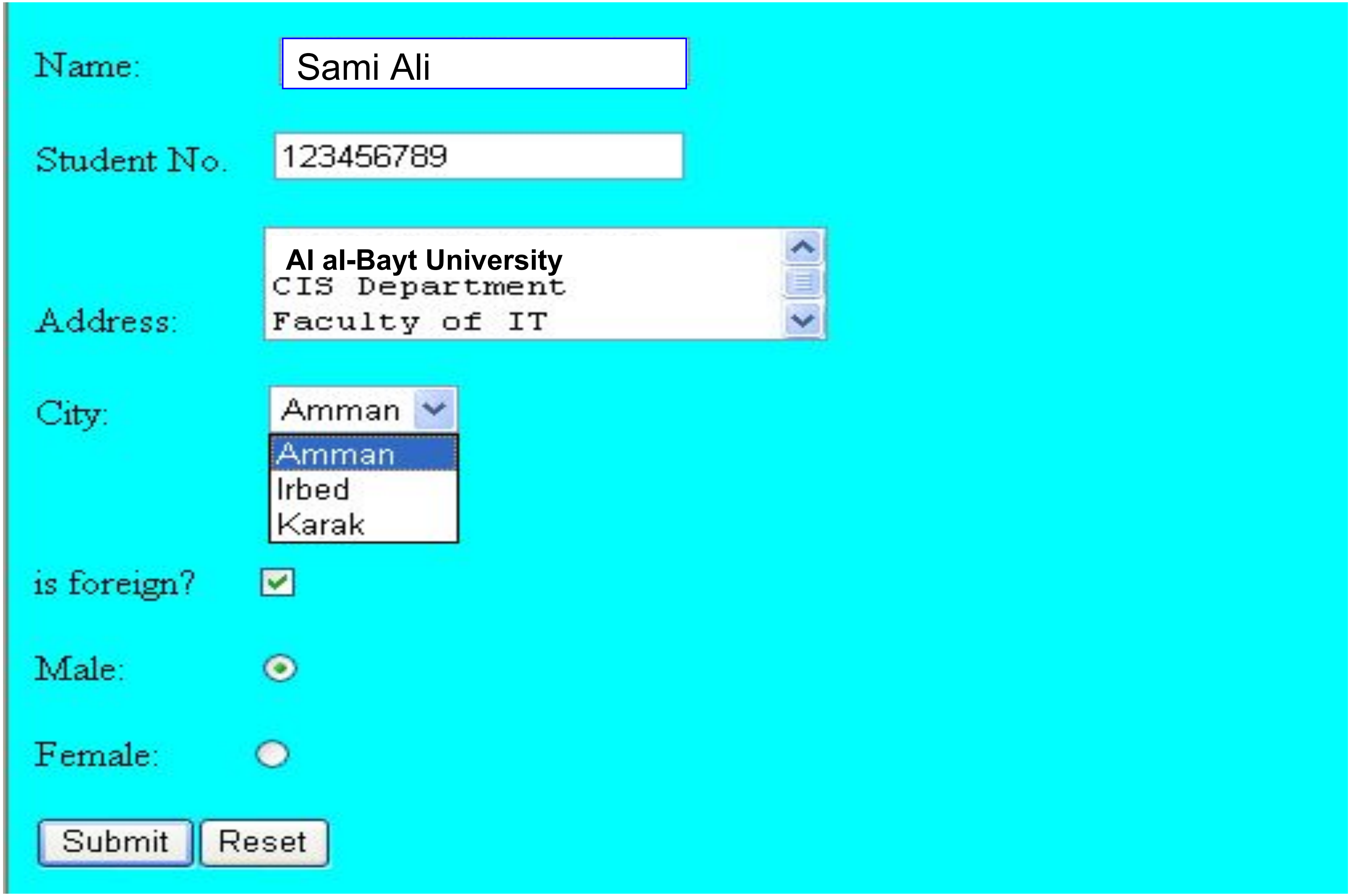
```
<HTML> <HEAD>
<TITLE> Sample Form</TITLE>
</HEAD>
<BODY BGCOLOR=“FFFFFF”>
<FORM ACTION = http://www.xnu.com/formtest.asp>
<P> First Name: <INPUT TYPE=“TEXT” NAME=“fname” MAXLENGTH=“50”> </P>
<P> <INPUT TYPE=“SUBMIT” NAME=“fsubmit1” VALUE=“Send Info”> </P>
</FORM>
</BODY> </HTML>
```

<FORM> element attributes

- **ACTION**: is the **URL** that is going to accept the data from the form, process it, and send a response back to the browser.
- **METHOD**: **GET** (default) or **POST** specifies which **HTTP** method will be used to send the form’s contents to the web server
- **NAME**: is a form name
- **TARGET**: is the target frame where the response page will show up.

Form Elements

- **Form elements have properties: Text boxes, Password boxes, Checkboxes, Option(Radio) buttons, Submit, Reset, File and Image.**
- **The properties are specified in the TYPE Attribute of the HTML element <INPUT></INPUT>.**



Name:	Sami Ali
Student No.	123456789
Address:	Al al-Bayt University CIS Department Faculty of IT
City:	Amman
is foreign?	☑
Male:	●
Female:	○

Form Elements

<INPUT> Element's Properties
TYPE= Type of INPUT entry field.
NAME = Variable name
VALUE= The data associated with the variable name
CHECKED= Button/box checked
SIZE= Number of visible characters in text field
MAXLENGHT= Maximum number of characters accepted.

Text Box

- **Text boxes*:*** Used to provide input fields for text, phone numbers, dates, etc.

<INPUT TYPE= " TEXT " >

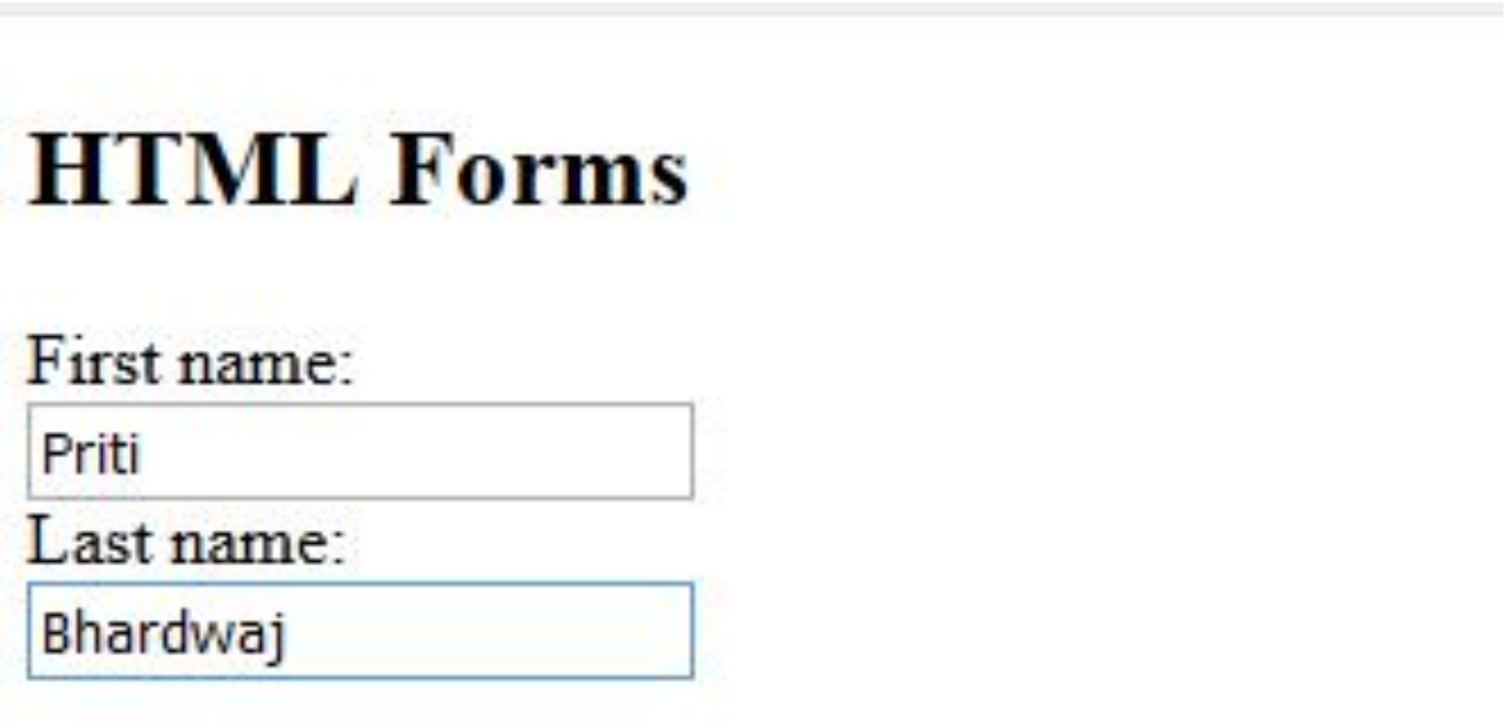
Browser will display

Textboxes use the following attributes:

- **TYPE:** text.
- **SIZE:** determines the size of the textbox in characters. **Default=20** characters.
- **MAXLENGHT *:*** determines the maximum number of characters that the field will accept.
- **NAME:** is the name of the variable
- **VALUE:** will display its contents as the default value.

Example on Text Box

```
<FORM >

First Name: <INPUT TYPE="TEXT" NAME="FName"
SIZE="15" MAXLENGTH="25"><BR>

Last Name: <INPUT TYPE="TEXT" NAME="LName"
SIZE="15" MAXLENGTH="25"><BR>
</FORM>
```



HTML Forms

First name:

Priti

Last name:

Bhardwaj

Password

- **Password:** Used to allow entry of passwords.

<INPUT TYPE= " PASSWORD " >

Browser will display

Text typed in a password box is starred out in the browser display.

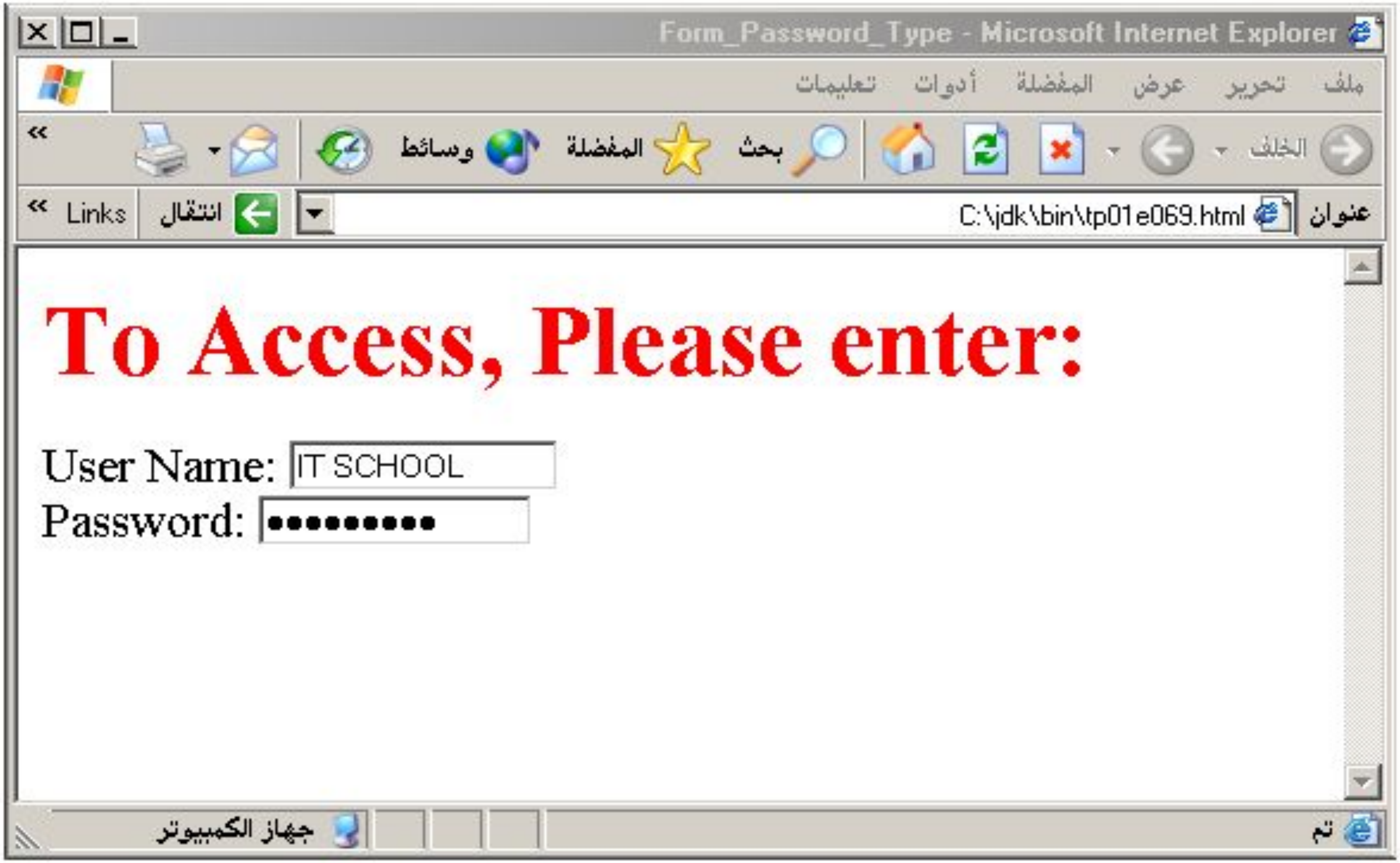
Password boxes use the following attributes:

- **TYPE:** password.
- **SIZE:** determines the size of the textbox in characters.
- **MAXLENGHT:** determines the maximum size of the password in characters.
- **NAME:** is the name of the variable
- **VALUE:** is usually blank.

Example on Password Box

```
<FORM >
User Name: <INPUT TYPE="TEXT" Name="FName"
SIZE="15" MAXLENGTH="25"><BR>

Password: <INPUT TYPE="PASSWORD NAME=“PWord"
    value="" SIZE="15” MAXLENGTH="25"><BR>
</FORM>
```

Output


Form_Password_Type - Microsoft Internet Explorer
ملف تحرير عرض المفضلة أدوات تعليمات
الخلف بحث المفضلة وسائط
عنوان C:\jdk\bin\tp01e069.html
انتقال Links
تم
جهاز الكمبيوتر

To Access, Please enter:

User Name: IT SCHOOL

Password: ●●●●●●●●●●

Check Box

- **Check Box:** Check boxes allow the users to select more than one option.

<INPUT TYPE=“CHECKBOX”>

Browser will display

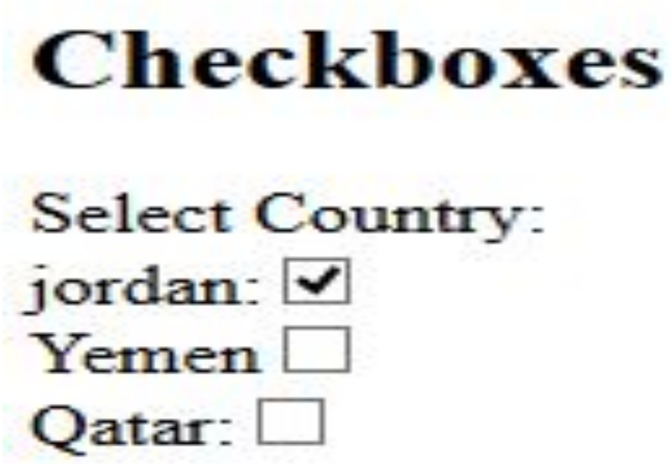
Checkboxes have the following attributes:

- **TYPE:** checkbox.
- **CHECKED:** is blank or CHECKED as the initial

status.

- **NAME*:*** is the name of the variable
- **VALUE:** is usually set to a value.

```
<html>
<body>

<h2>Checkboxes</h2>
<form>
Select Country:<BR>
jordan:<INPUT TYPE="Checkbox" Name="country1"
CHECKED><BR>
Yemen<INPUT TYPE="Checkbox"
Name="country2"><BR>
Qatar:<INPUT TYPE="Checkbox"
Name="country3"><BR> <BR>
</form>
</body>
</html>
```



Checkboxes

Select Country:

jordan: ☑

Yemen □

Qatar: □

Radio Button

- **Radio Button:** Radio buttons allow the users to select only one option.

<INPUT TYPE="RADIO">

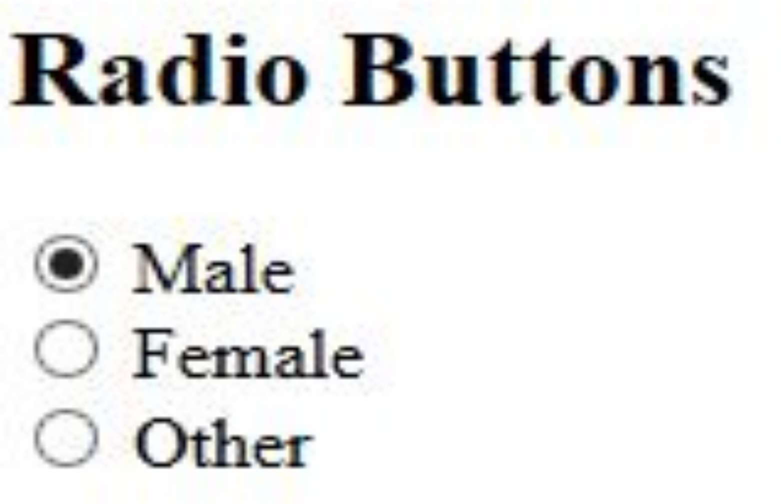
Browser will display

Radio buttons have the following attributes:

- **TYPE:** radio.
- **CHECKED:** is blank or CHECKED as the initial status. Only one radio button can be checked
- **NAME:** is the name of the variable
- **VALUE:** usually has a set value.

Example

```
<html>
<body>
<h2>Radio Buttons</h2>
<form>
  <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<br><br>
 </form>
</body>
</html>
```



Radio Buttons

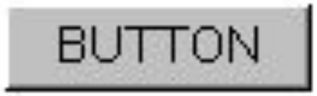
● Male
○ Female
○ Other

Button

- **Button:** To define a button

<INPUT TYPE=“BUTTON”>

Browser will display



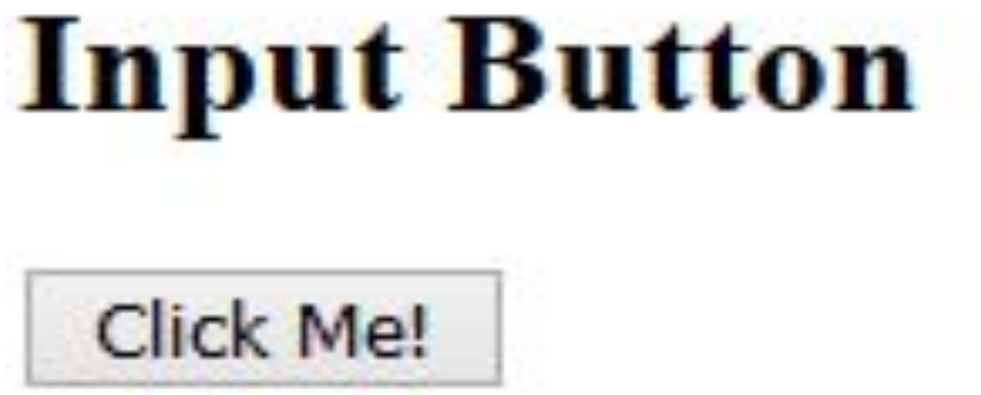
Button has the following attributes:

- **TYPE:** button.
- **NAME:** is the name of the button to be used

in scripting.

- **VALUE:** determines the text label on the button.

```html
<html>
<body>

<h2>Input Button</h2>

<input type="button" value="Click Me!">

</body>
</html>
```



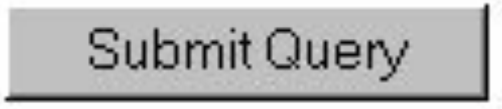
Input Button

Submit Button

- **Submit:** Every set of Form tags requires a Submit button. This is the element causes the browser to send the names and values of the other elements specified by the ACTION attribute of the FORM element.

<INPUT TYPE=“SUBMIT”>

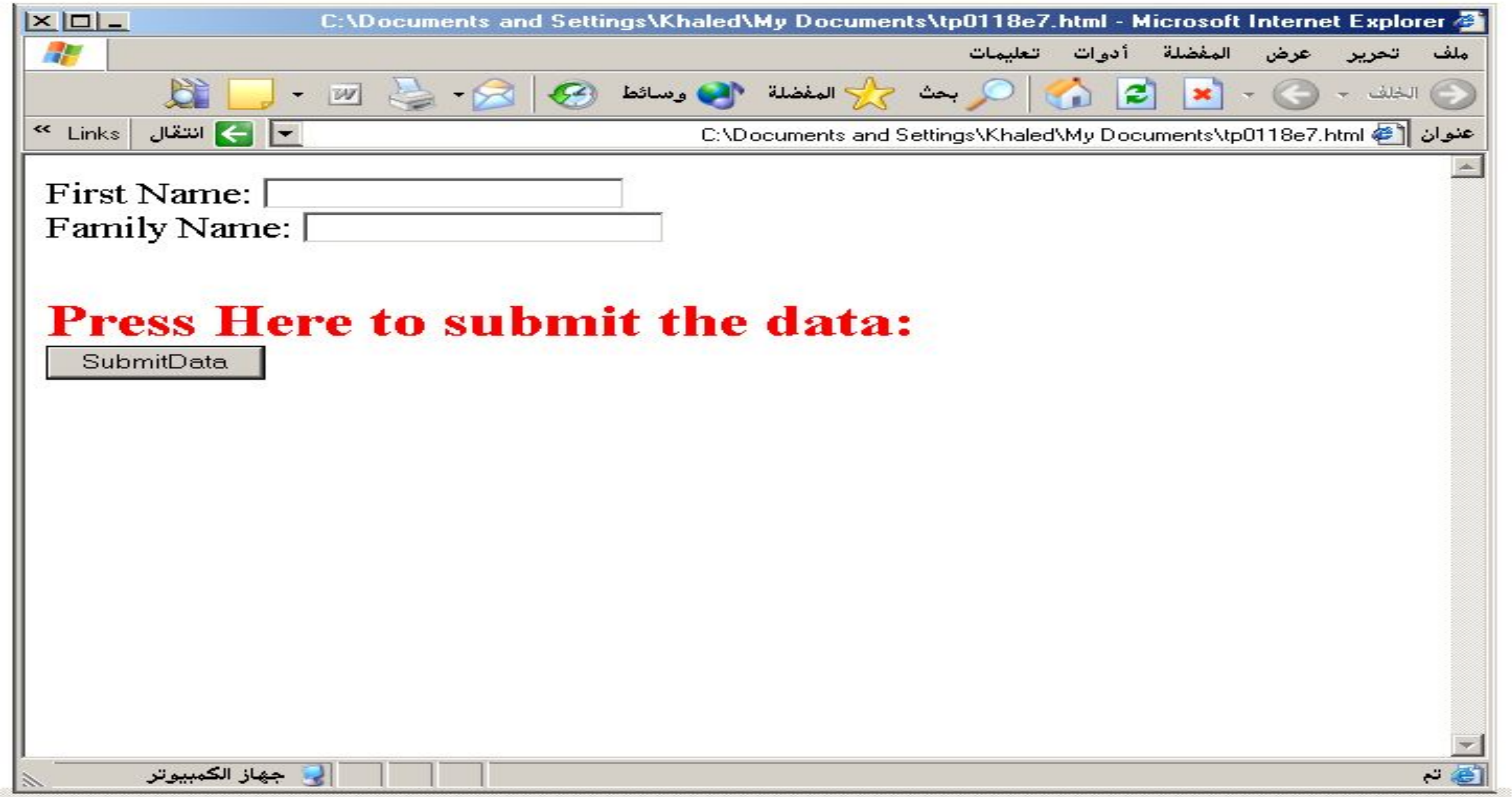
The browser will display



Submit has the following attributes

- **TYPE:** submit.
- **NAME**: value used for processing.
- **VALUE:** determines the text label on the button, usually Submit Query.

```
<FORM    Action=“/action_page.php"
method="get">
First Name: <INPUT TYPE="TEXT" Size=25
name="firstName"><BR>
Family Name: <INPUT TYPE="TEXT" Size=25
name="LastName"><BR>
<BR>
<FONT Color=red>
Press Here to submit the data:<BR>
<INPUT TYPE="submit" VALUE="SubmitData " >
</FORM>
```


C:\Documents and Settings\Khaled\My Documents\tp0118e7.html - Microsoft Internet Explorer
ملف تحرير عرض المفضلة أدوات تعليمات
الخلف
بحث
المفضلة
وسائط
عنوان
C:\Documents and Settings\Khaled\My Documents\tp0118e7.html
انتقال
Links
تم
جهاز الكمبيوتر

First Name:

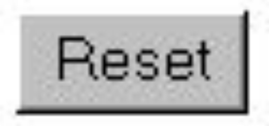
Family Name:

Press Here to submit the data:

SubmitData

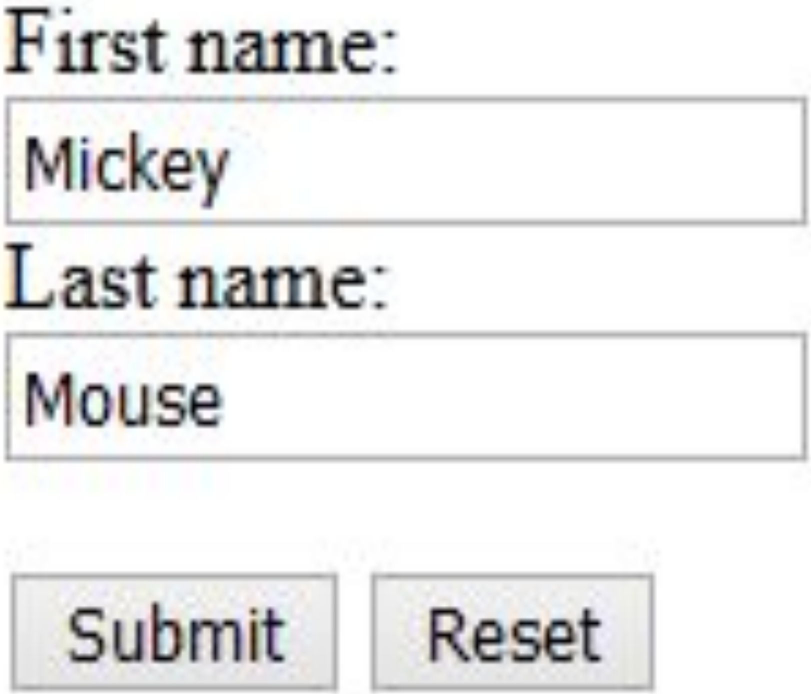
Reset Button

- **Reset:** It is a good idea to include one of these for each form where users are entering data. It allows the surfer to clear all the input in the form.
- **<INPUT TYPE=“RESET”>**
- Browser will display



-
- Reset buttons have the following attributes:
- **TYPE**: reset.
- **VALUE**: determines the text label on the button, usually Reset.

```
<form action="/action_page.php">
First name:<br>
<input type="text" name="firstname"
value="Mickey">
<br>
Last name:<br>
<input type="text" name="lastname"
value="Mouse">
<br><br>
<input type="submit" value="Submit">
<input type="reset">
</form>
```



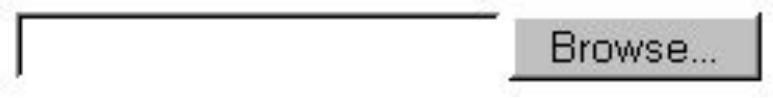
First name:

Mickey

Last name:

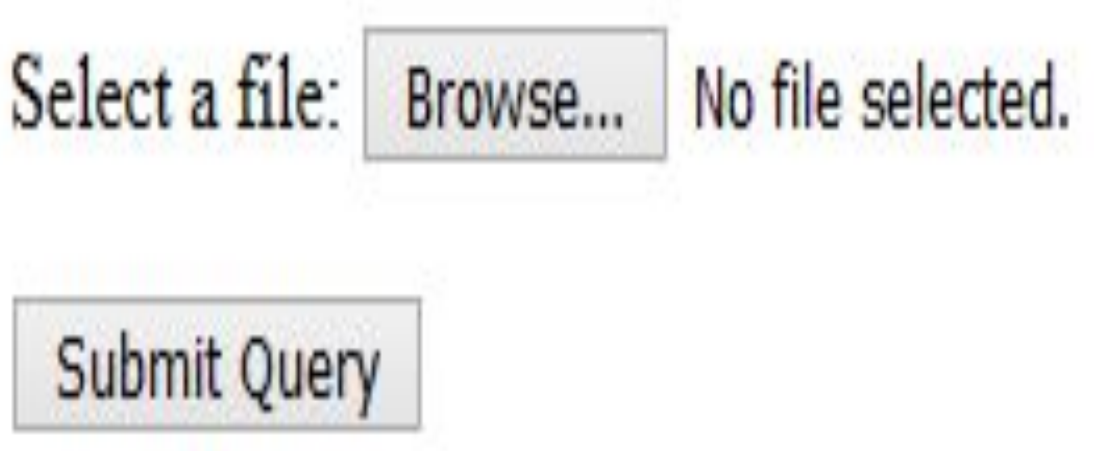
Mouse

File

- **File Upload**: You can use a file upload to allow surfers to upload files to your web server.
- **<INPUT TYPE=“FILE”>**
- Browser will display 

- File Upload has the following attributes:
- **TYPE**: file.
- **SIZE**: is the size of the text box in characters.
- **NAME**: is the name of the variable
- **MAXLENGHT**: is the maximum size of the input in the textbox in characters.

```
<form>
  Select a file: <input type="file"
    name="myFile"><br><br>
  <input type="submit">
</form>
```

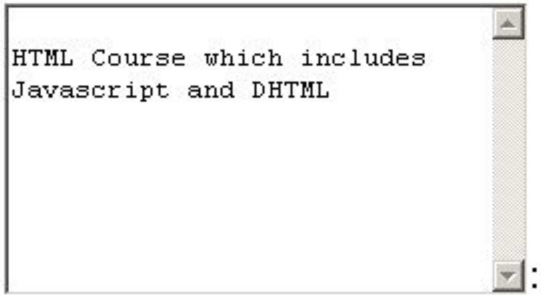


Select a file: No file selected.

TEXTAREA

- **<TEXTAREA></TEXTAREA>:** is an element that allows for free form text entry.

Browser will display



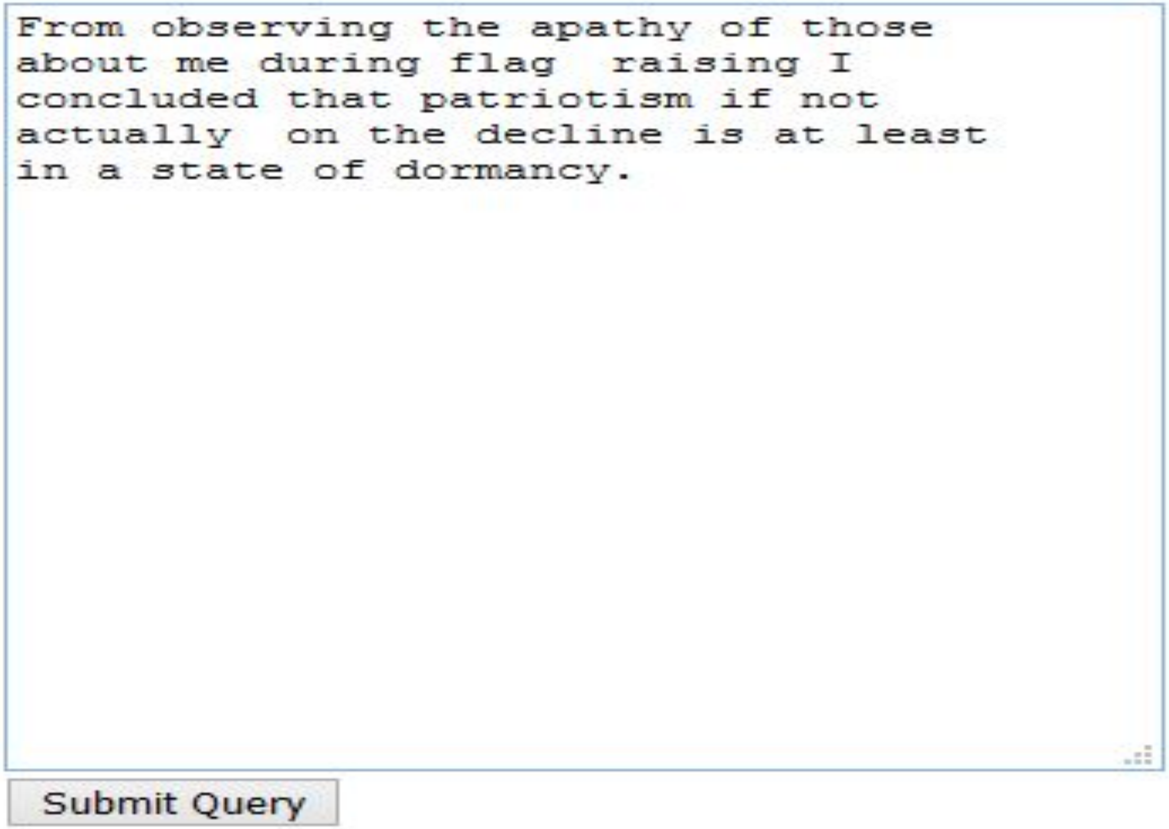
```
HTML Course which includes
Javascript and DHTML
```

:

Textarea has the following attributes:

- **NAME**: is the name of the variable
- **ROWS**: the number of rows to the textbox.
- **COLS**: the number of columns to the textbox.

```
<form>
 <TEXTAREA   COLS=40  ROWS=20
Name="comments"  >
From observing the apathy of those
about me during flag  raising I
concluded that patriotism if not
actually  on the decline is at least
in a state of dormancy.
</TEXTAREA>
<br>
<input type="submit">
</form>
```



From observing the apathy of those
about me during flag raising I
concluded that patriotism if not
actually on the decline is at least
in a state of dormancy.

Other Elements used in Forms

- The two following examples are **<SELECT></SELECT>** elements, where the attributes are set differently.

The Select elements attributes are:

- **NAME**: is the name of the variable
- **SIZE**: this sets the number of **visible** choices.
- **MULTIPLE**: the presence of this attribute signifies that the user can make multiple selections. By default only one selection is allowed.

Other Elements used in Forms

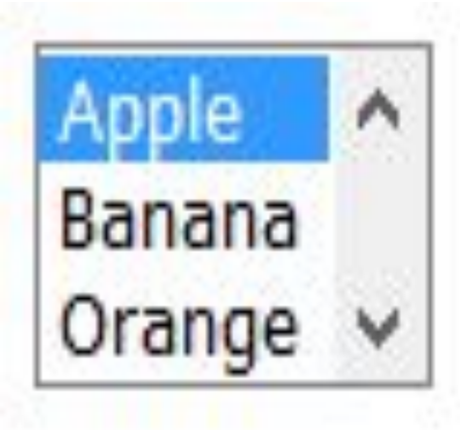
- **Option**

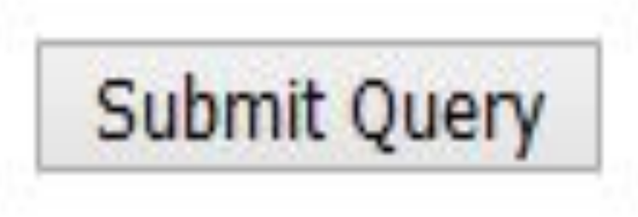
The list items are added to the **<SELECT>** element by inserting **<OPTION></OPTION>** elements.

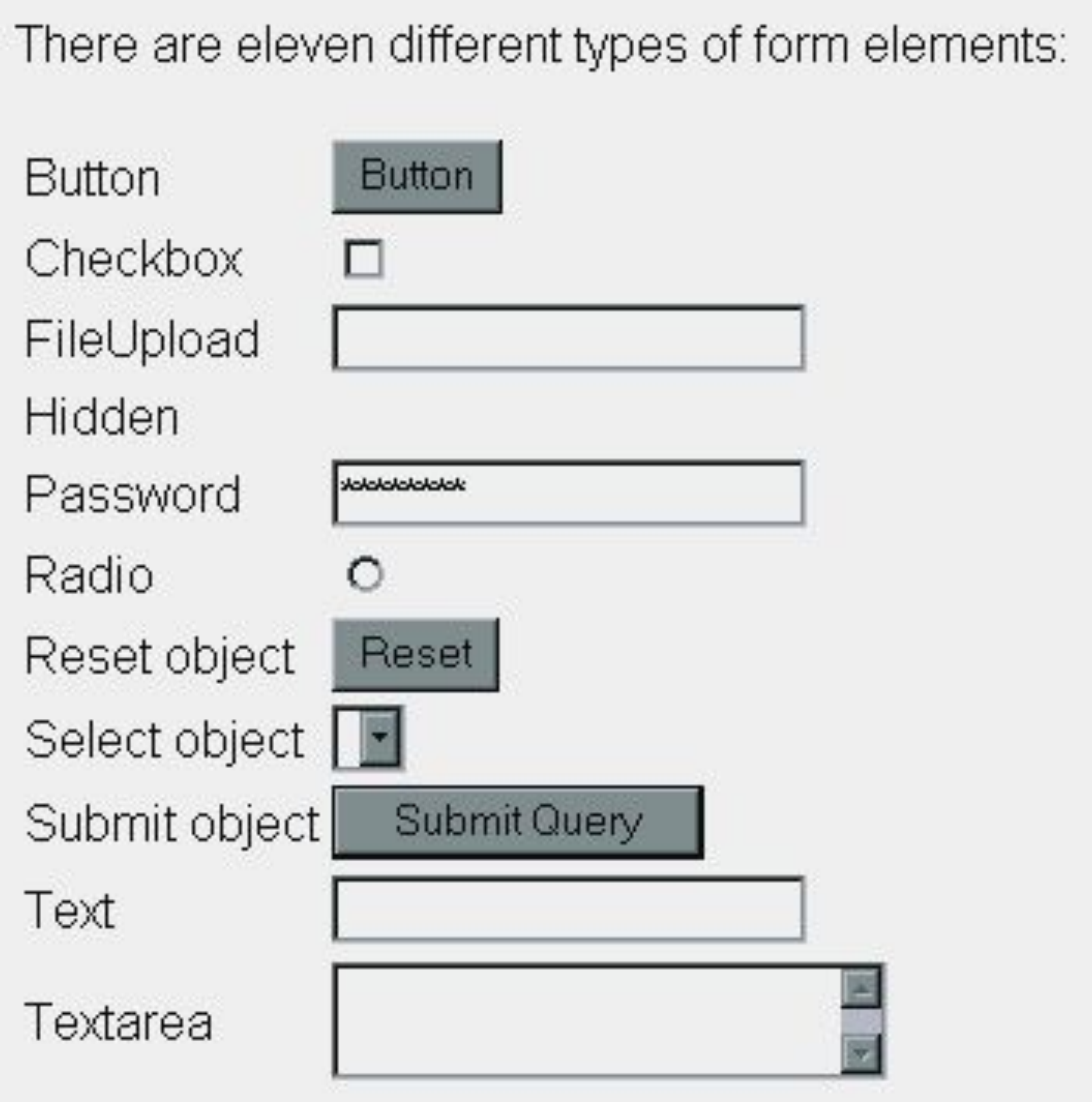
The Option Element’s attributes are:

- **SELECTED**: When this attribute is present, the option is selected when the document is initially loaded. **It is an error for more than one option to be selected.**
- **VALUE**: Specifies the value the variable named in the select element.

```
<form>
  <select name=“fruits" size="3">
    <option value=“apple“
selected>Apple</option>
    <option value=“banana">Banana</option>
    <option value=“orange">Orange</option>
    <option value=“kiwi">Kiwi</option>
  </select>
  <br><br>
  <input type="submit">
</form>
```


Apple
Banana
Orange


Submit Query



There are eleven different types of form elements:

Button	Button
Checkbox	□
FileUpload	
Hidden	
Password	**********
Radio	○
Reset object	Reset
Select object	▾
Submit object	Submit Query
Text	
Textarea	

iframe

- An HTML iframe is used to display a web page within a web page.
- The HTML <iframe> tag specifies an inline frame.

- An inline frame is used to embed another document within the current HTML document.

```
<iframe src="url" title="description"></iframe>
```

- Always include a title attribute for the <iframe>. This is used by screen readers to read out what the content of the iframe is.
- Use the height and width attributes to specify the size of the iframe.
- An iframe can be used as the target frame for a link.

- The target attribute of the link must refer to the name attribute of the iframe

Example

- <!DOCTYPE html>
- <html>
- <body>
- <h2>HTML Iframes</h2>
- <p>Just see the iframe</p>

- <iframe src=“first_iframe.html" name=“iframe_a” height="200" width="300" title="Iframe Example"></iframe>
- <p><a href="https://wwwgoogle.com" target="iframe_a">click</a></p>

- <p>When the target attribute of a link matches the name of an iframe, the link will open in the iframe.</p>
- </body>
- </html>

Some more tags

Tag& Description

- <article> Defines an article
- <aside> Defines content aside from the page content
- <details> Defines additional details that the user can view or hide
- <figcaption> Defines a caption for a <figure> element
- <figure> Specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
- <footer> Defines a footer for a document or section
- <header> Specifies a header for a document or section
- <main> Specifies the main content of a document
- <mark> Defines marked/highlighted text
- <nav> Defines navigation links
- <section> Defines a section in a document
- <summary> Defines a visible heading for a <details> element
- <time> Defines a date/time

<section> Element

- defines a section in a document.
- According to W3C's HTML5 documentation: "A section is a thematic grouping of content, typically with a heading."
- A Web site's home page could be split into sections for introduction, content, and contact information.

- <!DOCTYPE html>
- <html>
- <body>
- <section>
- <h1>Festivals of INDIA</h1>
- <p>
- Being a highly spiritual country, festivals are at the heart of people’s lives in India. The numerous and varied festivals that are held throughout the year offer a unique way of seeing Indian culture at its best.</p>
- </section>

- <section>
- <h1>DIWALI</h1>
- <p>
- Diwali is a five day festival that represents the start of the Hindu New Year. It's known as the "Festival of Lights" for all the fireworks, small clay lamps, and candles that are lit during the celebrations.</p>
- </section>

- </body>
- </html>

<article> Element

- The <article> element specifies independent, self-contained content.
- An article should make sense on its own, and it should be possible to read it independently from the rest of the web site. (complete, self-contained block of related elements.)
- Examples of where an <article> element can be used:

1. Forum post
2. Blog post
3. Newspaper article

- <article>
-   <h1>What Does WWF Do?</h1>
-   <p>WWF's mission is to stop the degradation of our planet's natural environment, and build a future in which humans live in harmony with nature.</p>
- </article>

<header> Element

- specifies a header for a document or section.
- The <header> element should be used as a container for introductory content.
- You can have several <header> elements in one document.

Example code

- <article>
- <header>
- <h1>What Does WWF Do?</h1>
- <p>WWF's mission:</p>
- </header>
- <p>WWF's mission is to stop the degradation of our planet's natural environment, and build a future in which humans live in harmony with nature.</p>
- </article>

output

What Does WWF Do?

WWF's mission:

WWF's mission is to stop the degradation of our planet's natural environment, and build a future in which humans live in harmony with nature.

<footer> Element

- element specifies a footer for a document or section.
- A <footer> element should contain information about its containing element.
- A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.

- Example code

- <footer>
-   <p>Posted by: Hege Refsnes</p>
-   <p>Contact information: <a href="mailto:someone@example.com">
-   someone@example.com</a>.</p>
- </footer>

output

- Posted by: Hege Refsnes
- Contact information: someone@example.com.

<nav> Element

- The <nav> element defines a set of navigation links.
- The <nav> element is intended for large blocks of navigation links. However, not all links in a document should be inside a <nav> element!

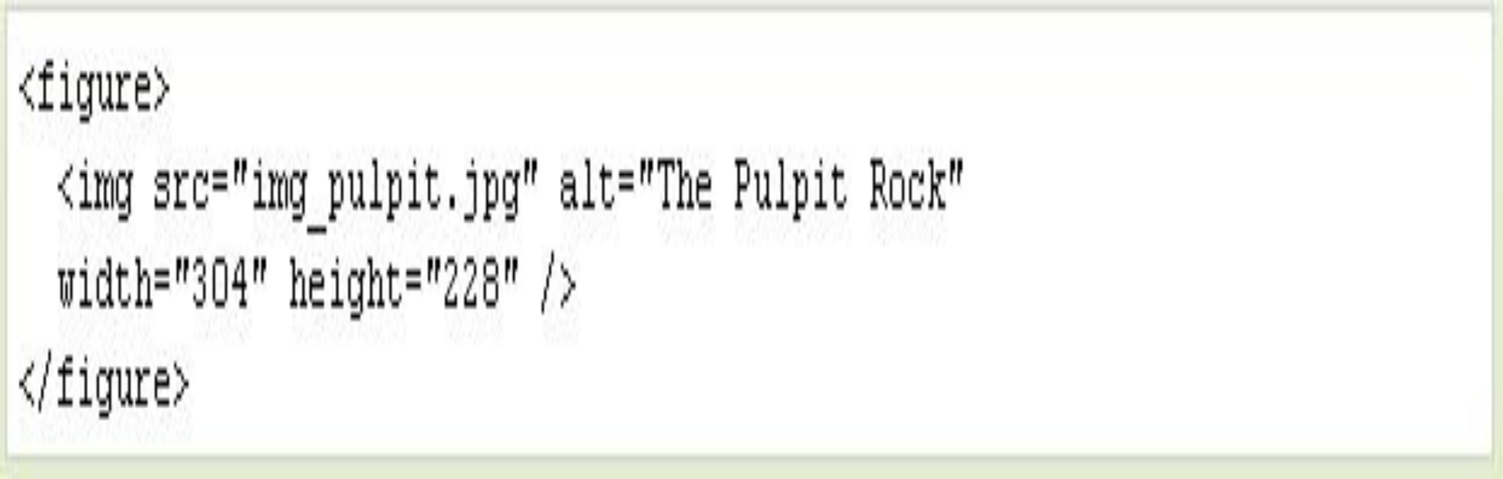
- <nav>
- <a href="/html/">HTML</a> |
- <a href="/css/">CSS</a> |
- <a href="/js/">JavaScript</a> |
- <a href="/jquery/">jQuery</a>
- </nav>

<figure> and <figcaption> Elements

- In books and newspapers, it is common to have captions with images.
- The purpose of a caption is to add a visual explanation to an image.
- With HTML5, images and captions can be grouped together in **<figure>** elements:
- The **<img>** element defines the image,
  the **<figcaption>** element defines the caption.

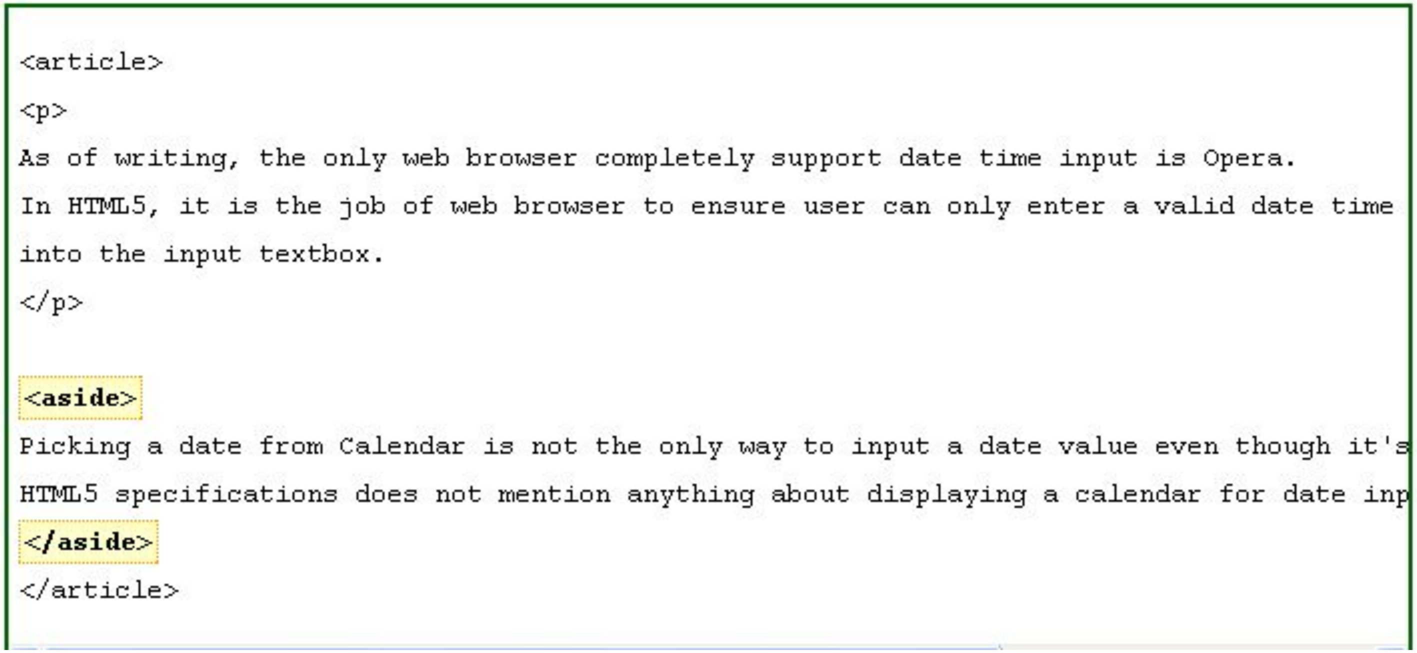
<figure>: The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

While the content of the <figure> element is related to the main flow, its position is independent of the main flow, and if removed it should not affect the flow of the document



```
<figure>
  <img src="img_pulpit.jpg" alt="The Pulpit Rock"
  width="304" height="228" />
</figure>
```

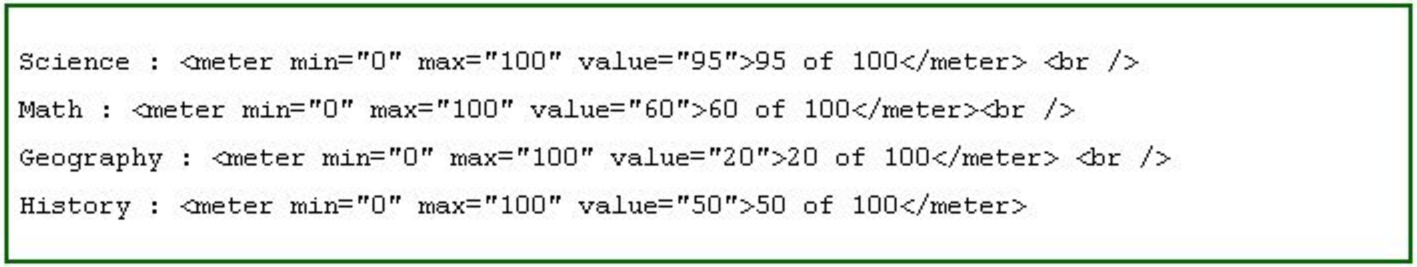
<aside>: The "aside" element is a section that somehow related to main content, but it can be separate from that content



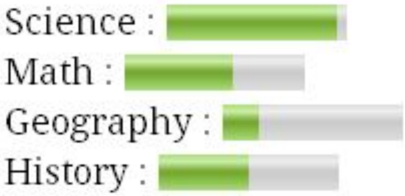
```
<article>
<p>
As of writing, the only web browser completely support date time input is Opera.
In HTML5, it is the job of web browser to ensure user can only enter a valid date time
into the input textbox.
</p>

<aside>
Picking a date from Calendar is not the only way to input a date value even though it's
HTML5 specifications does not mention anything about displaying a calendar for date inp
</aside>
</article>
```

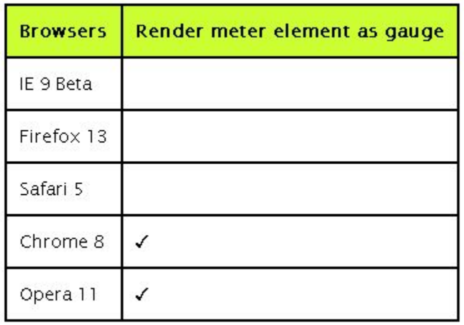
<meter>: "Meter" is a new element in HTML5 which represenet value of a known range as a gauge. The keyword here is "known range". That means, you are only allowed to use it when you are clearly aware of its minimum value and maximum value.

One example is score of rating. I would rate this movie <meter min="0" max="10" value="8">8 of 10</meter>.



```
Science : <meter min="0" max="100" value="95">95 of 100</meter> <br />
Math : <meter min="0" max="100" value="60">60 of 100</meter><br />
Geography : <meter min="0" max="100" value="20">20 of 100</meter> <br />
History : <meter min="0" max="100" value="50">50 of 100</meter>
```

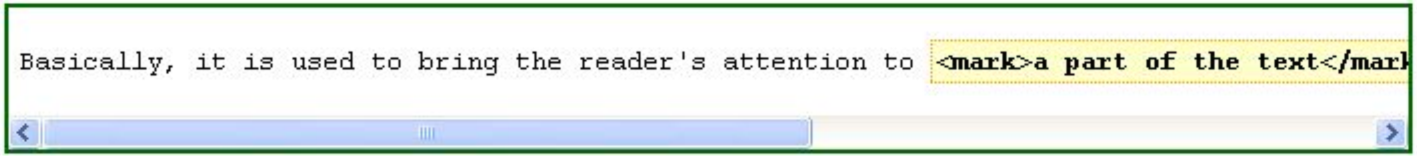




Browsers	Render meter element as gauge
IE 9 Beta	
Firefox 13	
Safari 5	
Chrome 8	✓
Opera 11	✓

<mark>: The mark <mark> element represents a run of text in one document marked or highlighted for reference purposes, due to its relevance in another context.

Basically, it is used to bring the reader's attention to a part of the text that might not have been



```
Basically, it is used to bring the reader's attention to <mark>a part of the text</mark
```

HTML Forms new tags

New Input Types:

<input type="">

- Color: Gives the end user a native color picker to choose a color.
- Date: Offers a datepicker.
- Datetime: An element to select a date and time (with time zone).
- Email: A field for entering e-mail address(es).

- Month: Choose a full month.
- Number: Picking a number.
- Range: Offers a slider to set to a certain value/position.
- Tel: Choosing a telephone number.
- Time: Input a certain time or select a time (no time zone).
- Week: Picking a specific week.

New Attributes

- Autocomplete: Possible values are “on” and “off”. The autocomplete attribute specifies whether a form or input field should have autocomplete on or off. When autocomplete is on, the browser automatically complete values based on values that the user has entered before.

- Autofocus: When present, it specifies that an <input> element should automatically get focus when the page loads. Only one form element can have autofocus in a given page.

- Formaction: This attribute specifies the URL of a file that will process the input control when the form is submitted. The formaction attribute overrides the action attribute of the <form> element; for instance if different buttons should submit the form to different URLs
- **Note:** The formaction attribute is used with type="submit" and type="image“

- Formmethod: This attribute defines the HTTP method for sending form-data to the action URL.The formmethod attribute overrides the method attribute of the <form> element, in case a button should change the method.
- **Note:** The formmethod attribute can be used with type="submit" and type="image".

- List: To connect with a <datalist> element by its id, to use its <option> elements as suggestions. They fulfill a common requirement: a text field with a set of predefined autocomplete options.
- Max: Maximum value for the value that can be put in.
- Min: Minimum value for the value that can be put in.

- Multiple: Allows for selection of multiple files for <input type=”file”> elements, and for multiple e-mail addresses separated by a comma.

- Readonly: If a field should be readonly. The readonly attribute is similar to the disabled attribute: it makes it impossible

  for the user to edit the form field. Unlike disabled, however, the field *can* receive focus, and its value is submitted with the form.

- Step: Possibility to control the size of each step for elements that accepts number or date input.
- The step attribute specifies the legal number intervals for an <input> element.
- Example: if step="3", legal numbers could be -3, 0, 3, 6, etc.

required Attribute

The Boolean required attribute tells the browser to only submit the form if the field in question is filled out correctly. Obviously, this means that the field can’t be left empty, but it also means that, depending on other attributes or the field’s type, only certain types of values will be accepted.

If a required field is empty or invalid, the form will fail to submit, and focus will move to the first invalid form element.

The required attribute can be set on any input type except button, range, color, and hidden, all of which generally have a default value.

placeholder Attribute

The placeholder attribute allows a short hint to be displayed inside the form element, space permitting, telling the user what data should be entered in that field.

The placeholder text disappears when the field gains focus, and reappears on blur if no data was entered.

New Elements

- Datalist: Contains a number of <option> elements with values that can be used as suggestions for other form elements through the usage of the list attribute on them.
- The <datalist> tag specifies a list of pre-defined options for an <input> element.
- The <datalist> tag is used to provide an "autocomplete" feature on <input> elements. Users will see a drop-down list of pre-defined options as they input data.
- Use the <input> element's list attribute to bind it together with a <datalist> element.

- Meter: The meter element is for displaying values on a bar, where you can custom control min, max and assigned value. You can also specify low, high and optimum to set up different kind of areas of the bar.
- Progress: Meant to be used to indicate progress of any kind in a web page, for instance file upload progress.

datalist

```
<!DOCTYPE html>
<html>
<body>

<form action="" method="get">
  <input list="browsers" name="browser">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
  <input type="submit">
</form>

<p><strong>Note:</strong> The datalist tag is not supported in Internet Explorer 9 and earlier versions, or in Safari.</p>

</body>
</html>
```

HTML audio & video

- What is Multimedia?
- Multimedia comes in many different formats.
- Examples: Pictures, music, sound, videos, records, films, animations, and more.
- Web pages often contains multimedia elements of different types and formats.

Browser Support

- The first web browsers had support for text only, limited to a single font in a single color.
- Later came browsers with support for colors and fonts, and even support for pictures!
- The support for sounds, animations, and videos is handled differently by various browsers. Different types and formats are supported, and some formats requires extra helper programs (plug-ins) to work.

Multimedia Formats

- Multimedia elements (like sounds or videos) are stored in media files.
- The most common way to discover the type of a file, is to look at the file extension. When a browser sees the file extension .htm or .html, it will treat the file as an HTML file. The .xml extension indicates an XML file, and the .css extension indicates a style sheet file. Pictures are recognized by extensions like .gif, .png and .jpg.
- Multimedia files also have their own formats and different extensions like: .swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi.

MP4 is the new and upcoming format for internet video.

MP4 is recommended by YouTube.

MP4 is supported by Flash Players

MP4 is supported by HTML5.

- Ogg:- ogg Theora Ogg. Developed by the Xiph.Org Foundation. Supported by HTML5.
- MPEG-4 or MP4:- mp4MP4. Developed by the Moving Pictures Expert Group. Based on QuickTime. Commonly used in newer video cameras and TV hardware. Supported by all HTML5 browsers. Recommended by YouTube.
- WebM: webm WebM. Developed by the web giants, Mozilla, Opera, Adobe, and Google. Supported by HTML5.
- Only MP4, WebM, and Ogg video is supported by the newest HTML5 standard

- Only MP3, WAV, and Ogg audio is supported by the newest HTML5 standard.
- WAV: wav WAV. Developed by IBM and Microsoft. Plays well on Windows, Macintosh, and Linux operating systems. Supported by HTML5.
- MP3: mp3 MP3 files are actually the sound part of MPEG files. MP3 is the most popular format for music players. Combines good compression (small files) with high quality. Supported by all browsers.

Playing Videos in HTML

- Before HTML5, there was no standard for showing videos on a web page.
- Before HTML5, videos could only be played with a plug-in (like flash).
- The HTML5 <video> element specifies a standard way to embed a video in a web page.
-

- With the help of video tag

- <!DOCTYPE html>
- <html>
- <body>
- <video width="320" height="240" controls>
-   <source src="movie.mp4" type="video/mp4">
-   <source src="movie.ogg" type="video/ogg">
-   Your browser does not support the video tag.
- </video>
- </body>
- </html>

- The **controls** attribute adds video controls, like play, pause, and volume.
- It is a good idea to always include **width** and **height** attributes.
- If height and width are not set, the browser does not know the size of the video. The effect will be that the page will change (or flicker) while the video loads.
- Text between the <video> and </video> tags will only display in browsers that do not support the <video> element.
- Multiple **<source>** elements can link to different video files. The browser will use the first recognized format.
-

- To start a video automatically use the **auto play** attribute:
- The autoplay attribute does not work in Safari and Opera, or in mobile devices like iPad and iPhone.

Audio on the Web

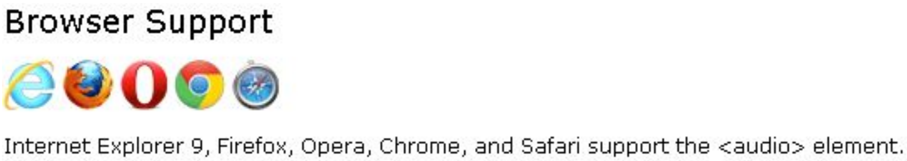
- The HTML5 <audio> element specifies a standard way to embed audio in a web page.

<audio controls>

- <source src="horse.ogg" type="audio/ogg">
- <source src="horse.mp3" type="audio/mpeg">
- Your browser does not support the audio element.
- </audio>

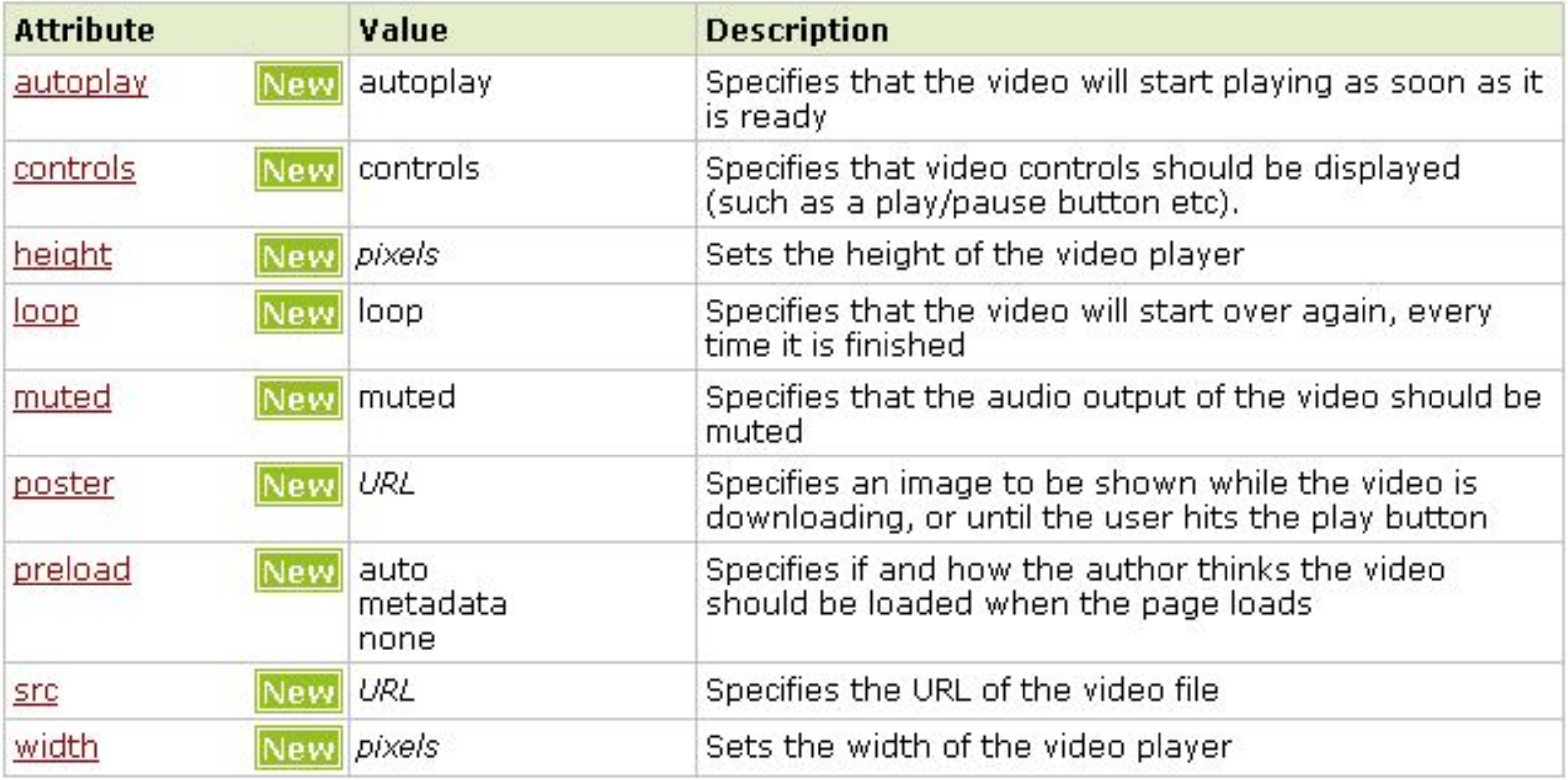
4.3. new html5 element

<video attributes>



Browser Support

Internet Explorer 9, Firefox, Opera, Chrome, and Safari support the <audio> element.



Attribute	Value	Description
autoplay New	autoplay	Specifies that the video will start playing as soon as it is ready
controls New	controls	Specifies that video controls should be displayed (such as a play/pause button etc).
height New	*pixels*	Sets the height of the video player
loop New	loop	Specifies that the video will start over again, every time it is finished
muted New	muted	Specifies that the audio output of the video should be muted
poster New	*URL*	Specifies an image to be shown while the video is downloading, or until the user hits the play button
preload New	auto metadata none	Specifies if and how the author thinks the video should be loaded when the page loads
src New	*URL*	Specifies the URL of the video file
width New	*pixels*	Sets the width of the video player

Link Relations in HTML5

- Defines relationships between documents
- Used in `<link>` and `<a>` tags
- Examples:
- `rel='stylesheet'` for CSS files
- `rel='nofollow'` for search engine instructions
- `rel='icon'` for favicons

- Example:

-
  ```html
- <link rel='stylesheet' href='styles.css'>
- ```
  ```

Objects in HTML5

- Used to embed multimedia (PDFs, SVGs, etc.)
- Can be an alternative to `<iframe>`

- Example (Embedding a PDF):

-
  ```html
- <object data='sample.pdf' type='application/pdf' width='600' height='400'>
-   <p>PDF not supported. <a href='sample.pdf'>Download</a>.</p>
- </object>
- ```
  ```