

# Introduction to

# HTML



# Definitions

- W W W – World Wide Web.
- HTML – **HyperText 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.

# Introduction To HTML

- HTML means Hypertext Markup Language. In 1960 Ted Nelson introduced Hypertext. HTML is a scripting language which is used to create web pages.
- HTML is a hypertext Language because it supports font styled text, pictures, graphics and animations and also it provides hyper links that used to browse the Internet easily.

# Rules to write HTML Code:-

- ❖ ☐ Every HTML document begins with start tag is terminates with an ending tag is</html>
- ❖ ☐ HTML documents should be saved with the extension .html or .htm.
- ❖ ☐ A tag is made up of left operator(<),a right operator(>) and a tag name between these two operators
- ❖ ☐ If you forget to mention the right operator(>) or if you give any space between left operator and tag name browser will not consider it as tag.
- ❖ ☐ At the same time if browser not understands the tag name it just ignores it, browser won't generate any errors.
- ❖ ☐ HTML language is not case sensitive, hence user can write the code in either upper case or lower case. No difference between <HTML></html>

# Tags

- Codes enclosed in brackets
- Usually paired  
`<TITLE>My Web Page</TITLE>`
- **Not** case sensitive  
`<TITLE> = <title> = <TITLE>`

# Choosing Text Editor

- There are many different programs that you can use to create web documents.
- HTML Editors enable users to create documents quickly and easily by pushing a few buttons. Instead of entering all of the HTML codes by hand.
- These programs will generate the HTML Source Code for you.

# Choosing Text Editor

- HTML Editors are excellent tools for experienced web developers; however; it is important that you learn and understand the HTML language so that you can edit code and fix “bugs” in your pages.
- For this Course, we will focus on using the standard Microsoft Windows text editors, Notepad. We may use also textpad.

# Starting Notepad

Notepad is the standard text editor that comes with the Microsoft Windows operating system. To start Notepad in Windows 9x or XP follow the steps below:

- Click on the “Start” button located on your Windows task bar.
- Click on “Programs” and then click on the directory menu labeled “Accessories”.
- Locate the shortcut “Notepad” and click the shortcut once.



# HTML Page Creation & Editing

In this chapter you will learn to create HTML pages with a standard text editor.

## Objectives

Upon completing this section, you should be able to

1. Choose a Text Editor.
2. Create a Basic Starting Document.
3. Understand and set Document Properties.
4. View Your Results in a Browser.

- **Basic Structure of HTML**
- An HTML document's basic structure consists of 5 elements:
- `<!DOCTYPE>`
- `<html>`
- `<head>`
- `<title>`
- `<body>`

- **<!DOCTYPE>**
- The tag in HTML is used to inform the browser about the HTML version used in the web page. It is referred as the **document type declaration (DTD)**.
- Syntax
- **<!DOCTYPE html>**

- **<html>**
- The <html> tag in HTML is used to specify the root of HTML and XHTML pages. The <html>**<html>**
- The <html> tag informs the browser that this is an HTML document. It is the second outer container for everything in an HTML document, followed by the tag. The <html> tag requires a beginning and ending tag.

### Syntax of the <html> Tag

```
<!DOCTYPE html>  
<html>  
...  
</html>
```

- **<head>**
- The <head> tag in HTML is used to contain metadata (**data about data**). It is used between the<html> and <body> tags.
- The head of an HTML document is a section of the document whose content is not displayed in the browser when the page loads. It only contains HTML document metadata, which specifies information about the HTML document.

- the head section of an HTML document plays an essential role in the creation of a website.
- The document title, character set, styles, links, scripts, and other meta information are defined by metadata.
- The following is a list of metadata tags:
  - <title>
  - <style>
  - <meta>
  - <link>
  - <script>
  - <base>

## Syntax of the <head> Tag

```
<!DOCTYPE html>
<html>
  <head>
    ...
  </head>

</html>
```

- **Example:** In this example, we are going to use the `<head>` tag containing the `<style>` (to add CSS to our content) and `<title>` (to add title to our webpage) tag.



```
<!DOCTYPE html>
<html>
<head>
<title>head tag</title>
<style>
  h1{
    color: blue;
  }
</style>
</head>
<body>
  <h1> head tag  </h1>
</body>
</html>
```

Output:

head tag

- **<title>**
- This <title> tag in HTML displays the title of a web page and can help in higher rankings in search results if appropriate keywords are included.
- The most significant meta element to add to our webpage is the <title> element. It gives a relevant title to the full HTML content. It appears at the top of the browser window and gives the webpage a fitting name when saved as a favorite or bookmark.
- It can be found in all HTML/XHTML documents. The <title> element must be positioned between the <head> element, and there can only be one title element per document.

```
<!DOCTYPE html>
<html>
<head>
  <title> ... </title>
</head>

</html>
```

- **<body>**
- The <body> tag in HTML specifies the main content of an HTML document that appears on the browser. It can contain headings, text, paragraphs, photos, tables, links, videos, etc.
- The <body> tag must come after the <head> tag, or it must be inserted between the </head> and </html> tags. This tag is essential for all HTML documents and should only be used once throughout the document.
-

## Syntax of the <body> Tag

```
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Body Tag</title>  
  </head>  
  <body>  
    <h1>...</h1>  
    <p>...</p>  
  </body>  
</html>
```

- **Example:** In the given example, we are going to use the `<body>` tag to add a heading, paragraph, and image to our webpage.

```
<!DOCTYPE html>
<html>
<head>
<title>Body Tag</title>
</head> <body>
<h1>Example of body tag</h1>
<p>This paragraph and the image displayed below is written
between the body tag.</p>

</body>
</html>
```

# Creating a Basic Starting Document

```
<HTML>
```

```
<HEAD>
```

```
  <TITLE>Al al-Bayt University</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

This is what is displayed.

```
</BODY>
```

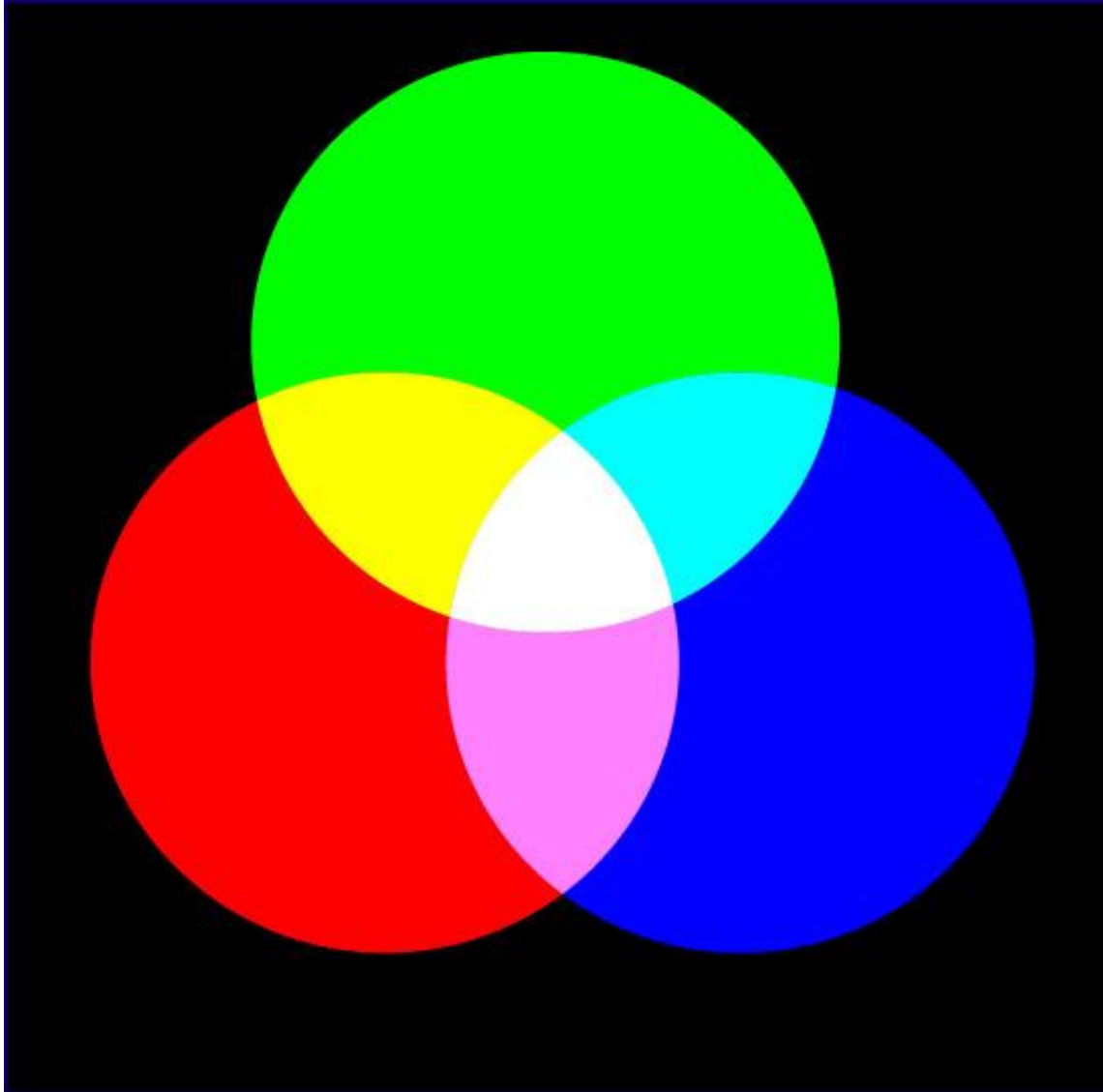
```
</HTML>
```



# Color Codes

- Colors are set using “**RGB**” color codes, which are, represented as hexadecimal values. Each 2-digit section of the code represents the amount, in sequence, of **red**, **green** or **blue** that forms the color.

# Main Colours



# 16 Basic Colors

Color Name	RGB Triplet	Hexadecimal	Color Name	RGB Triplet	Hexadecimal
<b>Aqua</b>	(0,255,255)	00FFFF	<b>Navy</b>	(0,0,128)	000080
<b>Black</b>	(0,0,0)	000000	<b>Olive</b>	(128,128,0)	808000
<b>Blue</b>	(0,0,255)	0000FF	<b>Purple</b>	(128,0,128)	800080
<b>Fuchsia</b>	(255,0,255)	FF00FF	<b>Red</b>	(255,0,0)	FF0000
<b>Gray</b>	(128,128,128)	808080	<b>Silver</b>	(192,192,192)	C0C0C0
<b>Green</b>	(0,128,0)	008000	<b>Teal</b>	(0,128,128)	008080
<b>Lime</b>	(0,255,0)	00FF00	<b>White</b>	(255,255,255)	FFFFFF
<b>Maroon</b>	(128,0,0)	800000	<b>Yellow</b>	(255,255,0)	FFFF00

# Color Codes

1. WHITE
2. BLACK
3. RED
4. GREEN
5. BLUE
6. MAGENTA
7. CYAN
8. YELLOW
9. AQUAMARINE
10. BAKER'S CHOCOLATE
11. VIOLET
12. BRASS
13. COPPER
14. PINK
15. ORANGE

1. #FFFFFF
2. #000000
3. #FF0000
4. #00FF00
5. #0000FF
6. #FF00FF
7. #00FFFF
8. #FFFF00
9. #70DB93
10. #5C3317
11. #9F5F9F
12. #B5A642
13. #B87333
14. #FF6EC7
15. #FF7F00

# The Body Element

- The **BODY** element of a web page is an important element in regards to the **page's appearance**. Here are the attributes of the **BODY** tag to control all the levels:

**TEXT="#RRGGBB"** to change the color of **all the text** on the page (**full page text color.**)

- This element contains information about the page's background color, the background image, as well as the text and link colors.

# Background Color

- It is very common to see web pages with their background color set to white or some other colors.
- To set your document's background color, you need to edit the <BODY> element by adding the BGCOLOR attribute. The following example will display a document with a white background color:

```
<BODY BGCOLOR="#FFFFFF"></BODY>
```

# TEXT Color

- The TEXT attribute is used to control the color of all the normal text in the document. The default color for text is black. The TEXT attribute would be added as follows:

```
<BODY BGCOLOR="#FFFFFF"  
TEXT="#FF0000"></BODY>
```

In this example the document's page color is white and the text would be red.

# Background Image

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

<h2>Background Image</h2>

<p>A background image for a p element:</p>

<p style="background-image: url('img_girl.jpg');">
You can specify background images<br>
for any visible HTML element.<br>
In this example, the background image<br>
is specified for a p element.<br>
By default, the background-image<br>
will repeat itself in the direction(s)<br>
where it is smaller than the element<br>
where it is specified. (Try resizing the<br>
browser window to see how the<br>
background image behaves.
</p>

</body>
</html>
```



# Background sound

- The HTML `<bgsound>` tag is used to play music in the background. This tag is for Internet Explorer only.
- Example
- You can try to run the following code to add background music in HTML –

```
<!DOCTYPE html>
<html>
  <head>
    <title>HTML bgsound Tag</title>
  </head>
  <body>
    <bgsound src = "/html/yourfile.mdi"/>
    <p>Plays sound file in the background.</p>
  </body>
</html>
```

# What is an HTML Tag?

HTML Tags are pre-defined elements in HTML, enclosed within these brackets `< >` signs. For example:

`<html>`, `<table>`, etc. All HTML tags has a particular function associated with them.

Each tag has a special function and a combination of various tags developes a website. **For example**, a `<p>` tag defines a paragraph in the website and a `<table>` tag displays a table.

# Types of tags in HTML-

- There are two types of tags in HTML that are used by the Website Designers:
  1. Paired Tags (Opening and Closing Tags)
  2. Unpaired Tags (Singular Tag)

## Paired Tags - Opening and Closing Tags

Paired tags are a set of two tags with the same name. In each Paired tag set, one is an opening tag, and the other one is the closing tag. The closing tag has a `/` slash, it means that the tag is closed now.

It is necessary to close a paired tag; otherwise, it can result in the malfunctioning of the website. When the content is written within paired tags, then it ensures that the effect of those tags would be limited to only the content between them.

## List of some paired tags in HTML:

Open Tag	Close Tag
<html>	</html>
<table>	</table>
<form>	</form>
<span>	</span>
<ul>	</ul>
<p>	</p>
<head>	</head>
<div>	</div>

## Unpaired Tags - Singular Tags

Unpaired tags are single tags with no closing tag. These tags are also called **Singular Tags**. These are also called **non-container tags** because they do not contain any content.

It is recommended to close the unpaired/singular tags also. But unfortunately, we do not have the closing tag for those. So, an unpaired tag is closed after adding a slash(/) just before the greater than `>` sign. For example: `<br />`.

Some Unpaired Tags are:

Open Tag
<hr>



- # HTML Elements:

## Basic HTML tags

### **1. Body tag:-**

Body tag contain some attributes such as bgcolor, background etc. bgcolor is used for background color, which takes background color name or hexadecimal number and #FFFFFF and background attribute will take the path of the image which you can place as the background image in the browser.

```
<body bgcolor="#F2F3F4" background= "c:\amer\imag1.gif">
```

### **2. Paragraph tag:-**

Most text is part of a paragraph of information. Each paragraph is aligned to the left, right or center of the page by using an attribute called as align.

```
<p align="left" | "right" | "center">
```

HTML is having six levels of heading that are commonly used. The largest heading tag is

# 

This tag places a horizontal line across the system. These lines are used to break the page. This tag also contains attribute i.e., width which draws the horizontal line with the screen size of the browser. This tag does not require an end tag.

42

### 5. base font:-

This specifies format for the basic text but not the headings.

```
<basefont size="10">
```

### 6. font tag:-

This sets font size, color and relative values for a particular text.

```
<font size="10" color="#f1f2f3">
```

### 7. bold tag:-

This tag is used for implement bold effect on the text

```
<b> ..... </b>
```

### 8. Italic tag:-

This implements italic effects on the text.

```
<i> ... .. </i>
```

### 9. strong tag:-

This tag is used to always emphasized the text

```
<strong> ..... </strong>
```

### **10. tt tag:-**

This tag is used to give typewriting effect on the text

<tt> .....</tt>

### **11. sub and sup tag:-**

These tags are used for subscript and superscript effects on the text.

<sub> .....</sub>

<sup> .....</sup>

### **12. Break tag:-**

This tag is used to break the line and start from the next line.

<br>

### **13. &amp; &lt; &gt; &nbsp; &quot;:-**

These are character escape sequence which are required if you want to display characters that HTML uses as control sequences.

Example: < can be represented as &lt;.

#### **14. Anchor tag:-**

This tag is used to link two HTML pages, this is represented by <a>

<a href=" path of the file"> some text </a>

href is an attribute which is used for giving the path of a file which you want to link.



# LINK, VLINK, and ALINK

These attributes control the colors of the different link states:

1. LINK – initial appearance – default = Blue.
2. VLINK – visited link – default = Purple.
3. ALINK – active link being clicked – default = Yellow.

The Format for setting these attributes is:

```
<BODY BGCOLOR="#FFFFFF" TEXT="#FF0000"  
    LINK="#0000FF"  
    VLINK="#FF00FF"  
    ALINK="FFFF00"> </BODY>
```

# Using Image Background

- The BODY element also gives you ability of setting an image as the document's background.
- An example of a background image's HTML code is as follows:

```
<BODY BACKGROUND="hi.gif"  
  BGCOLOR="#FFFFFF"></BODY>
```

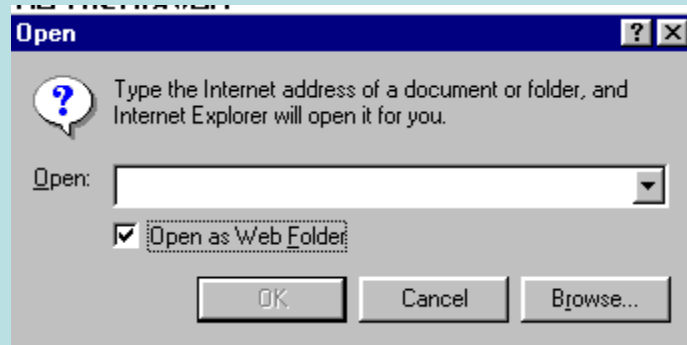


# Previewing Your Work

- Once you have created your basic starting document and set your document properties it is a good idea to save your file.
- To save a file, in NotePad, follow these steps:
  1. Locate and click on the menu called “File”.
  2. Select the option under File Menu labeled “Save As”.
  3. In the “File Name” text box, type in the entire name of your file (including the extension name .html).

# Edit, Save and View Cycle

- To preview Your Work, open a web browser and do the following:
  1. Click on the menu labeled “File”.
  2. Locate the menu option, “Open”.



# Edit, Save and View Cycle

3. In the “Open” dialog box, click on the “Browse” button and locate your web document.
  4. Click “OK” once you have selected your file.
- The web browser will load the same document but with the new revisions. This process is the Edit, Save and View Cycle.

- Character formatting:
- 2.6.1. Heading Tag (H1 to H6) and attribute (ALIGN)
- 2.6.2. Paragraph Tag and attribute (ALIGN)
- 2.6.3. Line Break (BR)
- 2.6.4. Horizontal Rule (HR) and attribute (ALIGN, SIZE, WIDTH, NOSHADE)
- 2.6.5. Comment in HTML ()
- 2.6.6. Text Formatting (B, I, U, BLOCKQUOTE, Q, PREFORMATTED, SUB, SUP, EM, STRIKE, SMALL, BIG, CENTER)

# Headings, Paragraphs, Breaks & Horizontal Rules

In this chapter you will add headings to your page, insert paragraphs, add some breaks, and add horizontal rules.

## Objectives

Upon completing this section, you should be able to

1. List and describe the different Heading elements.
2. Use Paragraphs to add text to a document.
3. Insert breaks where necessary.
4. Add a Horizontal Rule.

# 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:
  1. **H1**: should be used as the highest level of heading, **H2** as the next highest, and so forth.
  2. 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>.

# Horizontal Rule, <HR>

Attribute	Description	Default Value
SIZE	Height of the rule in pixels	2 pixels
WIDTH	Width of the rule in pixels or percentage of screen width	100%
NOSHADE	Draw the rule with a flat look instead of a 3D look	Not set (3D look)
ALIGN	Aligns the line (Left, Center, Right)	Center
COLOR	Sets a color for the rule (IE 3.0 or later)	Not set

# Horizontal Rule, <HR>

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

## Heading 1

Paragraph 1,....

Line 2

---

Line 3

# Character /Text Formatting

In this chapter you will learn how to enhance your page with Bold, Italics, and other character formatting options.

## **Objectives**

Upon completing this section, you should be able to

1. Change the color and size of your text.
2. Use Common Character Formatting Elements.
3. Align your text.
4. Add special characters.
5. Use other character formatting elements.

# Bold, Italic and other Character Formatting Elements

- **<FONT SIZE="+2"> Two sizes bigger</FONT>**
- The size attribute can be set as an absolute value from 1 to 7 or as a relative value using the "+" or "-" sign. Normal text size is 3 (from -2 to +4).
- **<B> Bold </B>**
- **<I> Italic </I>**
- **<U> Underline </U>**
- Color = "#RRGGBB" The COLOR attribute of the FONT element. E.g., **<FONT COLOR="#RRGGBB">this text has color</FONT>**
- **<PRE> Preformatted </PRE>** Text enclosed by PRE tags is displayed in a mono-spaced font. Spaces and line breaks are supported without additional elements or special characters.



# Bold, Italic and other Character Formatting Elements

- **<EM> Emphasis </EM>** Browsers usually display this as italics.
- **<STRONG> STRONG </STRONG>** Browsers display this as bold.
- **<TT> TELETYPE </TT>** Text is displayed in a mono-spaced font. A typewriter text, e.g. fixed-width font.
- **<CITE> Citation </CITE>** represents a document citation (**italics**). **For titles of books, films, etc. Typically displayed in italics. (A Beginner's Guide to HTML)**

# Bold, Italic and other Character Formatting Elements

<P> <FONT SIZE="+1"> One  
Size Larger </FONT> - Normal

–

<FONT SIZE="-1"> One Size  
Smaller </FONT> <BR>

<B> Bold</B> - <I> italics</I> -  
<U> Underlined </U> -

<FONT COLOR="#FF0000">  
Colored </FONT> <BR>

<EM> Emphasized</EM> -  
<STRONG> Strong  
</STRONG> - <TT> Tele Type  
</TT> <BR>

One Size Larger - Normal – One  
Size Smaller

**Bold** - *italics* - Underlined -  
Colored

*Emphasized* - **Strong** - Tele  
Type

# Alignment

- Some elements have attributes for alignment (ALIGN) e.g. **Headings, Paragraphs and Horizontal Rules.**
- The Three alignment values are : LEFT, RIGHT, CENTER.
- **<CENTER></CENTER>** Will center elements.

# Alignment

- **<DIV ALIGN="value"></DIV>** Represents a division in the document and can contain most other element type. The alignment attribute of the DIV element is well supported.
- **<TABLE></TABLE>** Inside a TABLE, alignment can be set for each individual cell.

# Special Characters & Symbols

- These Characters are recognized in HTML as they begin with an ampersand and end with with a semi-colon e.g. **&value;** The value will either be an entity name or a standard ASCII character number. They are called **escape sequences**.
- The next table represents some of the more commonly used special characters. For a comprehensive listing, visit the W3C's section on special characters at:  
[http://www.w3.org/MarkUp/HTMLPlus/htmlplus\\_13.html](http://www.w3.org/MarkUp/HTMLPlus/htmlplus_13.html)

# Special Characters & Symbols

Special Character	Entity Name	Special Character	Entity Name
<b>Ampersand</b>	<b>&amp;amp;</b> &	<b>Greater-than sign</b>	<b>&amp;gt;</b> >
<b>Asterisk</b>	<b>&amp;lowast;</b> **	<b>Less-than sign</b>	<b>&amp;lt;</b> <
<b>Cent sign</b>	<b>&amp;cent;</b> ¢	<b>Non-breaking space</b>	<b>&amp;nbsp;</b> ;
<b>Copyright</b>	<b>&amp;copy;</b> ©	<b>Quotation mark</b>	<b>&amp;quot;</b> "
<b>Fraction one qtr</b>	<b>&amp;frac14;</b> $\frac{1}{4}$	<b>Registration mark</b>	<b>&amp;reg;</b> ®
<b>Fraction one half</b>	<b>&amp;frac12;</b> $\frac{1}{2}$	<b>Trademark sign</b>	<b>&amp;trade;</b> TM

# Special Characters & Symbols

- Additional escape sequences support accented characters, such as:
- **&ouml;**
  - a lowercase o with an umlaut: ö
- **&ntilde;**
  - a lowercase n with a tilde: ñ
- **&Egrave;**
  - an uppercase E with a grave accent: È

**NOTE:** Unlike the rest of HTML, the escape sequences are **case sensitive**. You cannot, for instance, use &LT; instead of &lt;.

# Additional Character Formatting Elements

- **<STRIKE>** strike-through text**</STRIKE>**  
**DEL** is used for **STRIKE** at the latest browsers
- **<BIG>** places text in a big font**</BIG>**
- **<SMALL>** places text in a small font**</SMALL>**
- **<SUB>** places text in subscript position **</SUB>**
- **<SUP>** places text in superscript style position **</SUP>**



# Example

<P><STRIKE> strike-through text </STRIKE></BR>

<BIG>places text in a big font </BIG><BR>

<SMALL> places text in a small font</SMALL><BR>

<SUB> places text in subscript position </SUB>

Normal

<SUP> places text in superscript style position

</SUP><BR> </P>

# Blockquote

- The `<blockquote>` tag in HTML is used to display the long quotations (a section that is quoted from another source). It changes the alignment to make it unique from others. It contains both opening and closing tags. In blockquote tag, we can use elements like heading, list, paragraph, etc.
- Syntax:
- `<blockquote>-----</blockquote>`

```
<html>
  <body>
    <h1>GeeksforGeeks</h1>
    <h2><blockquote> Tag</h2>
    <!--blockquote Tag starts here -->
    <blockquote cite=
"https://www.geeksforgeeks.org/html-tutorials/">
```

```
<p>
```

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document with tag which defines the structure of web pages.

```
</p>
```

```
</blockquote>
```

```
<!--blockquote Tag ends here -->
```

```
</body>
```

```
</html>
```

## **<blockquote> Tag**

HTML stands for HyperText Markup Language. It is used to design web pages using a markup language. HTML is the combination of Hypertext and Markup language. Hypertext defines the link between the web pages. A markup language is used to define the text document within tag which defines the structure of web pages.

# <q>: The Inline Quotation element

The `<q>` [HTML](#) element indicates that the enclosed text is a short inline quotation. Most modern browsers implement this by surrounding the text in quotation marks. This element is intended for short quotations that don't require paragraph breaks; for long quotations use the `<blockquote>` element.

# Css: q:{font-style:italic;}

```
1 <p>When Dave asks HAL to open the pod  
  bay door, HAL answers: <q  
  cite="https://www.imdb.com/title/tt00626  
  22/quotes/qt0396921">I'm sorry, Dave.  
  I'm afraid I can't do that.</q></p>
```

When Dave asks HAL to open  
the pod bay door, HAL  
answers: *"I'm sorry, Dave. I'm  
afraid I can't do that."*

# Preformatted text

- HTML pre tag
- The **HTML <pre> tag** is used *to specify pre formatted texts*. Texts within <pre>.....</pre> tag is displayed in a fixed-width font. Usually it is displayed in Courier font. It maintains both space and line break.
- It is widely used to display language examples e.g. Java, C#, C, C++ etc because it displays the code as it is typed.
- HTML pre tag example

## HTML pre tag example

**<pre>**

This is a formatted text  
by using the HTML pre tag. It maintains  
both space and line break.

**</pre>**



```
<!DOCTYPE>
<html>
<body>
<pre>
package com.javatpoint;
public class FirstJava{
public static void main(String args[]){
System.out.println("hello java");
}
}
</pre>
</body>
</html>
```

# Lists

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

## **Objectives**

Upon completing this section, you should be able to

1. Create an unordered list.
2. Create an ordered list.
3. Create a defined list.
4. Nest Lists.

# List Elements

- HTML supplies several list elements. Most list elements are composed of one or more <LI> (List Item) elements.
- 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 in Netscape Navigator 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>

1. **List item ...**

2. **List item ...**

3. **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

i. List item ...

ii. List item ...

Text ....

iii. 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,
  - o Hang on to the canoe
  - o Use the canoe for support and
  - o 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

In this chapter you will learn about images and how to place images in your pages.

## **Objectives**

Upon completing this section, you should be able to

1. Add images to your pages.

# 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.
- **HSPACE:** is for Horizontal Space on both sides of the image specified in pixels. A setting of 5 will put 5 pixels of invisible space on both sides of the image.
- **VSPACE:** is for Vertical Space on top and bottom of the image specified in pixels. A setting of 5 will put 5 pixels of invisible space above and below the image.

# Some Examples on images

- 1) `<IMG SRC="jordan.gif" border=4>`
- 2) `<IMG SRC=" jordan.gif" width="60" height="60">`
- 3) `<IMG SRC="jordan.gif" ALT="This is a text that goes with the image">`
- 4) `<IMG SRC=" jordan.gif " Hspace="30" Vspace="10" border=20>`
- 5) `< IMG SRC =" jordan.gif" align="left">`  
`blast blast blast blast blast`

# Anchors, URLs and Image Maps

In this chapter you will learn about Uniform Resource Locator, and how to add them as Anchor or Links inside your web pages.

## Objectives

Upon completing this section, you should be able to

1. Insert links into documents.
2. Define Link Types.
3. Define URL.
4. List some commonly used URLs.
5. Plan an Image Map.

# 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.

# More on LINKs

```
<body LINK="#C0C0C0" VLINK="#808080"  
ALINK="#FF0000">
```

- **LINK** - standard link - to a page the visitor hasn't been to yet. (standard color is blue - #0000FF).
- VLINK** - visited link - to a page the visitor has been to before. (standard color is purple - #800080).
- ALINK** - active link - the color of the link when the mouse is on it. (standard color is red - #FF0000).

**If the programmer what to change the color**

- Click `<a href="http://www.yahoo.com"><font color="FF00CC">here</font></a>` to go to yahoo.

# Internal Links

- Internal Links : Links can also be created inside large documents to simplify navigation. Today's world wants to be able to get the information quickly. Internal links can help you meet these goals.

1. Select some text at a place in the document that you would like to create a link to, then add an anchor to link to like this:

`<A NAME="bookmark_name"></A>`

The Name attribute of an anchor element specifies a location in the document that we link to shortly. All NAME attributes in a document must be unique.

2. Next select the text that you would like to create as a link to the location created above.

`<A HREF="#bookmark_name">Go To Book Mark</A>`

# E-Mail (Electronic Mail)

E.g. <mailto:kmf@yahoo.com>

- The type of service is identified as the mail client program. This type of link will launch the users mail client.
- The recipient of the message is [kmf@yahoo.com](mailto:kmf@yahoo.com)

<A HREF="mailto:[kmf@yahoo.com](mailto:kmf@yahoo.com)">Send me  
More Information </A>

# Image Maps

- Image maps are images, usually in **gif** format that have been divided into regions; clicking in a region of the image cause the web surfer to be connected to a new URL. Image maps are graphical form of creating links between pages.
- There are two type of image maps:

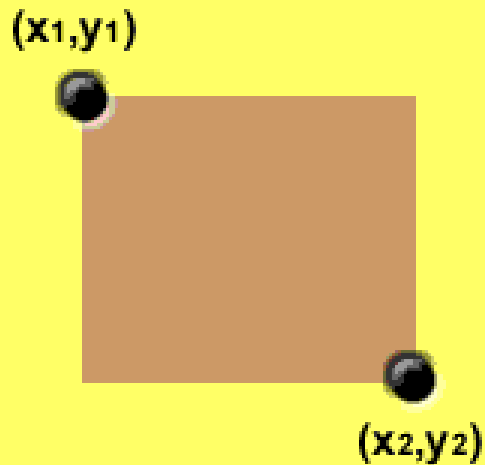
**Client side and server side**

Both types of image maps involve a listing of co-ordinates that define the mapping regions and which URLs those coordinates are associated with. This is known as the map file.

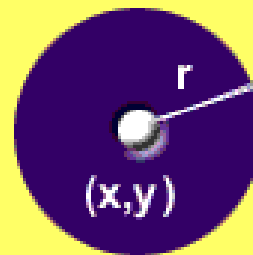


# Area Shapes Used

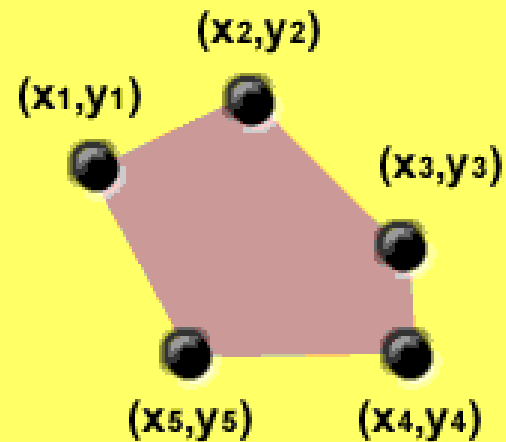
Finding the coordinates....



Rectangle



Circle



Polygon

# Client-Side Image Maps

- Client-side image maps (USEMAP) use a map file that is part of the HTML document (in an element called MAP), and is linked to the image by the Web browser.

```
<IMG SRC="note.GIF" Width=200 Height=200  
border="5" USEMAP="#map1">
```

```
<MAP NAME="map1">
```

```
<AREA SHAPE="RECT" COORDS="0,0,90,90"  
HREF="hi.html" ALT="see me...">
```

```
<AREA SHAPE="RECT" COORDS="100,100,160,160"  
HREF="divPara.html" ALT="see him..." >
```

```
<AREA SHAPE="CIRCLE" COORDS="150,50,20"  
HREF="house.html" ALT="see it..." >
```

```
</MAP>
```

We can use Poly as well as Rect.....

# Shapes, Coords

- Types of Shapes
  - Rect → used for squares and ordered shapes.
  - Circle → used for circles.
  - Poly → used for unordered shapes.
- Number of coordenations for each shape:
  - Rect → 4 numbers for two corners
  - Circle → 3 numbers for the center & R
  - Poly → depends on the number of corners of the shape( 2 numbers for each corner)

# Tables

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

Objectives:

Upon completing this section, you should be able to:

1. Insert a table.
2. Explain a table's attributes.
3. Edit a table.
4. Add a table header.

# Tables

- The `<TABLE></TABLE>` element has four sub-elements:
  1. Table Row `<TR></TR>`.
  2. Table Header `<TH></TH>`.
  3. Table Data `<TD></TD>`.
  4. 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.



# Table Attributes

- **CellPadding**: 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, will be titled in IE3.0 and above.
- **BorderColor, BorderColorDark**.

# Table Caption

- A table caption allows you to specify a line of text that will appear centered above or below 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> <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

Column 1 Header	Column 2 Header
Row 1 Col 1	
Row 2 Col 1	Row 2 Col 2
	Row 3 Col 2

# Special Things to Note

- **TH, TD and TR should always have end tags.**  
Although the end tags are formally optional, many browsers will mess up the formatting of the table if you omit the end tags. In particular, you should ***always*** use end tags if you have a TABLE within a TABLE -- in this situation, the table parser gets hopelessly confused if you don't close your TH, TD and TR elements.
- **A default TABLE has no borders**  
By default, tables are drawn without border lines. You need the BORDER attribute to draw the lines.
- **By default, a table is flush with the left margin**  
TABLEs are plopped over on the left margin. If you want centered tables, You can either: place the table inside a DIV element with attribute ALIGN="center".  
Most current browsers also supports table alignment, using the ALIGN attribute. Allowed values are "left", "right", or "center", for example: <TABLE ALIGN="left">. The values "left" and "right" float the table to the left or right of the page, with text flow allowed around the table. This is entirely equivalent to IMG alignment



# 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



# Frames

- Frames are a relatively new addition to the HTML standard. First introduced in Netscape Navigator 2.0.

Objectives:

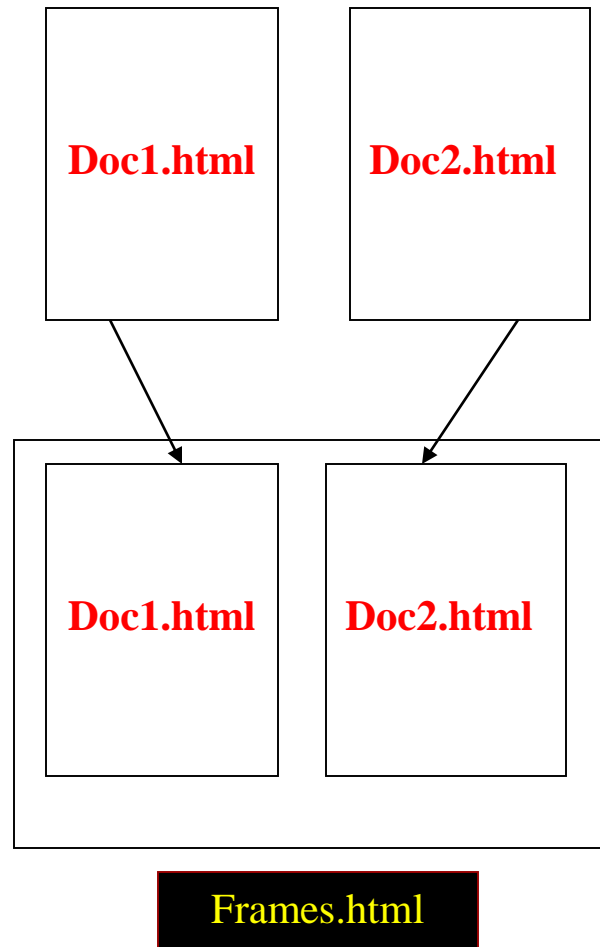
Upon completing this section, you should be able to:

- Create a Frame based page.
- Work with the Frameset, Frame, and Noframes elements.
- Use the attributes of the Frames elements to control the display.
- Set Targets appropriately.

# Frames

- A framed page is actually made up of multiple HTML pages. There is one HTML document that describes how to break up the single browser window into multiple windowpanes. Each windowpane is filled with an HTML document.
- For Example to make a framed page with a windowpane on the left and one on the right requires three HTML pages. ***Doc1.html*** and ***Doc2.html*** are the pages that contain content. ***Frames.html*** is the page that describes the division of the single browser window into two windowpanes.

# Frames



# Frame Page Architecture

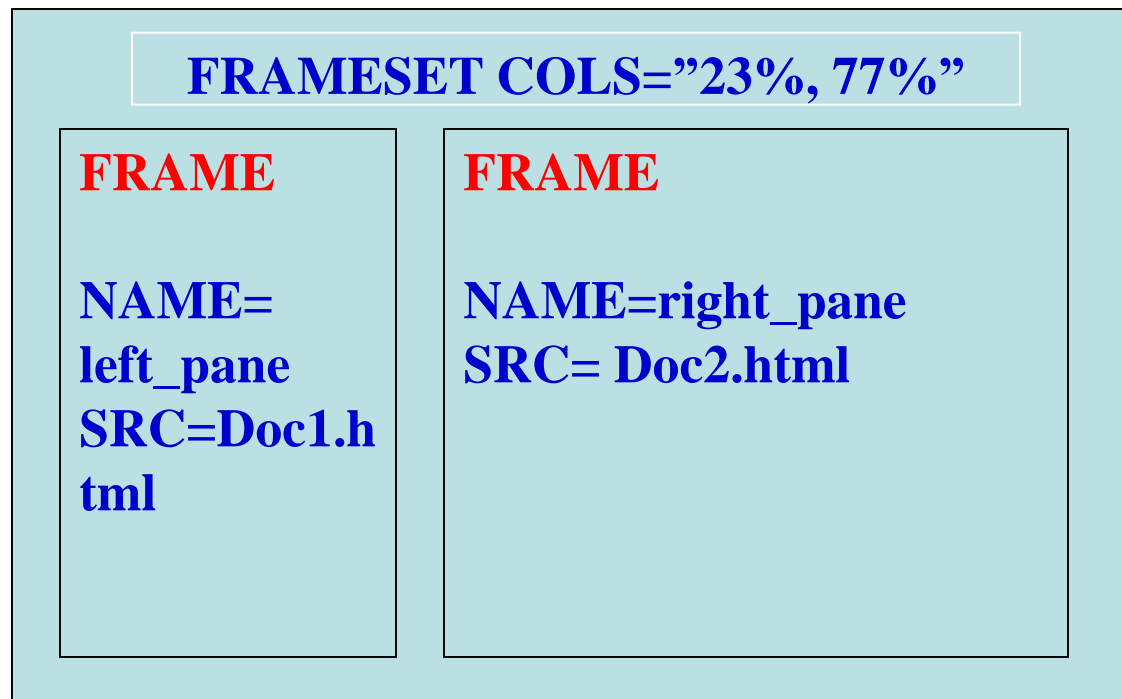
- A **<FRAMESET>** element is placed in the html document before the **<BODY>** element. The **<FRAMESET>** describes the amount of screen real estate given to each windowpane by dividing the screen into **ROWS** or **COLS**.
- The **<FRAMESET>** will then contain **<FRAME>** elements, **one per division** of the browser window.
- Note: Because there is no **BODY** container, FRAMESET pages can't have **background images** and **background colors** associated with them.

# Frame Page Architecture

```
<HTML>
<HEAD>
<TITLE> Framed Page </TITLE>
<FRAMEset COLS="23%,77%">
<FRAME SRC="Doc1.html">
<FRAME SRC="Doc2.html">
</FRAMEset >
</HEAD>

</HTML>
```

**The Diagram below is a graphical view  
of the document described above**





# <FRAMESET> Container

**<FRAMESET>** : The FRAMESET element creates divisions in the browser window in a single direction. This allows you to define divisions as either rows or columns.

- **ROWS** : Determines the size and number of rectangular rows within a <FRAMESET>. They are set from top of the display area to the bottom.

## Possible values are:

- Absolute pixel units, I.e. “360,120”.
- A percentage of screen height, e.g. “75%,25%”.
- Proportional values using the asterisk (\*). This is often combined with a value in pixels , e.g. “360,\*”.
- <Frameset cols=“200,20%,\*,2\*”>

# Creating a Frames Page

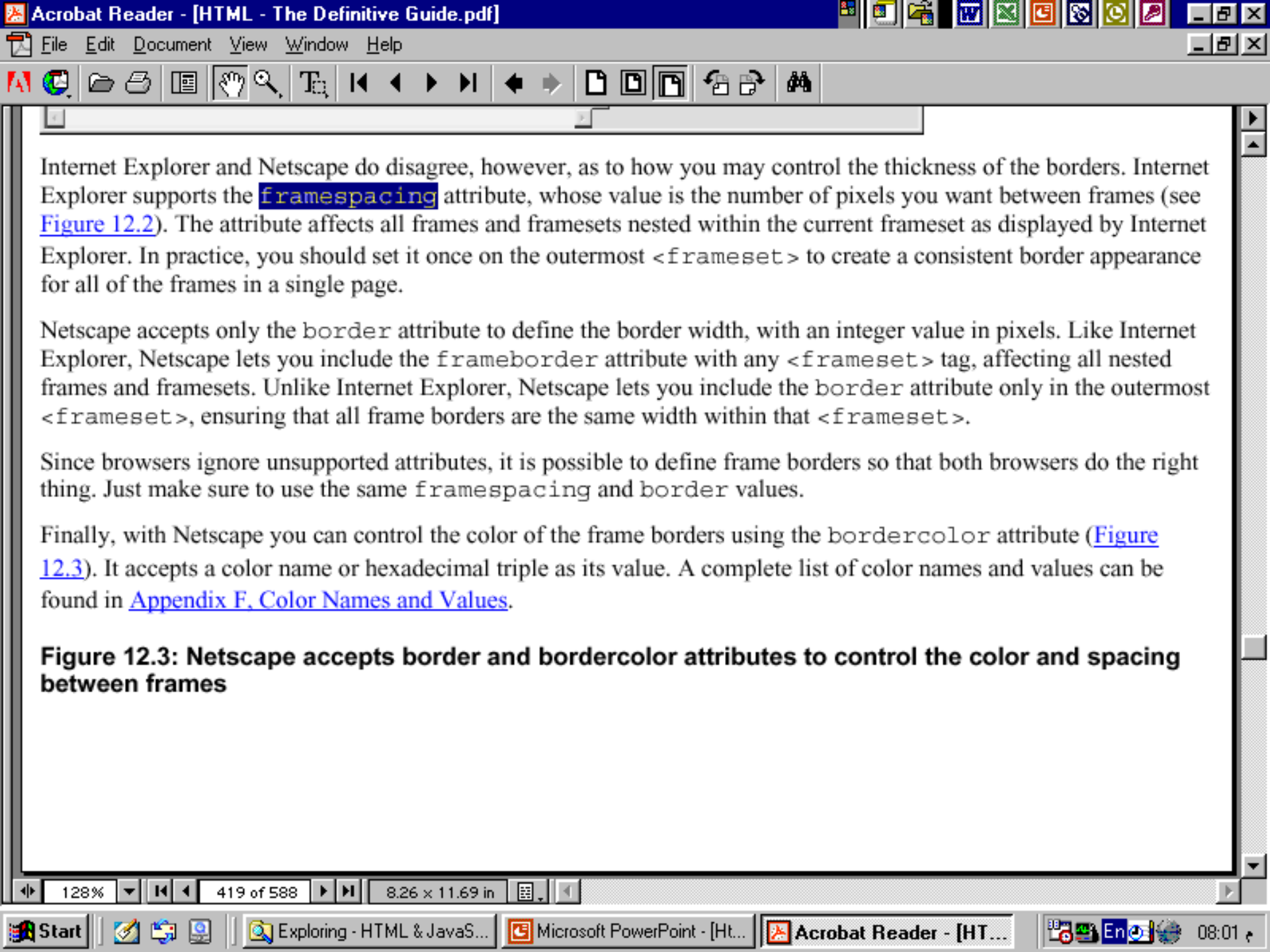
- **COLS**: Determines the size and number of rectangular columns within a <FRAMESET>. They are set from **left** to **right** of the display area.

## Possible values are:

- Absolute pixel units, I.e. “480,160”.
- A percentage of screen width, e.g. “75%,25%”.
- Proportional values using the asterisk (\*). This is often combined with a value in pixels , e.g. “480,\*”.

# Creating a Frames Page

- **FRAMEBORDER** : Possible values **0**, **1**, **YES**, **NO**. A setting of zero will create a borderless frame.
- **FRAMESPACING**: This attribute is specified in **pixels**. If you go to borderless frames you will need to set this value to zero as well, or you will have a gap between your frames where the border used to be.
- **BORDER(thickness of the Frame)**: This attribute specified in pixels. A setting of zero will create a borderless frame. Default value is 5.
- **BORDERCOLOR**: This attribute is allows you choose a color for your border. This attribute is rarely used.



Internet Explorer and Netscape do disagree, however, as to how you may control the thickness of the borders. Internet Explorer supports the `framespacing` attribute, whose value is the number of pixels you want between frames (see [Figure 12.2](#)). The attribute affects all frames and framesets nested within the current frameset as displayed by Internet Explorer. In practice, you should set it once on the outermost `<frameset>` to create a consistent border appearance for all of the frames in a single page.

Netscape accepts only the `border` attribute to define the border width, with an integer value in pixels. Like Internet Explorer, Netscape lets you include the `border` attribute with any `<frameset>` tag, affecting all nested frames and framesets. Unlike Internet Explorer, Netscape lets you include the `border` attribute only in the outermost `<frameset>`, ensuring that all frame borders are the same width within that `<frameset>`.

Since browsers ignore unsupported attributes, it is possible to define frame borders so that both browsers do the right thing. Just make sure to use the same `framespacing` and `border` values.

Finally, with Netscape you can control the color of the frame borders using the `bordercolor` attribute ([Figure 12.3](#)). It accepts a color name or hexadecimal triple as its value. A complete list of color names and values can be found in [Appendix F, Color Names and Values](#).

**Figure 12.3: Netscape accepts `border` and `bordercolor` attributes to control the color and spacing between frames**

# <FRAME>

**<FRAME>**: This element defines a single frame within a frameset. There will be a FRAME element for each division created by the FRAMESET element. This tag has the following attributes:

- **SRC**: Required, as it provides the URL for the page that will be displayed in the frame.
- **NAME**: Required for frames that will allow targeting by other HTML documents. Works in conjunction with the target attribute of the <A>, <AREA>, <BASE>, and <FORM> tags.

# <FRAME>

- **MARGINWIDTH**: Optional attribute stated in pixels. Determines horizontal space between the <FRAME> contents and the frame's borders.
- **MARGINHEIGHT**: Optional attribute stated in pixels. Determines vertical space between the <FRAME> contents and the frame's borders.
- **SCROLLING**: Displays a scroll bar(s) in the frame. Possible values are:
  1. **Yes** – always display scroll bar(s).
  2. **No** – never display scroll bar(s).
  3. **Auto** – browser will decide based on frame contents.

By default: scrolling is auto.

# <FRAME>

- **NORESIZE**: Optional – prevents viewers from resizing the frame. By default the user can stretch or shrink the frame's display by selecting the frame's border and moving it up, down, left, or right.

# <NOFRAMES>

- **<NOFRAMES>**: Frame – capable browsers ignore all HTML within this tag including the contents of the BODY element. This element does not have any attributes.

<HTML>

<HEAD>

<TITLE> Framed Page </TITLE>

</HEAD>



# <NOFRAMES>

<FRAMESET COLS="23%,77%">

<FRAME SRC="" NAME="left\_pane">

<FRAME SRC="" NAME="right\_pane">

<NOFRAMES>

<P> This is a Framed Page. Upgrade your browser to support frames.</P>

</NOFRAMES></FRAMESET>

# Compound FRAMESET Divisions

- In this case a second **FRAMESET** element will be inserted in the place of the **FRAME** element that would describe the second row.
- The second **FRAMESET** element will divide the remaining screen real estate into **2** columns.
- This nested **FRAMESET** will then be followed by **2 FRAME** elements to describe each of the subsequent frame divisions created.

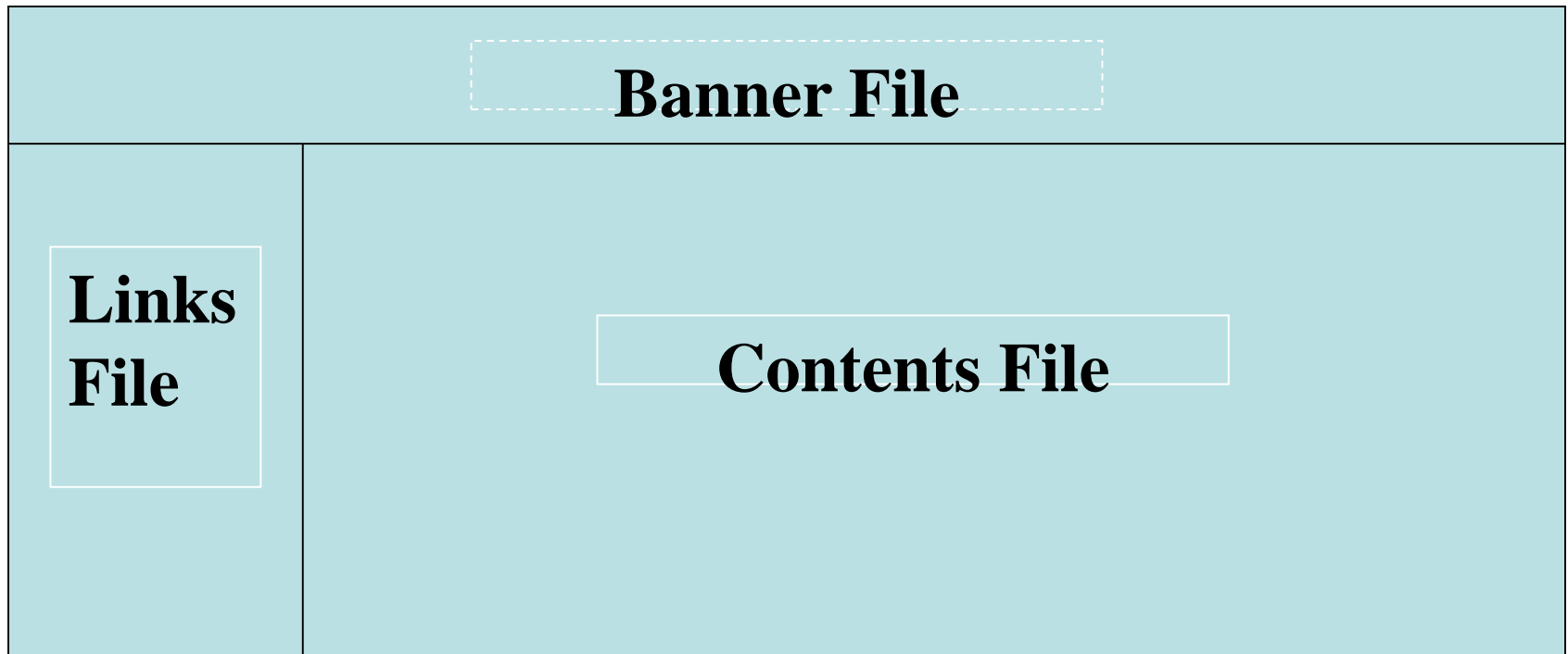
# Compound FRAMESET Divisions

```
<html>
<head>
<title> Compound Frames Page</title>
</head>
<frameset rows="120,*">
<frame src="banner_file.html"
      name="banner">
<frameset cols="120,*">
<frame src="links_file.html"
      name="links">
<frame src="content_file.html"
      name="content">
```

```
<noframes>
<p>
Default
  message
</p>
</noframes>
</frameset>
</frameset>
</head>
```

# Compound FRAMESET Divisions

You may want to create a frames design with a combination of rows and columns.



# Compound FRAMESET Divisions Example

**<HEAD>**

**<FRAMESET ROWS="25%,50%,25%"**

**<FRAME SRC="">**

**<FRAMESET COLS="25%,\*">**

**<FRAME SRC="">**

**<FRAME SRC="">**

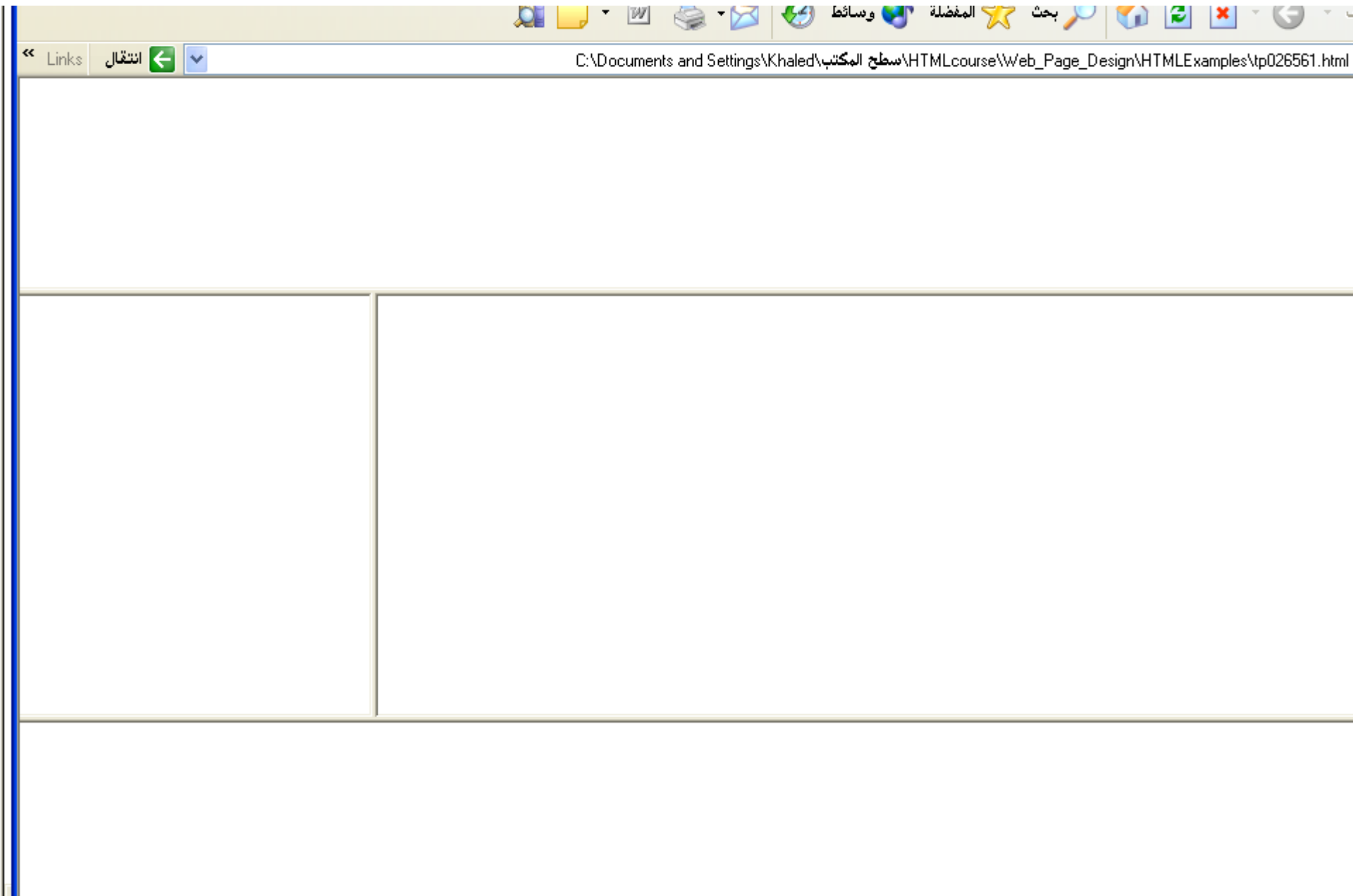
**</FRAMESET>**

**<FRAME SRC="">**

**</FRAMESET>**

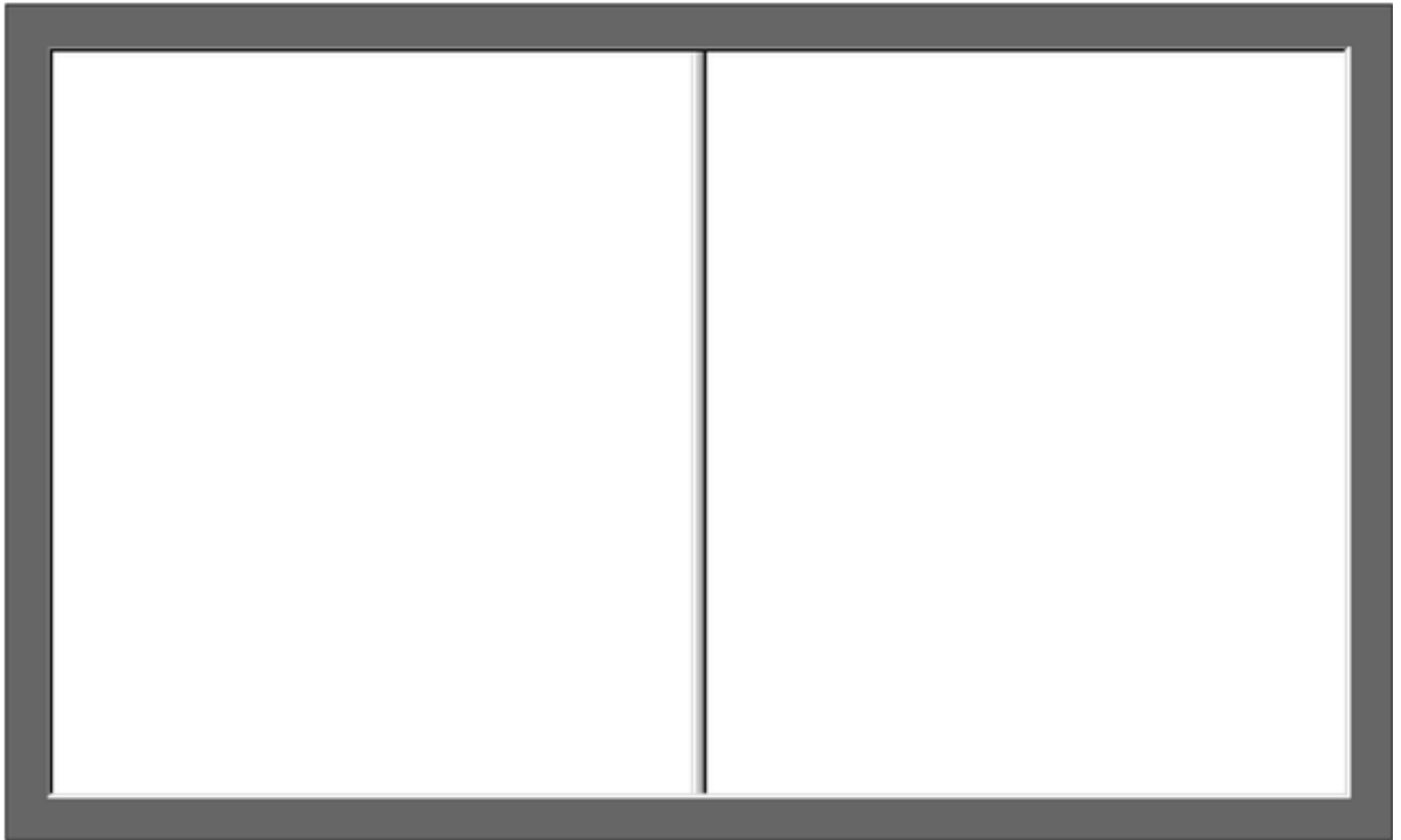
**</HEAD>**

# Output



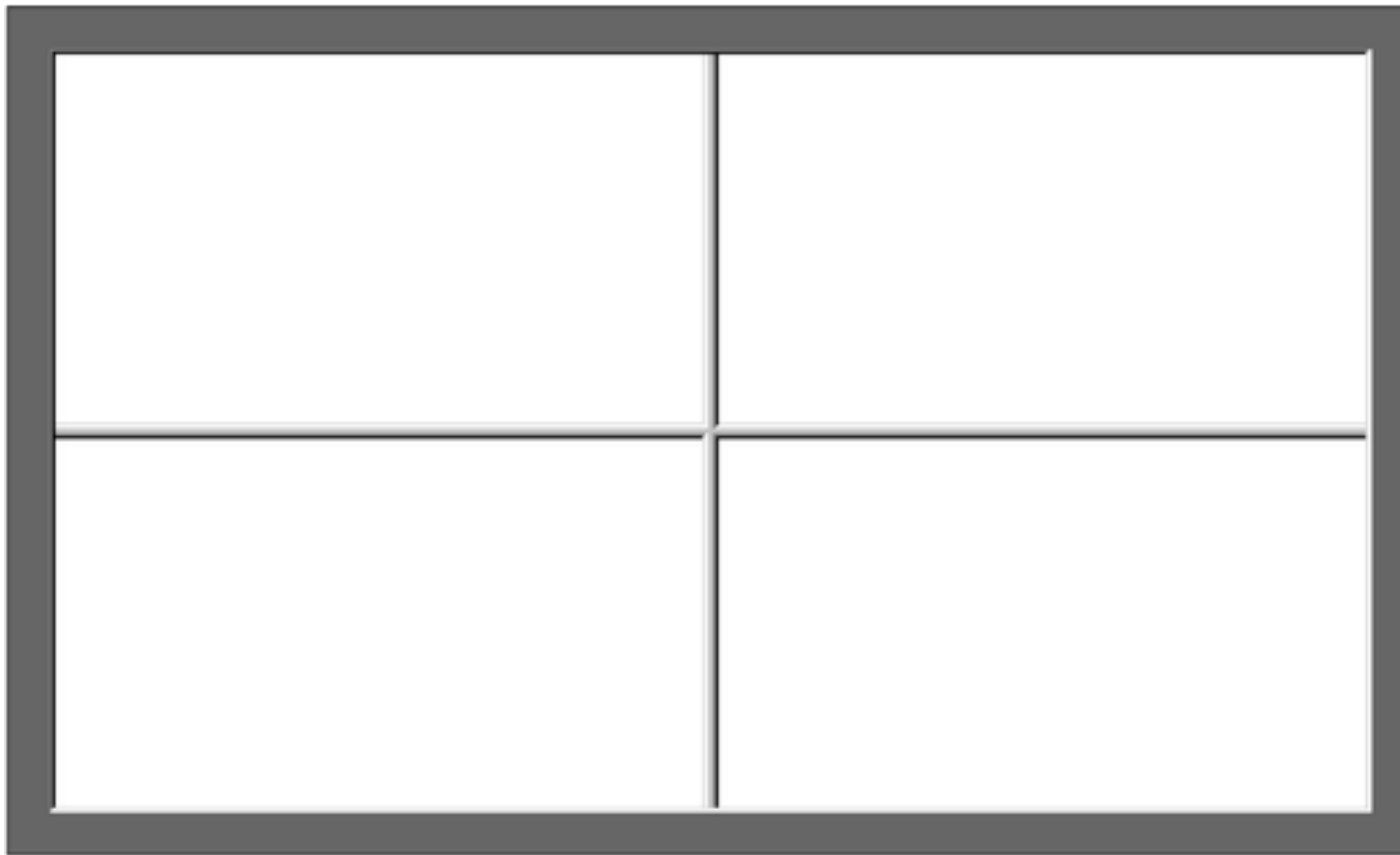


**Figure 5-14:** Frames created with `<FRAMESET ROWS="50%, 50%">`



**Figure 5-15:** Frames created with `<FRAMESET COLS="50%, 50%">`





**Figure 5-13:** Frames created with `<FRAMESET ROWS="50%, 50%" COLS="50%, 50%">`

# Frame Formatting

- Example:

```
<frameset rows="20%, *, 20%">
```

```
    <frame src="header.html" noresize  
    scrolling=no>
```

```
    <frame src="body.html">
```

```
    <frame src="navigationbar.html"  
    noresize scrolling=no>
```

```
</frameset>
```

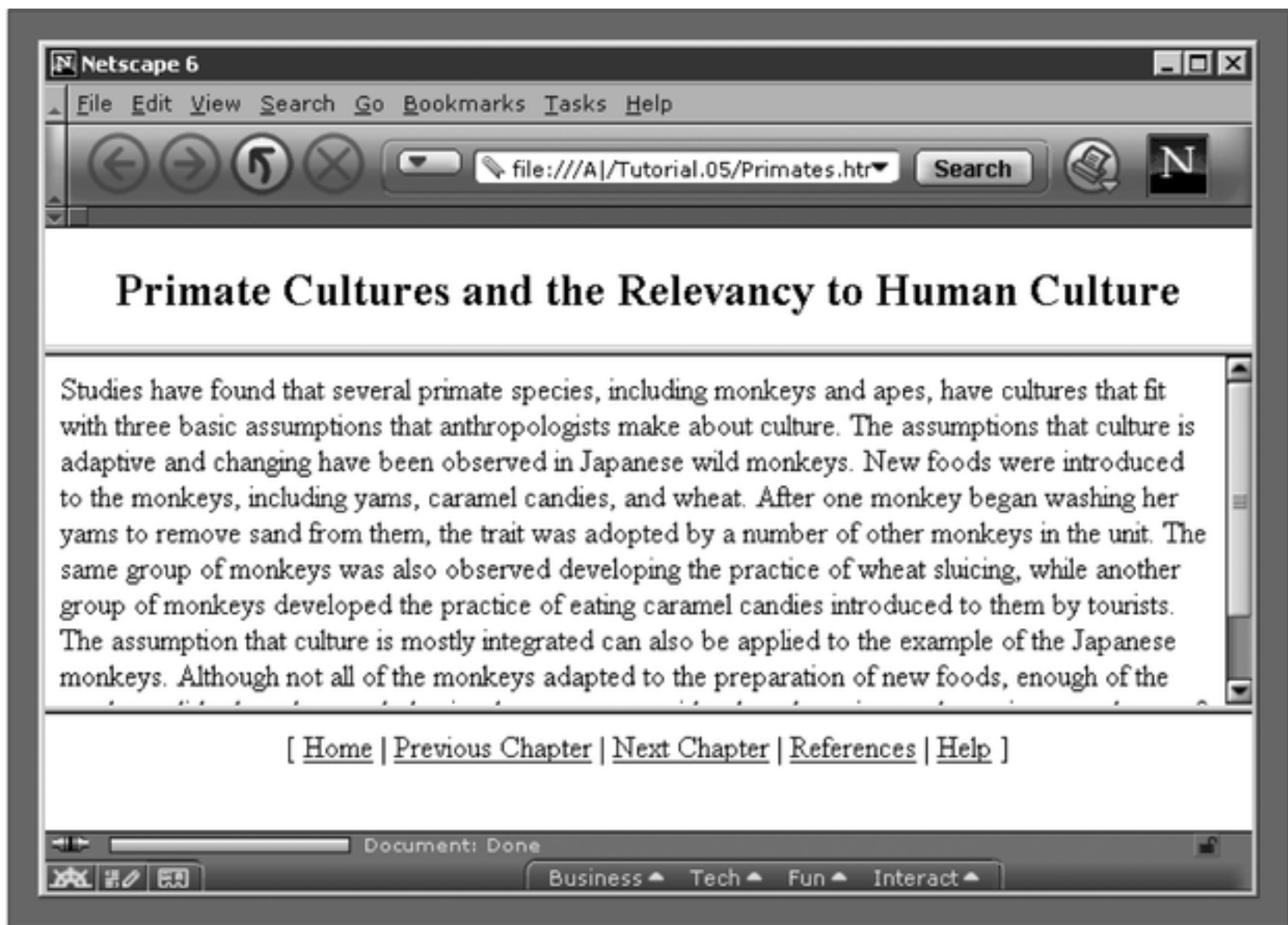


Figure 5-24: Output of program that includes NORESIZE and SCROLLING attributes

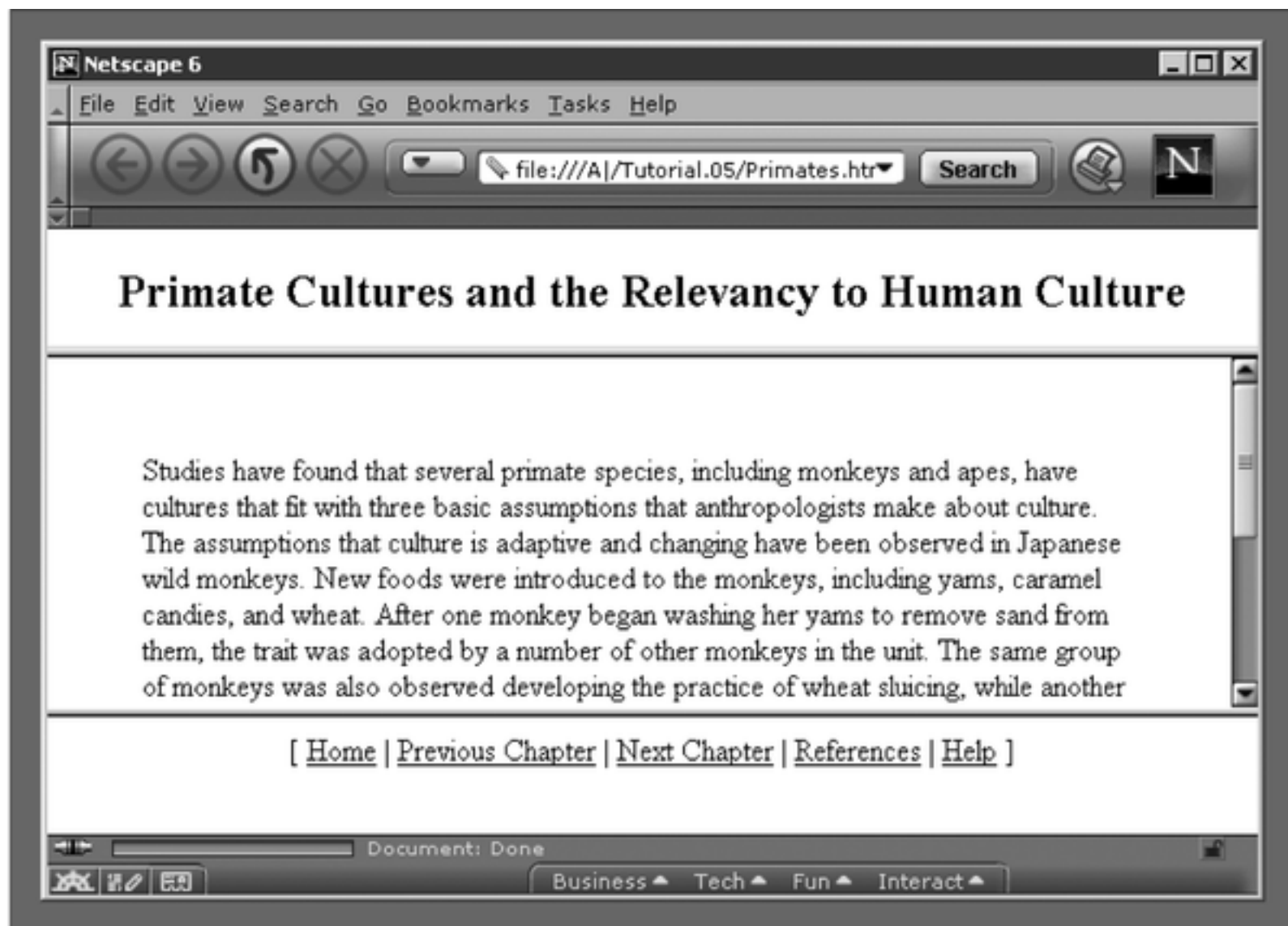


Figure 5-25: Middle frame changed to <FRAME SRC="body.html" MARGINHEIGHT=50 MARGINWIDTH=50>

# What do the following mean?

- 1) <FRAMESET COLS="2\*, 3\*, 5\*">
- 2) <FRAMESET COLS="150, 20%, \*, 3\*">

So what are the space-allocation priorities?

Absolute pixel values are always assigned **space first**, in order from **left** to **right**. These are followed by **percentage** values of the total space. Finally, **proportional** values are divided based upon what space is **left**.

# Generic Frame Formula

- The <FRAME> tag has six associated attributes: SRC, NAME, MARGINWIDTH, MARGINHEIGHT, SCROLLING, and NORESIZE. Here's a complete generic FRAME:
- <FRAME SRC="url" NAME="window\_name"  
SCROLLING=YES|NO|AUTO  
MARGINWIDTH="value"  
MARGINHEIGHT="value"  
NORESIZE>

# What will be the Output?

**<FRAMESET ROWS="\*, 2\*, \*" COLS="2\*, \*">**

**<FRAME SRC="">**

**<FRAME SRC="">**

**<FRAME SRC="">**

**<FRAME SRC="">**

**<FRAME SRC="">**

**<FRAME SRC="">**

**</FRAMESET>**

# Targets

- When you use links for use in a frames environment you will need to specify an additional attribute called **TARGET**.
- The **TARGET** attribute uses the NAME attribute of the **FRAME** element.
- If we were to place a link in doc1.html that linked to doc3.html and we wanted doc3.html to be displayed in the right windowpane; the HTML code would appear in doc1.html as follows:

```
<A HREF="doc3.html"  
TARGET="right_pane">Link to Document 3  
</A>
```



# Special Targets

- There are **4** special target names that cannot be assigned by the NAME attribute of the FRAME tag.
- 1. **TARGET=“\_top”** : This loads the linked document into the full browser window with the URL specified by the HREF attribute. All frames disappear, leaving the new linked page to occupy the entire window. The back is turned on.
- 2. **TARGET=“\_blank”** : Opens an unnamed new browser window and loads the document specified in the URL attribute into the new window (and your old window stays open). The back is turned off. Other windows remains on.
- 3. **TARGET=“\_self”** : Loads the document in the same window where the anchor was {*Clicked*}. This is the **default** setting for linking elements.
- 4. **TARGET=“\_parent”** : the \_parent frame is a prior frameset that the current frameset was “spawned” from. If there isn’t one it is the browser window. The document is loaded into the area occupied by the columns or rows frameset containing the frame that contains the link. The back is turned on. All windows disappear.

If a frame contains the following link, then clicking the link launches a new, unnamed browser display window that contains the content defined in stuff.HTM. This can be a simple HTML document, or an entirely new FRAMESET definition.

**1.      <A HREF="stuff.html" TARGET="\_blank">**

If a frame contains the following link, then clicking the link will simply cause the frame which contains the link to clear, and its content will be replaced with whatever is in stuff.htm.

**2.      <A HREF="stuff.html" TARGET="\_self">**

If a frame contains the following link, the frameset that contains the frame that contains this link will be replaced by stuff.HTM.

**3.      <A HREF="stuff.html" TARGET="\_parent">**

Finally, if a frame contains the following link, clicking the link replaces the entire browser window with the contents of stuff.HTM.

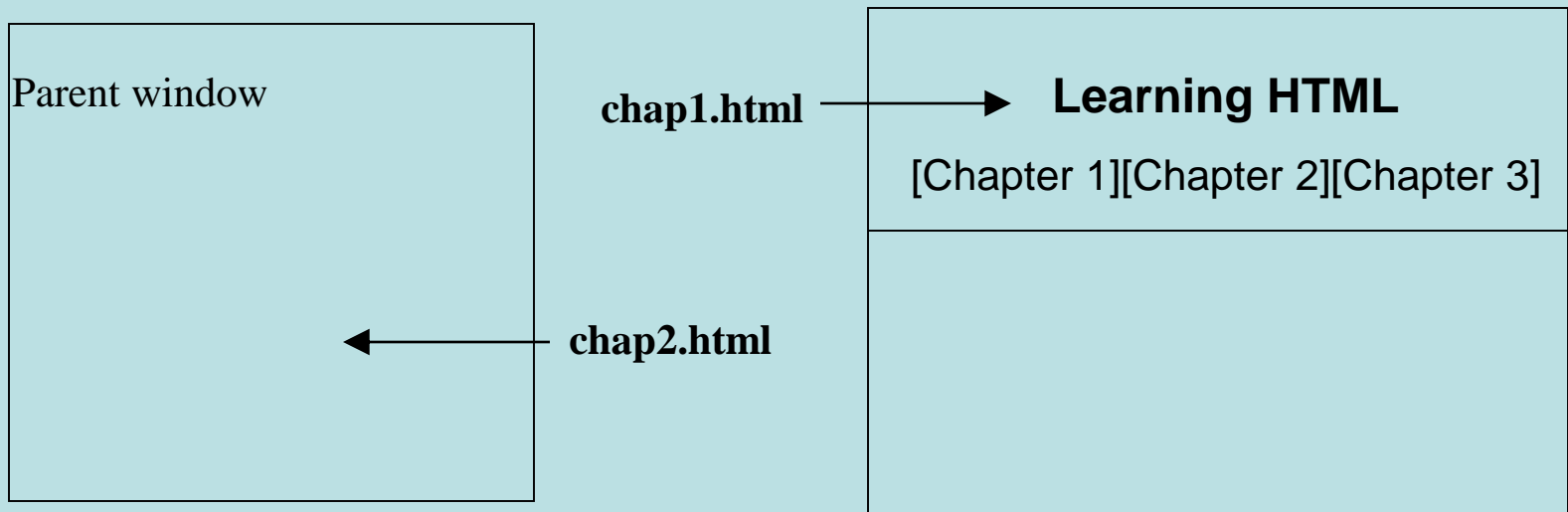
**4.      <A HREF="stuff.html" TARGET="\_top">**

# Targeting links to frames

The TARGET attribute allows you to specify the frame into which a page is to be loaded into in a frames setting.

```
<A HREF="chap1.html" TARGET="_self"> [Chapter 1]</A>
```

```
<A HREF="chap1.html" TARGET="_parent"> [Chapter 2]</A>
```



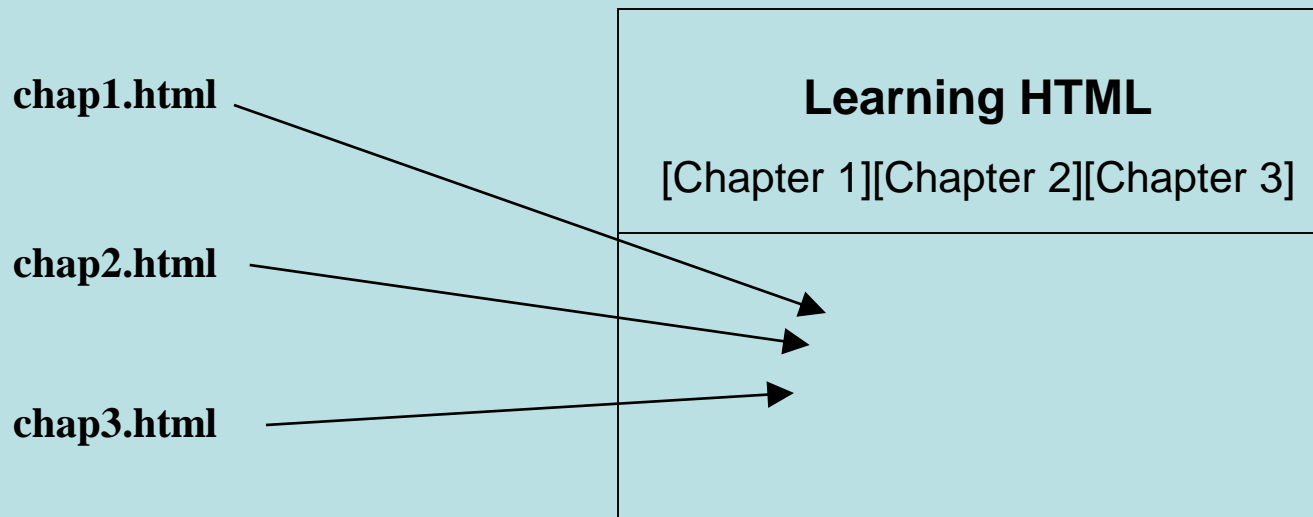
# Targeting links to frames

The TARGET attribute allows you to specify the frame into which a page is to be loaded into in a frames setting.

```
<A HREF="chap1.html" TARGET="bottom"> [Chapter 1]</A>
```

```
<A HREF="chap2.html" TARGET="bottom"> [Chapter 2]</A>
```

```
<A HREF="chap3.html" TARGET="bottom"> [Chapter 3]</A>
```



# 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.

## ***Objectives:***

Upon completing this section, you should be able to

1. Create a FORM.
  2. Add elements to a FORM.
  3. Define CGI (**Common Gateway Interface**).
  4. Describe the purpose of a CGI Application.
  5. Specify an action for the FORM.
- Forms work in all browsers.
  - Forms are Platform Independent.

# 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** of the **CGI** (Common Gateway Interface) program 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. The CGI application should be written to accept the data from either method.
- **NAME**: is a form name used by **VBScript** or **JavaScripts**.
- **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**, **Hidden** 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



Amman

Irbed

Karak

is foreign?



Male:



Female:



Submit

Reset

# Form Elements

## <INPUT> Element's Properties

**TYPE=** Type of INPUT entry field.

**NAME =** Variable name passed to CGI application

**VALUE=** The data associated with the variable name to be passed to the CGI application

**CHECKED=** Button/box checked

**SIZE=** Number of visible characters in text field

**MAXLENGTH=** 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.
- **MAXLENGTH** : determines the maximum number of characters that the field will accept.
- **NAME**: is the name of the variable to be sent to the CGI application.
- **VALUE**: will display its contents as the default value.

# Example on Text Box

```
<TITLE>Form_Text_Type</TITLE>
</HEAD> <BODY>
<h1> <font color=blue>Please enter the following
  bioData</font></h1>
<FORM name="fome1" Method= " get " Action= " URL " >
First Name: <INPUT TYPE="TEXT" NAME="FName"
SIZE="15" MAXLENGTH="25"><BR>
Last Name: <INPUT TYPE="TEXT" NAME="LName"
SIZE="15" MAXLENGTH="25"><BR>
Nationality: <INPUT TYPE="TEXT" NAME="Country"
SIZE="25" MAXLENGTH="25"><BR>
The Phone Number: <INPUT TYPE="TEXT" NAME="Phone"
SIZE="15" MAXLENGTH="12"><BR>
</FORM> </BODY> </HTML>
```

# Output

Form\_Text\_Type - Microsoft Internet Explorer

ملف تحرير عرض المفضلة أدوات تعليمات

« < > >> الخلف

« Links انتقال < >> عنوان C:\jdk\bin\tp01c7aa.html

## Please enter the following bioData

First Name:

Last Name:

Nationality:

The Phone Number:

جهاز الكمبيوتر

# 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.

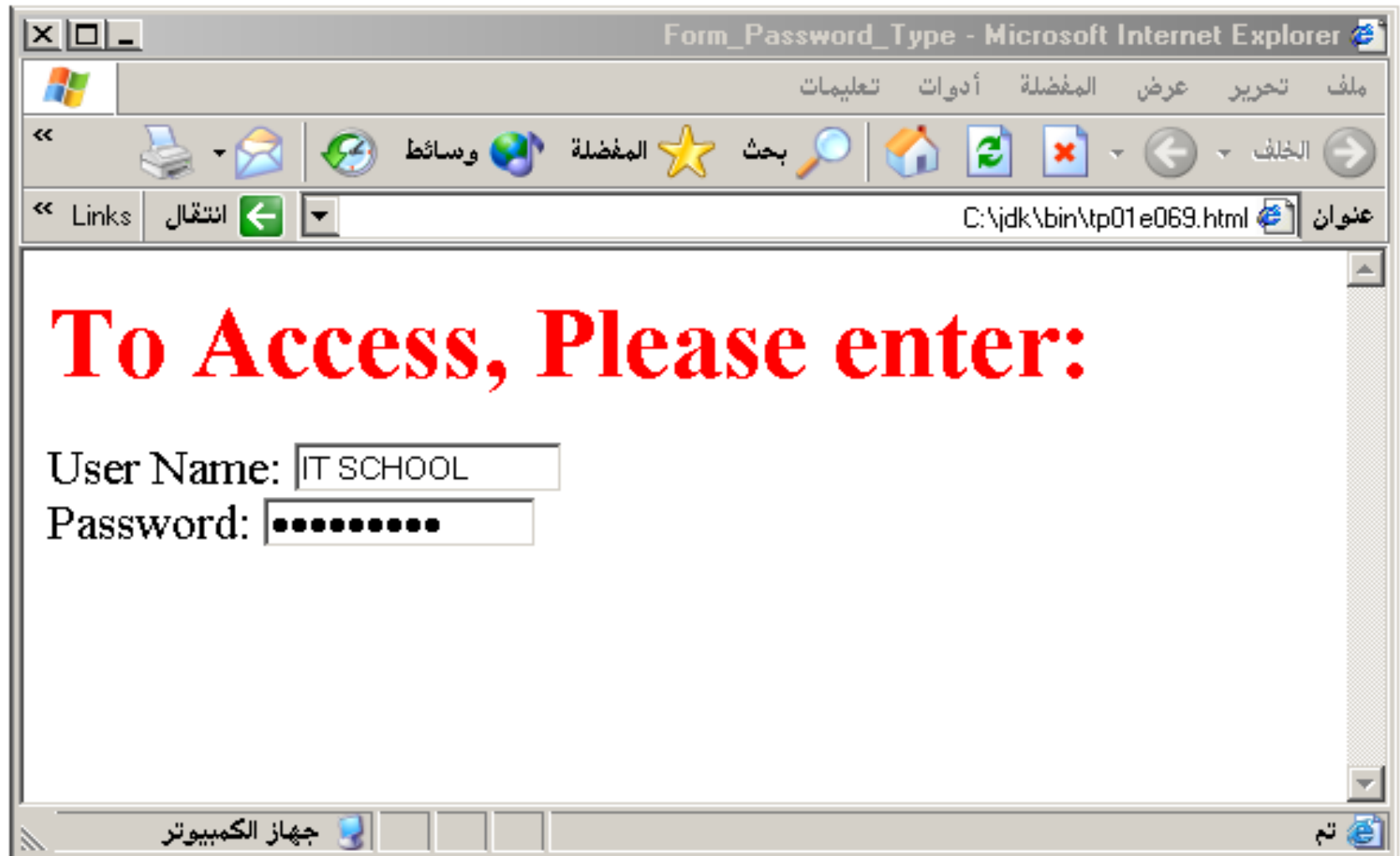
Password boxes use the following attributes:

- **TYPE:** password.
- **SIZE:** determines the size of the textbox in characters.
- **MAXLENGTH:** determines the maximum size of the password in characters.
- **NAME:** is the name of the variable to be sent to the CGI application.
- **VALUE:** is usually blank.

# Example on Password Box

```
<HTML><HEAD>
<TITLE>Form_Password_Type</TITLE></HEAD>
<BODY>
<h1> <font color=red>To Access, Please
enter:</font></h1>
<FORM name="fome2" Action="url" method="get">
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></BODY> </HTML>
```

# Output





# Hidden

- **Hidden:** Used to send data to the CGI application that you don't want the web surfer to see, change or have to enter but is necessary for the application to process the form correctly.

**<INPUT TYPE="HIDDEN">**

**Nothing is displayed in the browser.**

Hidden inputs have the following attributes:

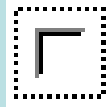
- **TYPE:** hidden.
- **NAME:** is the name of the variable to be sent to the CGI application.
- **VALUE:** is usually set a value expected by the CGI application.

# Check Box

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

**<INPUT TYPE="CHECKBOX">**

Browser will display

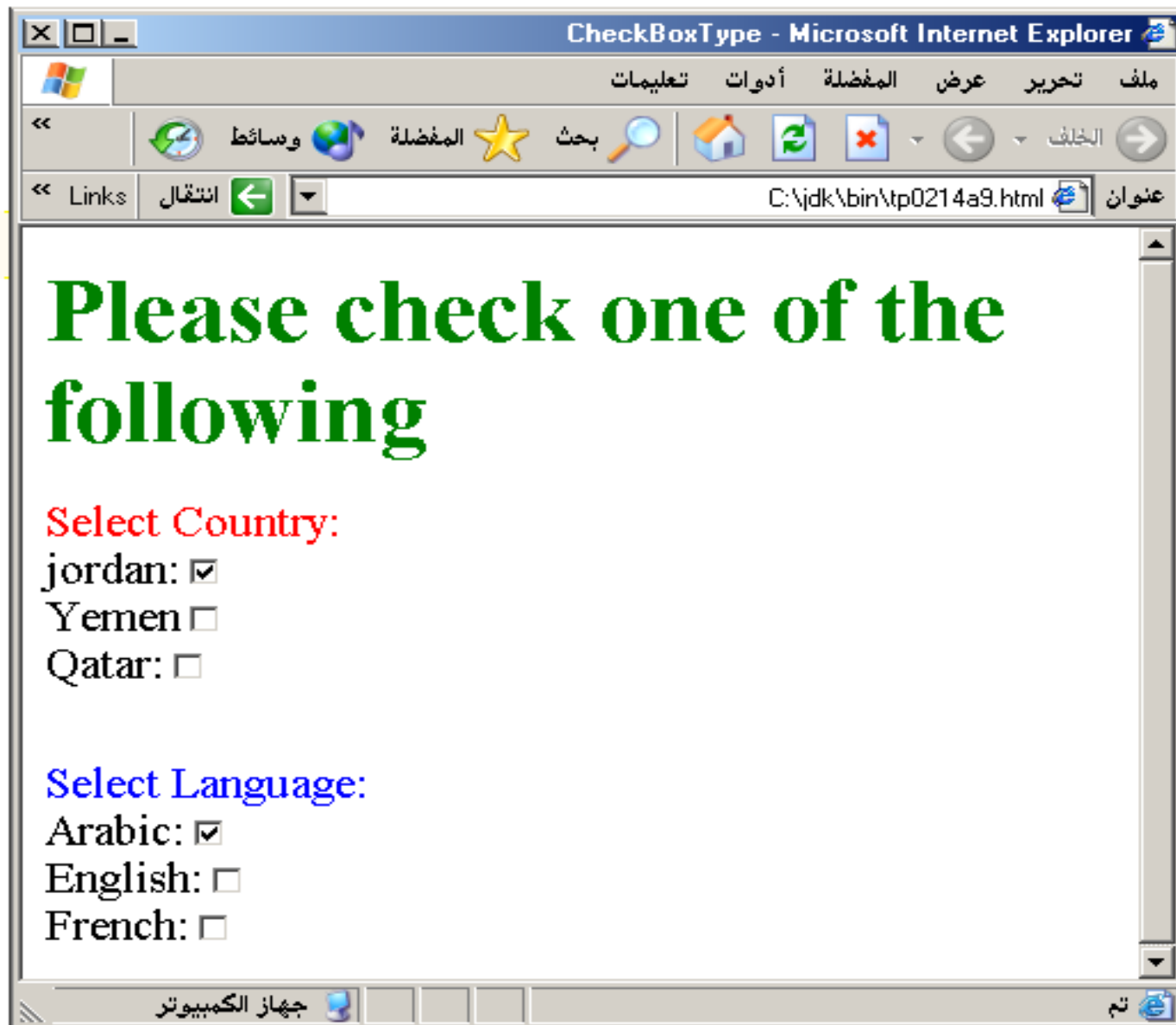


Checkboxes have the following attributes:

- **TYPE:** checkbox.
- **CHECKED:** is blank or CHECKED as the initial status.
- **NAME:** is the name of the variable to be sent to the CGI application.
- **VALUE:** is usually set to a value.

```
<HTML> <HEAD><TITLE>CheckBoxType</TITLE> </HEAD>
<BODY>
<h1> <font color=green>Please check one of the
following</font></h1>
<FORM name="fome3" Action="url" method="get">
<font color=red> Select Country: </font><BR>
jordan:<INPUT TYPE="CheckBox" Name="country"
CHECKED><BR>
Yemen<INPUT TYPE="CheckBox" Name="country"><BR>
Qatar:<INPUT TYPE="CheckBox" Name="country"><BR>
<BR>
<font color=blue>Select Language:</font><BR>
Arabic:<INPUT TYPE="CheckBox" Name="language"
CHECKED><BR> English:<INPUT TYPE="CheckBox"
Name="language"><BR>
French:<INPUT TYPE="CheckBox" Name="language">
<BR></FORM> </BODY></HTML>
```

# Output



# Radio Button

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

**<INPUT TYPE="RADIO">**

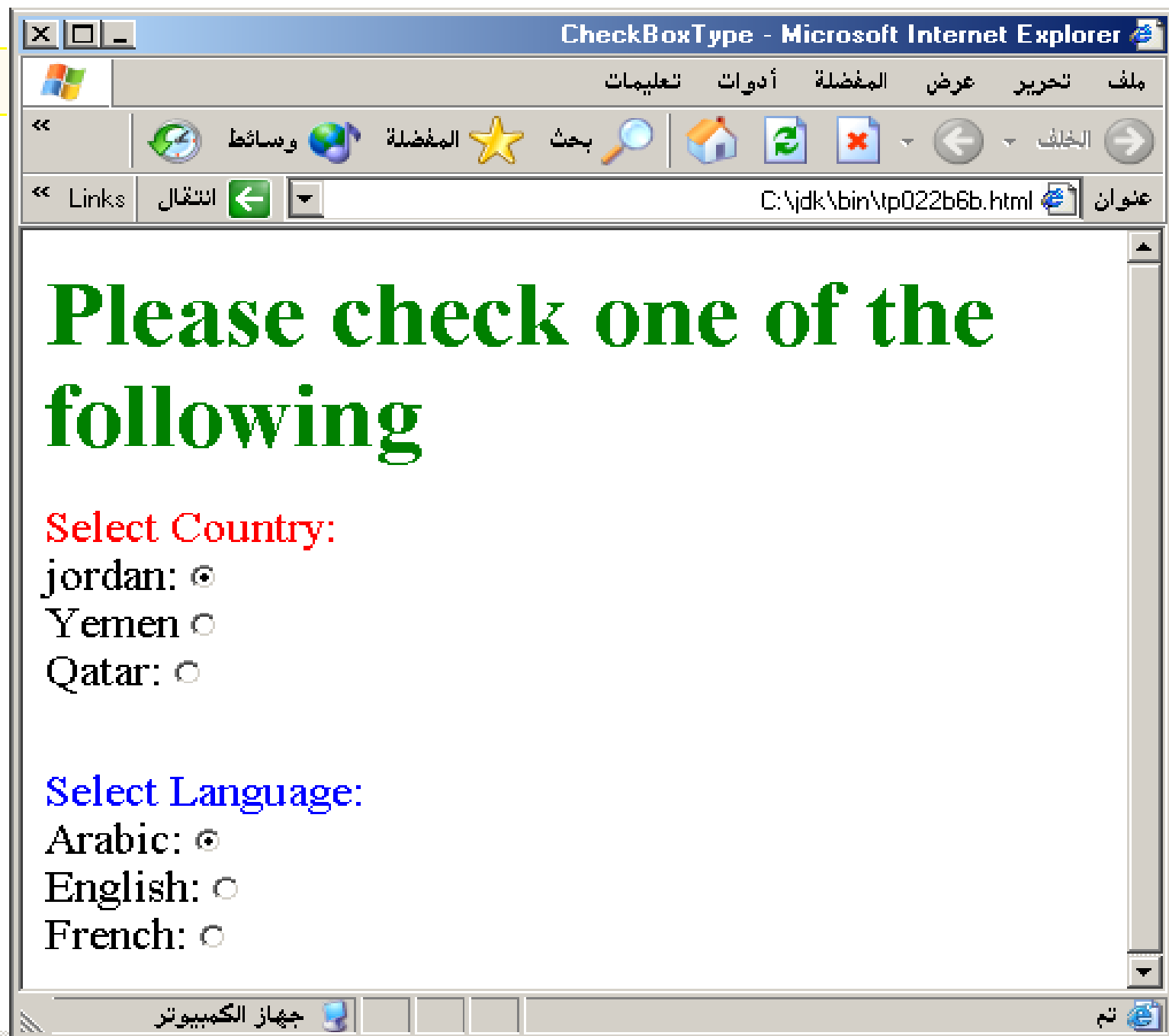
Browser will display



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 to be sent to the CGI application.
- **VALUE**: usually has a set value.

```
<HTML> <HEAD><TITLE>CheckBoxType</TITLE> </HEAD>
<BODY>
<h1> <font color=green>Please check one of the
following</font></h1>
<FORM name="fome3" Action="url" method="get">
<font color=red> Select Country: </font><BR>
jordan:<INPUT TYPE= "RADIO" Name="country"
CHECKED><BR>
Yemen<INPUT TYPE="RADIO " Name="country"><BR>
Qatar:<INPUT TYPE="RADIO" Name="country"><BR>
<BR>
<font color=blue>Select Language:</font><BR>
Arabic:<INPUT TYPE="RADIO" Name="language"
CHECKED><BR> English:<INPUT TYPE=" RADIO "
Name="language"><BR>
French:<INPUT TYPE=" RADIO " Name="language">
<BR></FORM> </BODY></HTML>
```



```
<HTML><HEAD>
<TITLE>RADIOBox</TITLE> </HEAD>
<BODY>
```

### Form #1:

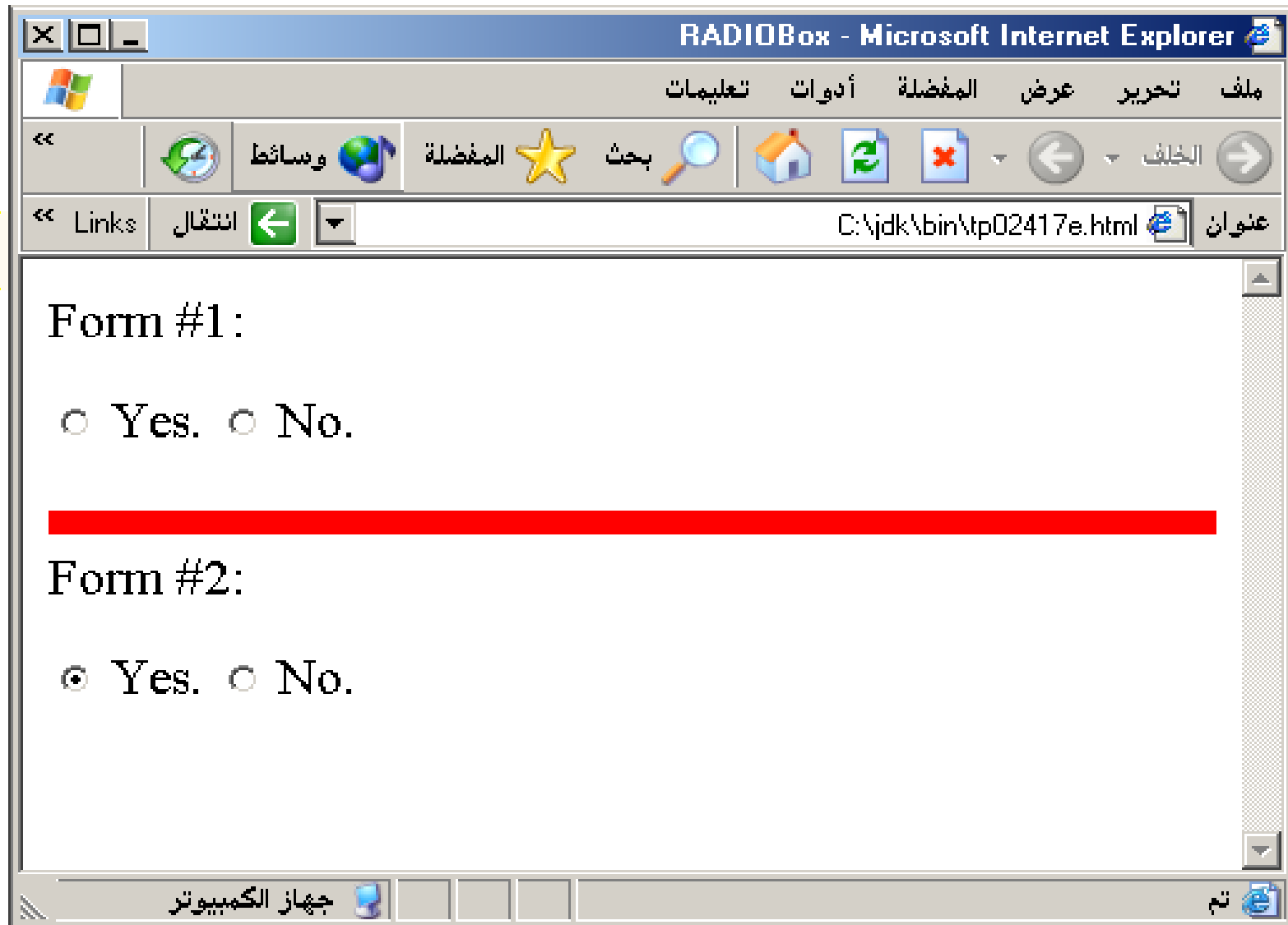
```
<FORM>
  <INPUT TYPE="radio" NAME="choice" VALUE="one"> Yes.
  <INPUT TYPE="radio" NAME="choice" VALUE="two"> No.
</FORM>
<HR color=red size="10" >
```

### Form #2:

```
<FORM>
  <INPUT TYPE="radio" NAME="choice" VALUE="three"
CHECKED> Yes.
  <INPUT TYPE="radio" NAME="choice" VALUE="four"> No.
</FORM>
</BODY></HTML>
```



# Output



# Push Button

- **Push Button:** This element would be used with JavaScript to cause an action to take place.

**<INPUT TYPE="BUTTON">**

Browser will display



Push 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.

**<DIV align=center><BR><BR>**

**<FORM>**

**<FONT Color=red>**

**<h1>Press Here to see a baby crying:<BR>**

**<INPUT TYPE="button"  
VALUE="PressMe"><BR><BR>**

**<FONT Color=blue>**

**Click Here to see a baby shouting:<BR>**

**<INPUT TYPE="button" VALUE="ClickMe" >  
<BR><BR>**

**<FONT Color=green>**

**Hit Here to see a baby eating:<BR>**

**<INPUT TYPE="button" VALUE="HitME" > <BR><BR>**

**<FONT Color=yellow>**

**</FORM></DIV>**



# 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 to the CGI Application specified by the ACTION attribute of the FORM element.

**<INPUT TYPE="SUBMIT">**

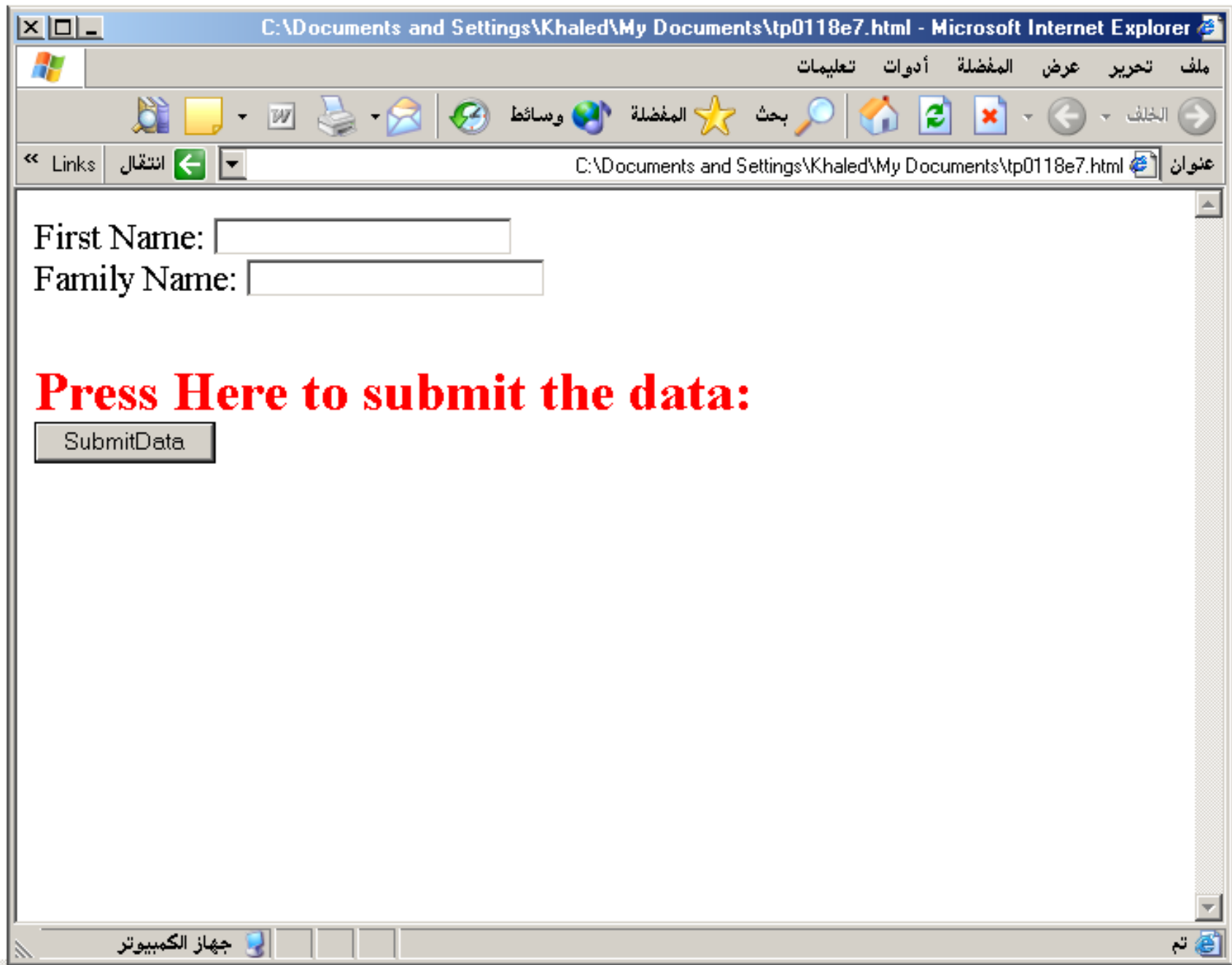
The browser will display



Submit has the following attributes:

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

```
<FORM Action="URL" 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>
```



# 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="URL" 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>
<STRONG><font size=5>Press Here to submit
the data:</font></STRONG><BR>
<INPUT TYPE="submit" VALUE="SubmitData">
<INPUT TYPE="RESET" VALUE="Reset">
</FORM>
```

...C:\Documents and Settings\Khaled\My Documents\tp0125cb.html - Microsoft I

ملف تحرير عرض المفضلة أدوات تعليمات

« » المفضلة وسائط بحث الخلف

« Links انتقال عنوان C:\Documents and Settings\Khaled\My Documents\tp0125cb.html

First Name:

Family Name:

**Press Here to submit the data:**

جهاز الكمبيوتر

# Image Submit Button

- **Image Submit Button:** Allows you to substitute an image for the standard submit button.

**<INPUT TYPE="IMAGE" SRC="jordan.gif">**

Image submit button has the following attributes:

- **TYPE:** Image.
- **NAME:** is the name of the button to be used in scripting.
- **SRC:** URL of the Image file.

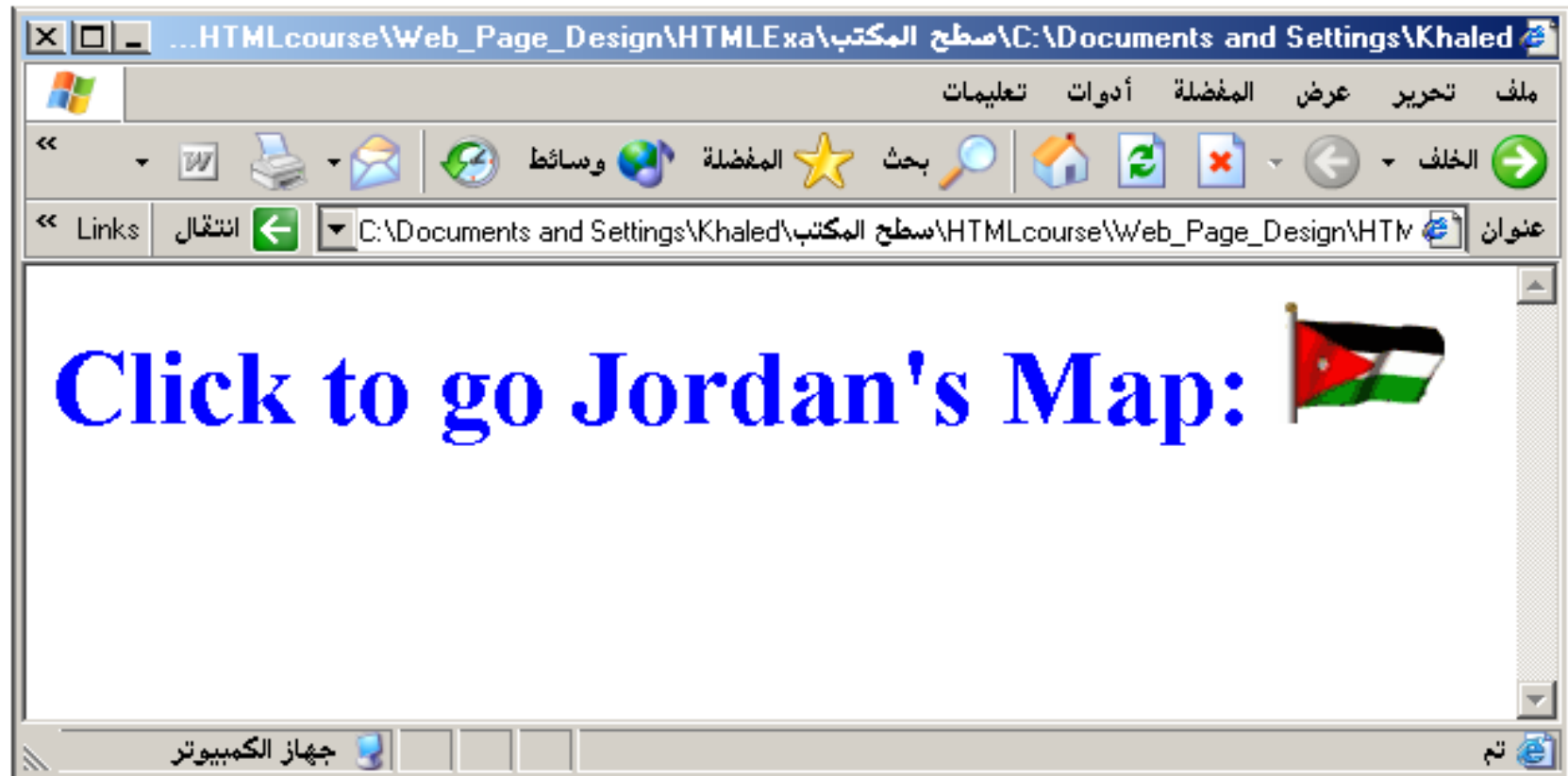
<form>

<H1><font color=blue>


Click to go Jordan's Map:

<INPUT TYPE="IMAGE" SRC="jordan.gif">

</form>



# 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 to be sent to the CGI application.
- **MAXLENGTH:** is the maximum size of the input in the textbox in characters.

**<BODY bgcolor=lightblue>**

**<form>**

**<H3><font color=forestgreen>**

**Please attach your file here to for uploading to  
My <font color =red>SERVER...<BR>**

**<INPUT TYPE="File" name="myFile"  
size="30">**

**<INPUT TYPE="Submit" value="SubmitFile">**

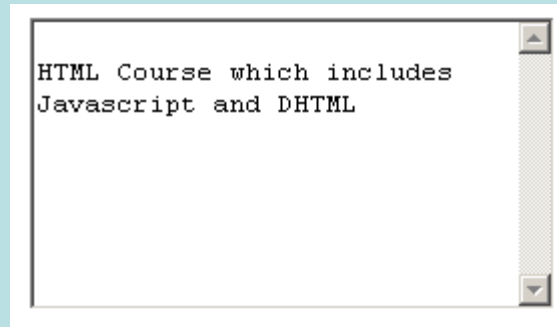
**</form>**

**</BODY>**

# Other Elements used in Forms

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

Browser will display

A screenshot of a web browser window showing a text area. The text area is a rectangular box with a vertical scrollbar on the right side. Inside the box, the text "HTML Course which includes Javascript and DHTML" is displayed in a monospaced font. The text is aligned to the left and occupies the top portion of the text area.

Textarea has the following attributes:

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

**<BODY bgcolor=lightblue>**

**<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.**

**Written by Khaled Al-Fagih**

**</TEXTAREA>:**

**</form>**

**</BODY>**



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.  
Written by Khaled Al-Fagih

### 10.6.1.2 The `wrap` attribute

Normally, text typed in the text area by the user is transmitted to the server exactly as typed, with lines broken only where the user pressed the Enter key. Since this is often not the desired action by the user, you can enable word wrapping within the text area. When the user types a line that is longer than the width of the text area, the browser automatically moves the extra text down to the next line, breaking the line at the nearest point between words in the line.

With the `wrap` attribute set to `virtual`, the text is wrapped within the text area for presentation to the user, but the text is transmitted to the server as if no wrapping had occurred, except where the user pressed the Enter key.

With the `wrap` attribute set to `physical`, the text is wrapped within the text area and is transmitted to the server as if the user had actually typed it that way. This is the most useful way to use word wrap, since the text is transmitted exactly as the user sees it in the text area.

To obtain the default action, set the `wrap` attribute to `off`.

As an example, consider the following 60 characters of text being typed into a 40-character-wide text area:

```
Word wrapping is a feature that makes life easier for users.
```

With `wrap=off`, the text area will contain one line and the user will have to scroll to the right to see all of the text. One line of text will be transmitted to the server.

With `wrap=virtual`, the text area will contain two lines of text, broken after the word "makes."  
Only one line of text will be transmitted to the server: the entire line with no embedded newline characters.

With `wrap=physical`, the text area will contain two lines of text, broken after the word "makes."  
Two lines of text will be sent to the server, separated by a newline character after the word "makes."

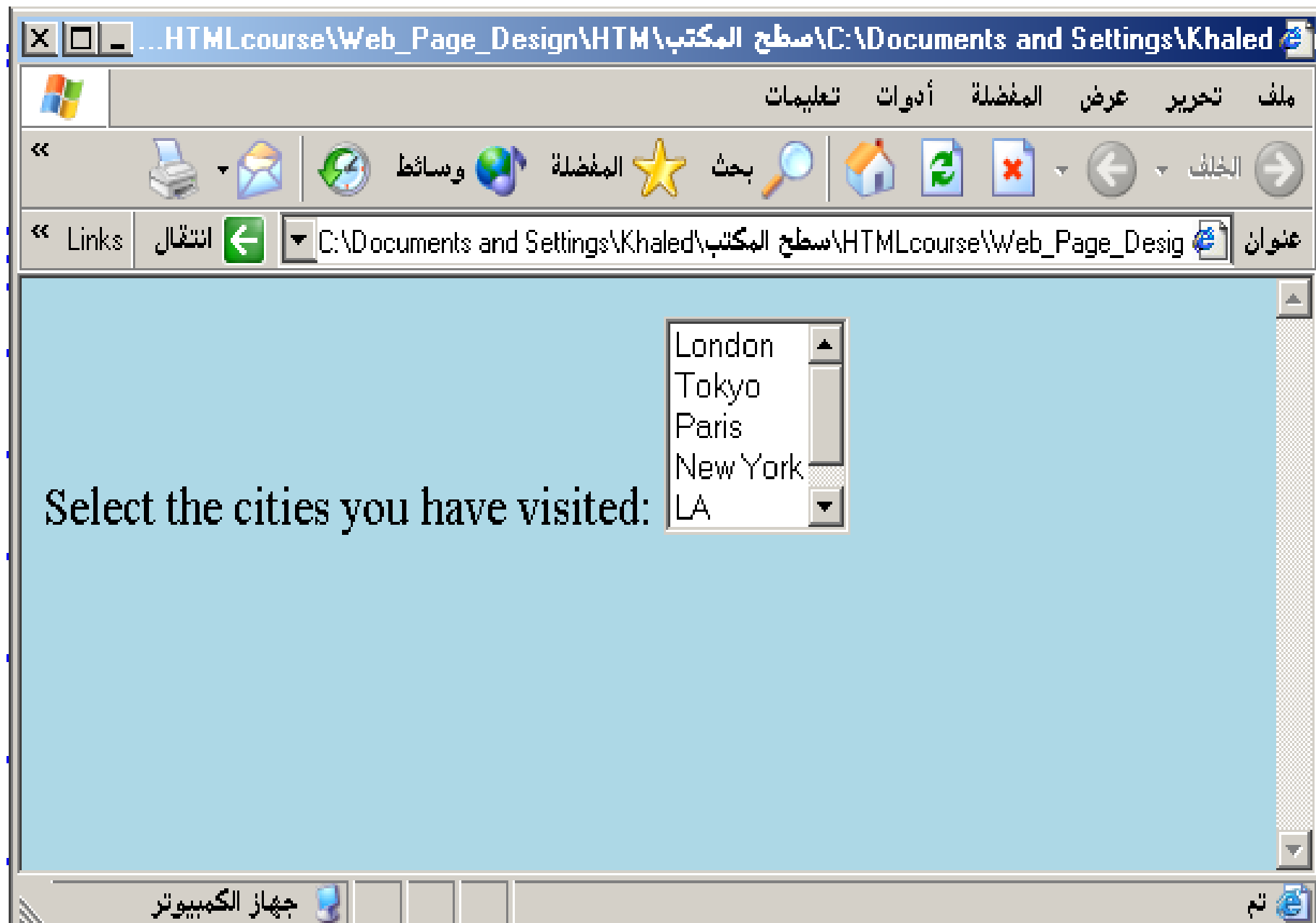
# Other Elements used in Forms

- 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 to be sent to the CGI application.
- **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.

```
<BODY bgcolor=lightblue>
<form>
Select the cities you have visited:
<SELECT name="list" size=5>
<option> London</option>
<option> Tokyo</option>
<option> Paris</option>
<option> New York</option>
<option> LA</option>
<option> KL</option>
</SELECT>
</form>
</BODY>
```



# Other Elements used in Forms

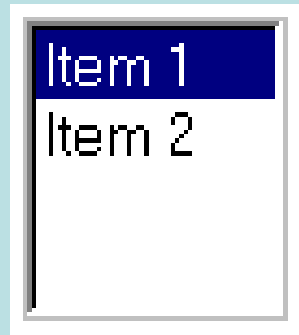
- **Drop Down List:**



- **Name:** is the name of the variable to be sent to the CGI application.
- **Size: 1.**

# Other Elements used in Forms

- **List Box:**



- **Name:** is the name of the variable to be sent to the CGI application.
- **SIZE:** is greater than one.



# 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.

**</HEAD>**

**<BODY>**

**<h2><font color=blue>What type of Computer do you have?</font><h2>**

**<FORM>**

**<SELECT NAME="ComputerType" size=4>**

**<OPTION value="IBM" SELECTED> IBM</OPTION>**

**<OPTION value="INTEL"> INTEL</OPTION>**

**<OPTION value=" Apple"> Apple</OPTION>**

**<OPTION value="Compaq"> Compaq</OPTION>**

**</SELECT>**

**</FORM></BODY></HTML>**

# What type of Computer do you have?

IBM  
INTEL  
Apple  
Compaq

```
<HEAD> <TITLE>SELECT with Mutiple </TITLE>  
</HEAD>
```

```
<BODY>
```

```
<h2><font color=blue>What type of Computer do you  
have?</font><h2>
```

```
<FORM>
```

```
<SELECT NAME="ComputerType" size=5 multiple>
```

```
<OPTION value="IBM" > IBM</OPTION>
```

```
<OPTION value="INTEL"> INTEL</OPTION>
```

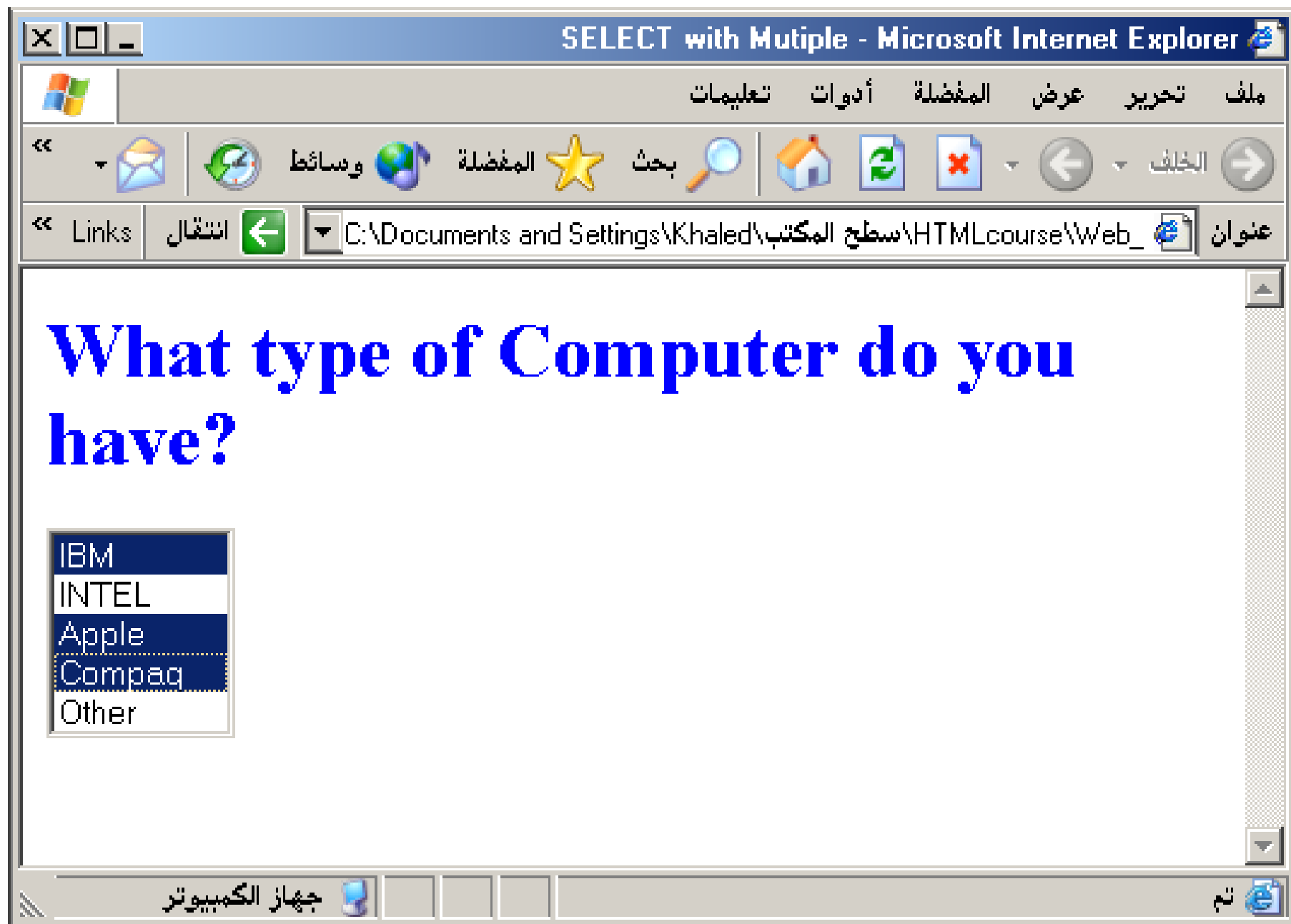
```
<OPTION value=" Apple"> Apple</OPTION>
```

```
<OPTION value="Compaq" SELECTED>  
Compaq</OPTION>
```

```
<OPTION value=" other"> Other</OPTION>
```

```
</SELECT>
```

```
</FORM></BODY></HTML>
```



There are eleven different types of form elements:

Button

Checkbox

☐

FileUpload

Hidden

Password

Radio

☐

Reset object

Select object

Submit object

Text

Textarea