

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

ІНФОРМАТИКА 2: ОСНОВИ HTML

навч. посіб. англійською мовою: конспект лекцій для студентів спеціальності
051 «Економіка» освітньої програми «Міжнародна економіка»

*Рекомендовано Вченою радою ФММ
КПІ ім. Ігоря Сікорського
як навчальний посібник для студентів
спеціальності 051 «Економіка»
освітньої програми «Міжнародна економіка»*

Київ - 2019

Рецензент *Підлісна О.А.*, к.е. наук, доцент

Відповідальний редактор: *Фартушний І.Д.*, канд. фіз.-мат. наук, доцент

*Гриф надано Вченою радою ФММ
(Протокол № 8 від 01 квітня 2019 р.)*

Електронне мережне навчальне видання

Стець Олена Вікторівна, канд. фіз.-мат. наук, доц.

ІНФОРМАТИКА 2 (ОСНОВИ HTML)

навч. посіб. англійською мовою: конспект лекцій для студентів спеціальності 051 «Економіка» освітньої програми «Міжнародна економіка»

Інформатика 2:основи HTML: англійською мовою [Електронний ресурс]: навч. посіб. конспект лекцій для студентів спеціальності 051 «Економіка» освітньої програми «Міжнародна економіка». Уклад.: Стець О.В. – Електронні текстові дані (1 файл: 3,3 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 146 с.

У посібнику наведено основні засади теоретичних знань і практичних навичок володіння сучасними засобами інформаційних технологій та створення web-ресурсів на базі мови гіпертекстової розмітки HTML як інструменту для формування відповідних інформаційних систем. Детально розглянуто основні структурні елементи мови з прикладами їх створення та використання.

Для студентів першого бакалаврського рівня вищої освіти студентів спеціальності 051 «Економіка» освітньої програми «Міжнародна економіка»

© О.В. Стець, 2019

© КПІ ім. Ігоря Сікорського, 2019

3MICT

Introduction to Hyper Text Markup Language.....	4
Tags and Attributes in HTML.....	21
Describes the structure of Web pages using markup.....	37
Elements are the building blocks of HTML pages.....	56
Tags label pieces of content such as "heading", "paragraph", "table", and so on.....	78
The Form shell.....	100
Frame structure.....	120

INTRODUCTION TO HTML

Part 1



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 (Examples: Internet Explorer, Chrome, FireFox, Opera, etc.).



WHAT IS HTML?

- **HTML**, otherwise known as HyperText Markup Language, is the language used to create Web pages
- Using **HTML**, you can create a Web page with text, graphics, sound, and video



HTML DOCUMENTS

- Plain text files, with extension [htm](#) or [html](#).
- The extension tells the browser interpret the file according to HTML standards.
- Use any plain text editor to create html files manually.
- Browsers may display other types of files as well, according to the file's extension.
 - [TXT](#) - .txt - plain text file, no formatting.
 - [PDF](#) - .portable document format (Adobe corporation)
 - [JPG](#) - .jpg - JPEG image file.



TAGS

- The essence of HTML programming is tags
- A tag is a keyword enclosed by angle brackets (Example: `<I>`)
- There are opening and closing tags for many but not all tags; The affected text is between the two tags



STRUCTURAL TAGS

`<HTML>`

These tags enclose the entire Web page document.

`</HTML>`

`<HEAD>`

These tags enclose the Head part of the document

`</HEAD>`

`<TITLE>`

These tags enclose the title of the document. This text appears in the title bar in the browser and on the bookmark list if someone bookmarks your web page.

`</TITLE>`



STRUCTURE OF A WEB PAGE

- All Web pages share a common structure
- All Web pages should contain a pair of `<HTML>`, `<HEAD>`, `<TITLE>`, and `<BODY>` tags
- DOCTYPE - A pre-processor directive at the top of the document tells the browser what kind of standards apply to the document

```
<!DOCTYPE>
```

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Example </TITLE>
```

```
</HEAD>
```

```
<BODY>
```

This is where you would include the text and images on your Web page.

```
</BODY>
```

```
</HTML>
```



SAMPLE STRUCTURE OF A WEB SITE

```
<HTML>  
  <HEAD>  
    <TITLE> John Q. Public's Web Page </TITLE>  
  </HEAD>  
  
  <BODY>  
    This is John Public's Webpage!  
  </BODY>  
</HTML>
```



HEADER TAGS

Header Tags -- Used for marking sections and subsections in a document.

`<H1>`Header 1 -- Giant-sized and bold `</H1>`

`<H2>`Header 2 -- Large and bold `</H2>`

`<H3>`Header 3 -- Normal-sized and bold `</H3>`

`<H4>`Header 4 -- Small and bold `</H4>`

`<H5>`Header 5 -- Very Small and bold `</H5>`

`<H6>`Header 6 -- Tiny and bold `</H6>`



HEADER TAGS (CONT.)

H1 = Giant-sized and bold

H2 = Large and bold

H3 = Normal-sized and bold

H4 = Small and bold

H5 = Very Small and bold

H6 = Tiny and bold



BREAKING LINES AND PARAGRAPHS

- `<P> text </P>`
 - Paragraph tag
 - Most browsers render (process) this with blank lines between each paragraph
 - `
`
 - Line break tag
 - Used when the webmaster wants a carriage return but doesn't want a blank line to follow
-

Example:

```
<p>text a</p>
```

```
<p>text b</p>
```

```
<br>text c
```

```
<br>text d
```



text a

text b

text c

text d



TEXT FORMATTING TAGS

Some basic text formatting styles:

Tag	Result
<code><I> Italics </I></code>	<i>Italics</i>
<code> Bold </code>	Bold
<code><PRE> Preformatted Text </PRE></code> Text	Preformatted
<code> Strong </code>	Strong
<code><ADDRESS> Address </ADDRESS></code>	<i>Address</i>
<code><CITE> Citations </CITE></code>	<i>Citations</i>
<code><CODE> Source Code </CODE></code>	Source Code



FONT MODIFICATIONS

Web creators can also change the way text looks by using the tag

SIZE="number" - changes size of the font; 1=smallest, 7 = largest

Big Small

Big Small

COLOR="color-name" - changes text color

This is red

This is red

FACE="font-name" - changes font

This is the verdana font; this is the chicago font.

This is the verdana font; this is chicago font.



 MODIFICATIONS (CONT.)

One can combine font modifications:

```
<FONT SIZE="7" FACE="courier" COLOR="red">Big, Courier & Red</FONT>
```

Big, Courier & Red

```
<FONT SIZE="7"><FONT FACE="courier">Big & Courier</FONT> - Just  
Big</FONT>
```

Big & Courier - Just Big



LISTS -- UNORDERED LISTS

Unordered lists:

Item One

Item Two

Item Three

Item Four



• Item One

• Item Two

• Item Three

• Item Four

Unordered List Attributes:

type="disc/circle/square"

• Disc (default) ○ Circle ■ Square



LISTS -- ORDERED LISTS

Ordered (Numbered) Lists:

 Item One

 Item Two

 Item Three

 Item Four



1. Item One

2. Item Two

3. Item Three

4. Item Four

Ordered List Attributes:

type="i/I/a/A/1"

(default)

i = i. Item One

I = I. Item One

a = a. Item One

A = A. Item One

1 = 1. Item One

ii. Item Two

II. Item Two

b. Item Two

B. Item Two

2. Item Two

iii. Item Three

III. Item Three

c. Item Three

C. Item Three

3. Item Three

iv. Item Four

IV. Item Four

d. Item Four

D. Item Four

4. Item Four

start="xx"

- This attribute lets you specify which number/letter will start the list




LISTS -- DEFINITION LISTS

Definition Lists:

```
<DL>
  <DT>List Name One
    <DD>This is where information about List Name One would
go</DD>
  </DT>
  <DT>List Name Two
    <DD>This is where information about List Name Two would
go</DD>
  </DT>List Name One
</DL>
      This is where information about List Name One
      would go

      List Name Two

      This is where information about List Name Two
      would go
```



THE TASK

- create a page about yourself using all the commands for formatting text, list, headings, etc.



INTRODUCTION TO HTML

Part 2



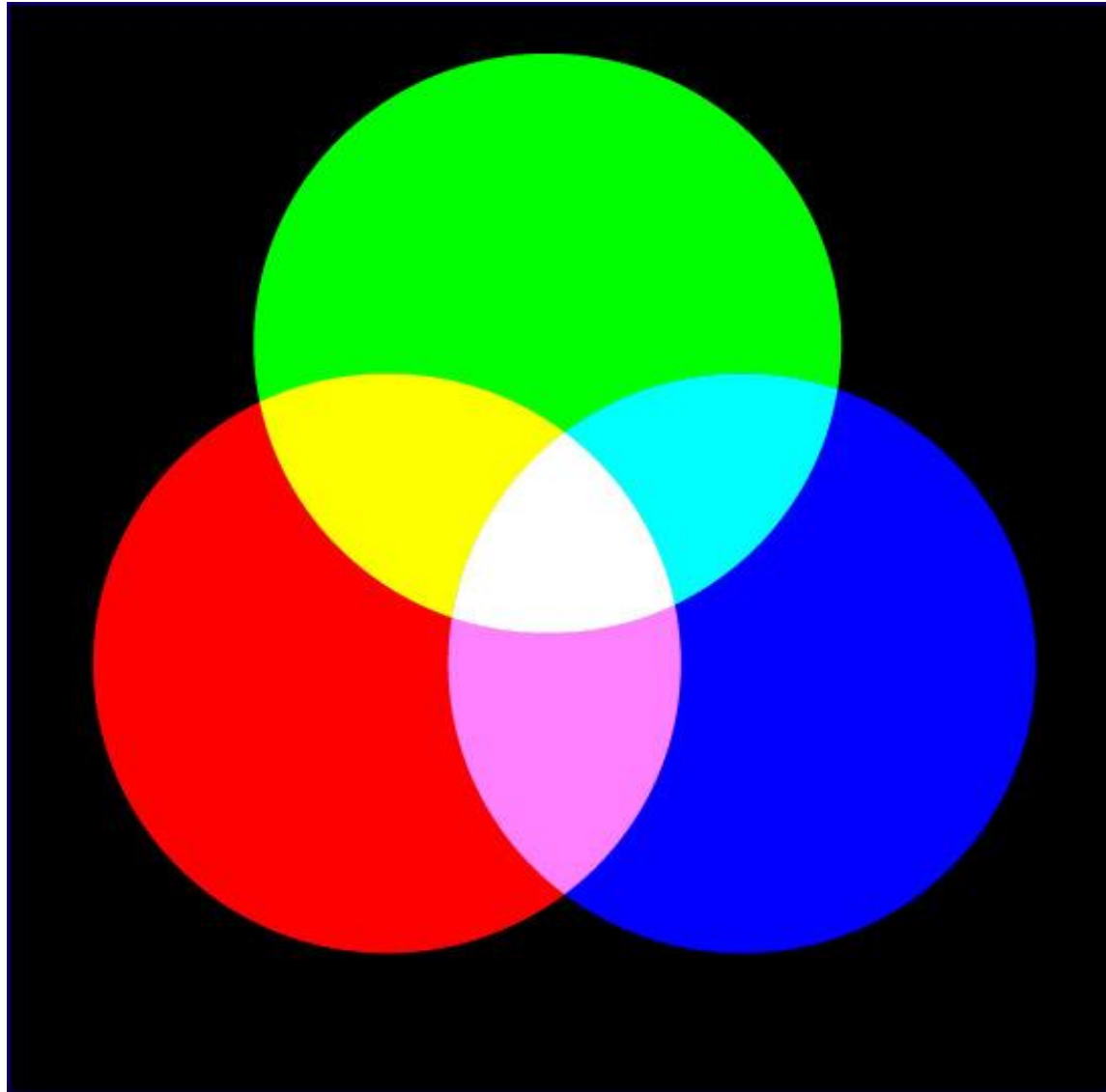
Each html-page uses colors and images



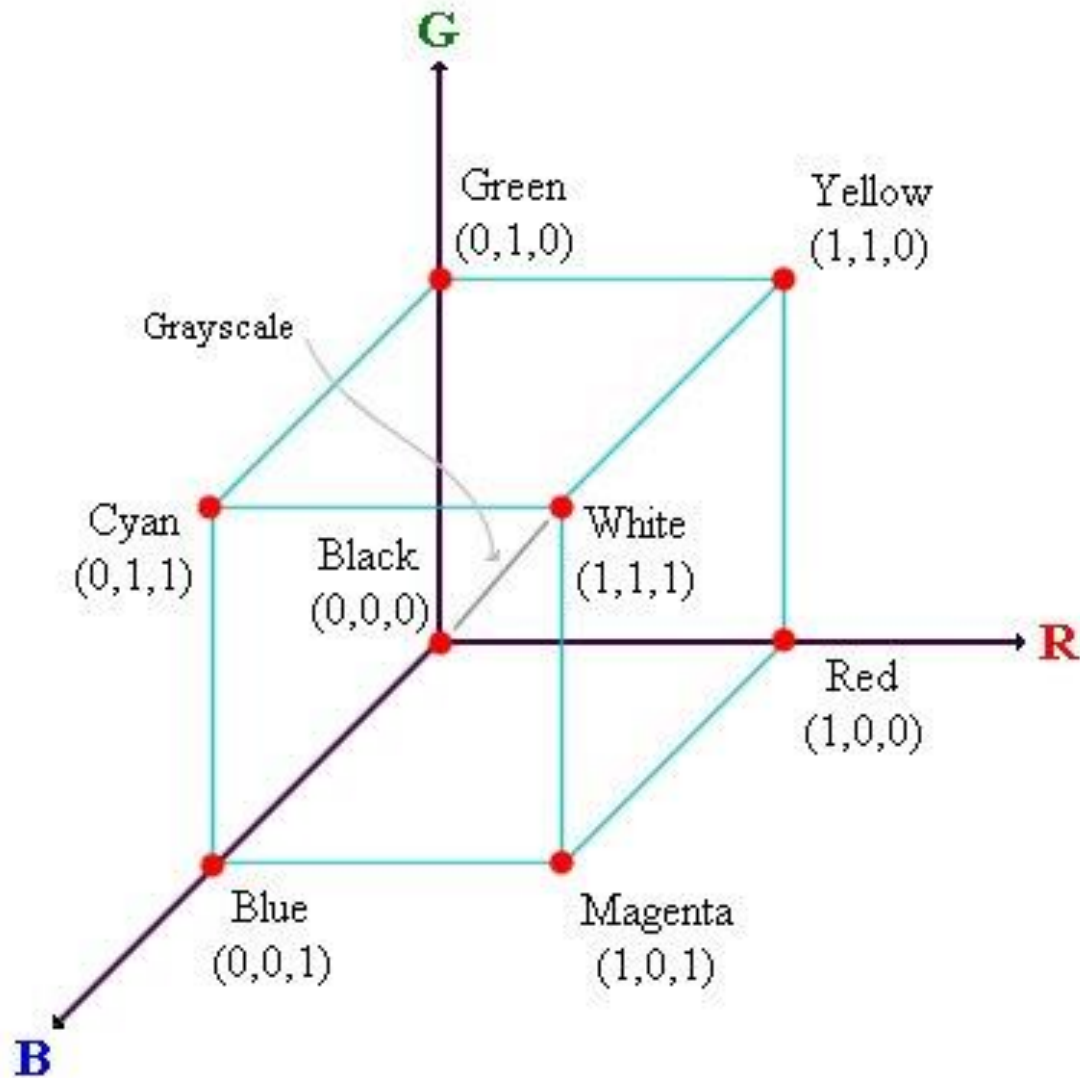
COLOR CODES

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. For example, a **RGB** value with 00 as the first two digits has no red in the color.

MAIN COLOURS



RGB COLOUR MODEL



16 BASIC COLORS

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

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.

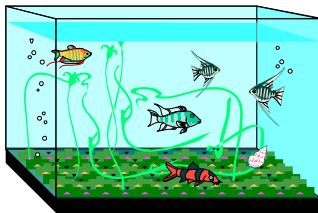
An example of a background image's HTML code is as follows:

```
<BODY BACKGROUND="foto.jpg">.....</BODY>
```

GRAPHICS

To have a graphic appear on a webpage, web designers must to put the `` tag in with the address where the graphic "lives":

```
<IMG SRC="fish.gif">
```



GRAPHICS

If in the tag we write only the name of the file with the image, it means that this file should be located in the same folder as our HTML-file

Otherwise, you need to write the full path to the file

For example: " C:\MyImages\fish.gif"



GRAPHICS

Graphics attributes:

alt="text": insert a description of the graphic for those who are using browsers that cannot process images (e.g., page readers for the blind)

width="xx/xx%": width in pixels/percentage

height="xx/xx%": height in pixels/percentage

border="xx": pixel length of the border surrounding the image.

hspace="xx": places a buffer of space horizontally around the image

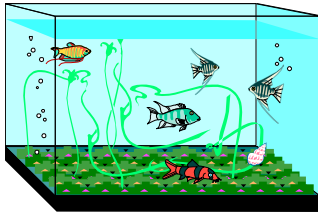
vspace="xx": places a buffer of space vertically around the image

align="top/middle/bottom/right/left": aligns image in relation to the text (see next 2 slides)



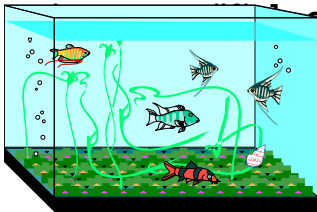
GRAPHICS

`All about Fish`



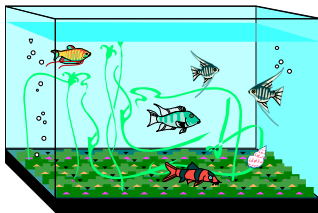
All about Fish

`"gif" align="middle">All about Fish`



All about Fish

`"if" align="bottom">All about Fish`

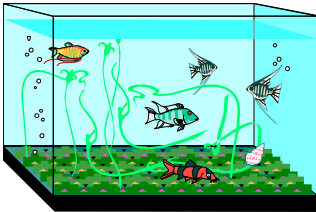


All about Fish

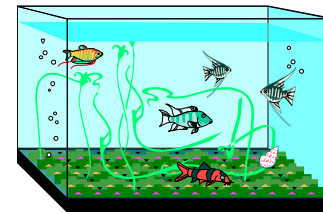


GRAPHICS

``



``



E.G. 1

- Loads a jpeg* file named “selfie01” into the document at the location of the element.
- JPEG - Joint Photographic Experts Group compression format – file extension, .jpg



E.G. 2

```
<img src = “selfie01.gif” align =  
“right”>
```

- Loads a gif* file named “selfie01” into the document at the location of the element, and moves it to the right hand side of the page.
- GIF - Graphics Interchange Format compression format - file extension, .gif



E.G. 3

```
<img src = "selfie01.bmp" align = "middle" height = "100" width =  
"200">
```

- Loads a *.bmp file into the document, centers it vertically, relative to the line, and gives it a height of 100 pixels and a width of 200 pixels.
- This may distort the original proportions of the image. BMP- BitMap: simpler non-compressed graphics image format. - file extension, .bmp



E.G. 4

```
<img src = “selfie01.jpg” alt = “Sorry,  
no picture available!”>
```

- Loads the image “selfie01.jpg” if available.
- If for some reason the image can not be loaded, the browser should display the text “**Sorry, no picture available!**” in its place.
- This facility was originally intended to accommodate text-only browsers.



ANIMATED GIF FILE

- A *.gif image file which is composed of a series of frames or images, simulates motion by displaying the images in the file consecutively.
- It is an electronic equivalent of an older style celluloid motion picture.



THE TASK

Try to create pages using all the considered tags and attributes for working with colors and images



INTRODUCTION TO HTML

Part 3



THE ANCHOR ELEMENT

- The **anchor** element, the single letter **a**, is used to connect or link different documents or parts of one document.
- Any text associated with the anchor element, called **anchor text** is displayed on the screen.
- When an anchor text is clicked, a '**jump**' is made to the destination or target location.



HYPertext LINKS

- Used to create a link that jumps from one point in a document to another document.
- Destination may be **external** (another web site) or **local** (a document in the local machine).



GENERAL FORM:

** anchor text **



LINKS

- A link lets you move from one page to another, play movies and sound, send email, download files, and more....
- A link has three parts: a **destination**, a **label**, and a **target**
- To create a link type
` label `



ANATOMY OF A LINK

` label `

- In the above link, “page.html” is the destination. The destination specifies the address of the Web page or file the user will access when he/she clicks on the link.
- The label is the text that will appear underlined or highlighted on the page



EXAMPLE: LINKS

To create a link to KPI, I would type:

```
<A HREF="https://kpi.ua/">KPI</A>
```



CHANGING THE COLOR OF LINKS

- The `LINK`, `VLINK`, and `ALINK` attributes can be inserted in the `<BODY>` tag to define the color of a link
 - `LINK` defines the color of links that have not been visited
 - `VLINK` defines the color of links that have already been visited
 - `ALINK` defines the color of a link when a user clicks on it



USING LINKS TO SEND EMAIL

- To create a link to an email address, type

` Label`

- For example, to create a link to send email to myself, I would type:

`email `



ANCHORS

- Anchors enable a user to jump to a specific place on a Web site
- Two steps are necessary to create an anchor. First you must create the anchor itself. Then you must create a link to the anchor from another point in the document.



ANCHORS

- To create the anchor itself, type

```
<A NAME="anchor name">label</A>
```

at the point in the Web page where you want the user to jump to

- To create the link, type

```
<A HREF="#anchor name">label</A>
```

at the point in the text where you want the link to appear



INTRODUCTION TO HTML

Part 4



TABLES

- Tables can be used to display rows and columns of data, create multi-column text, captions for images, and sidebars
- The `<TABLE>` tag is used to create a table; the `<TR>` tag defines the beginning of a row while the `<TD>` tag defines the beginning of a cell



ADDING A BORDER

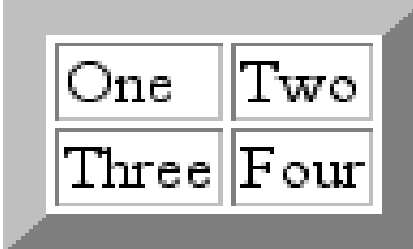
- The `BORDER=n` attribute allows you to add a border `n` pixels thick around the table
- To make a solid border color, use the `BORDERCOLOR="color"` attribute
- To make a shaded colored border, use `BORDERCOLORDARK="color"` and `BORDERCOLORLIGHT="color"`



CREATING SIMPLE TABLE

```
<TABLE BORDER=10>  
  <TR>  
    <TD>One</TD>  
    <TD>Two</TD>  
  </TR>  
  <TR>  
    <TD>Three</TD>  
    <TD>Four</TD>  
  </TR>  
</TABLE>
```

Here's how it would
look on the Web:



One	Two
Three	Four



ADJUSTING THE WIDTH

- When a Web browser displays a table, it often adds extra space. To eliminate this space use the `WIDTH =n` attribute in the `<TABLE>` and `<TD>` tags
- Keep in mind - a cell cannot be smaller than its contents, and if you make a table wider than the browser window, users will not be able to see parts of it.



CENTERING A TABLE

- There are two ways to center a table
 - Type `<TABLE ALIGN=CENTER>`
 - Enclose the `<TABLE>` tags in opening and closing `<CENTER>` tags



WRAPPING TEXT AROUND A TABLE

- It is possible to wrap text around a table. This technique is often used to keep images and captions together within an article.
- To wrap text around a table, type
`<TABLE ALIGN = LEFT>`
to align the table to the left while the text flows to the right
- Create the table using the `<TR>`, `<TD>`, and `</TABLE>` tags as you normally would



ADDING SPACE AROUND A TABLE

- To add space around a table, use the HSPACE=n and VSPACE=n attributes in the <TABLE> tag
- Example:
<TABLE HSPACE=20 VSPACE=20>



SPANNING CELLS ACROSS COLUMNS

- It is often necessary to span one cell across many columns. For example, you would use this technique to span a headline across the columns of a newspaper article.
- To span a cell across many columns, type `<TD COLSPAN=n>`, where n is the number of columns to be spanned



SPANNING CELLS ACROSS ROWS

- To span a cell across many rows, type `<TD ROWSPAN=n>`, where n is the number of rows



SPANNING CELLS ACROSS ROWS

```
<TABLE border=4 cellspacing=0 >
<CAPTION>Table </caption>
<TR><TD bgcolor="white"><B> cell 1</b>
<TD bgcolor="white"><B> cell 2</b>
<TR><TD rowspan=3 bgcolor="white"> 1
<TD> cell 3
<TR><TD>cell 4
<TR><TD>cell 5
<TR><TD colspan=2 bgcolor="white"
align="center"> All
</TABLE>
```

Table

cell 1	cell 2
1	cell 3
	cell 4
	cell 5
All	



ALIGNING CELL CONTENT

- By default, a cell's content are aligned horizontally to the left and vertically in the middle.
- Use `VALIGN=direction` to change the vertical alignment, where “direction” is top, middle, bottom, or baseline
- Use `ALIGN=direction` to change the horizontal alignment where “direction” is left, center, or right



CONTROLLING CELL SPACING

- Cell spacing is the space *between* cells while cell padding is the space *around* the contents of a cell
- To control both types of spacing, use the `CELLSPACING =n` and `CELLPADDING=n` attributes in the `<TABLE>` tag



CHANGING A CELL'S COLOR

- To change a cell's color, add the `BGCOLOR="color"` attribute to the `<TD>` tag
- Example:
`<TD BGCOLOR="blue">`



DIVIDING YOUR TABLE INTO COLUMN GROUPS

- You can divide your table into two kinds of column groups: structural and non-structural.
- Structural column groups control where dividing lines are drawn; Non-structural groups do not
- Both let you format an entire column of cells at once



COLUMN GROUPS

- To create structural column groups, type `<COLGROUP SPAN=n>` after the `<TABLE>` tag, where `n` is the number of columns in the group
- To create non-structural column groups, type `<COL SPAN=n>`, where `n` is the number of columns in the group



DIVIDING TABLE INTO HORIZONTAL SECTIONS

- You can also create a horizontal section consisting of one or more rows. This allows you to format the rows all at once
- To create a horizontal section, type `<THEAD>`, `<TBODY>`, or `<TFOOT>` before the first `<TR>` tag of the section
- Netscape does not support these tags





CONTROLLING LINE BREAKS

- Unless you specify otherwise a browser will divide the lines in a cell as it sees fit.
- The NOWRAP attribute placed within the `<TD>` tag forces the browser to keep all the text in a cell on one line
- Example:
 - `<TD NOWRAP>Washington, D.C.`



THE TASK

create the Table like this :

Number	Cell	
	Point	Photo
1	point 1	
	point 1	
	point 1	
2	point 2	
	point 2	
	point 2	



INTRODUCTION TO HTML

Part 5



MAP

- An imagemap is a graphic image where a user can click on different parts of the image and be directed to different destinations. imagemaps are made by defining each of the hot areas in terms of their x and y coordinates (relative to the top left hand corner). With each set of coordinates, you specify a link that users will be directed to when they click within the area.
- As an example, say you have a map of the World that you wish to act as an image map. Each country could have their hot areas defined on the map to take you to different pages.



MAP

`<MAP>...</MAP>`

- The HTML `<map>` tag is used for declaring an image map
- The `<map>` tag is used in conjunction with the `<area>` tag and `` tag to specify clickable areas (sometimes referred to as "hot spots") on an image.



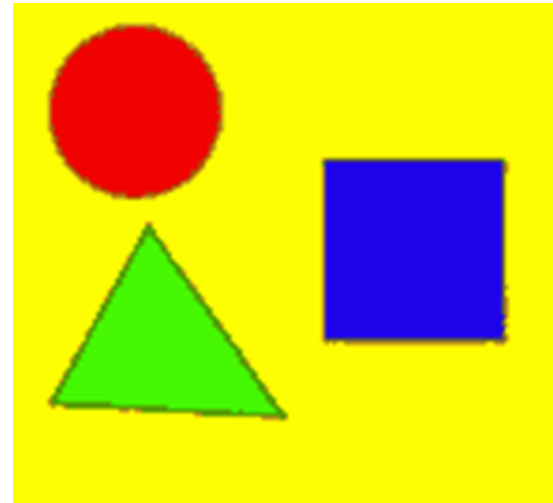
MAP

- The MAP element defines a set of graphic image areas in which various sites can refer to different resources.
- Areas are specified by the `<AREA> ... </AREA>` tags and the shape attribute.
- The coordinates of image points can be determined using any graphic editor (for example, Paint)



MAP

- **For example, I want to create a map on the image [map.gif](#)**



MAP

There are three types of contours
shape:

circle - the coordinates of the center and the radius are given

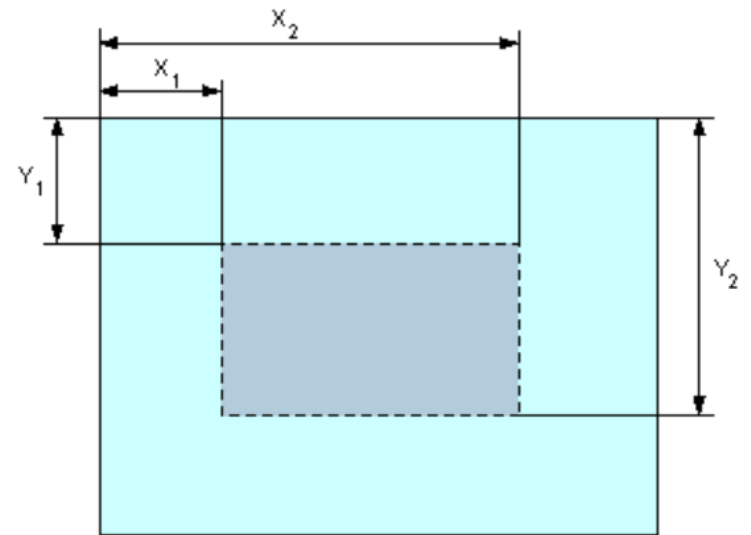
poly - the coordinates of all the points are given

Rect - the coordinates of the upper left and right lower vertices are given



MAP

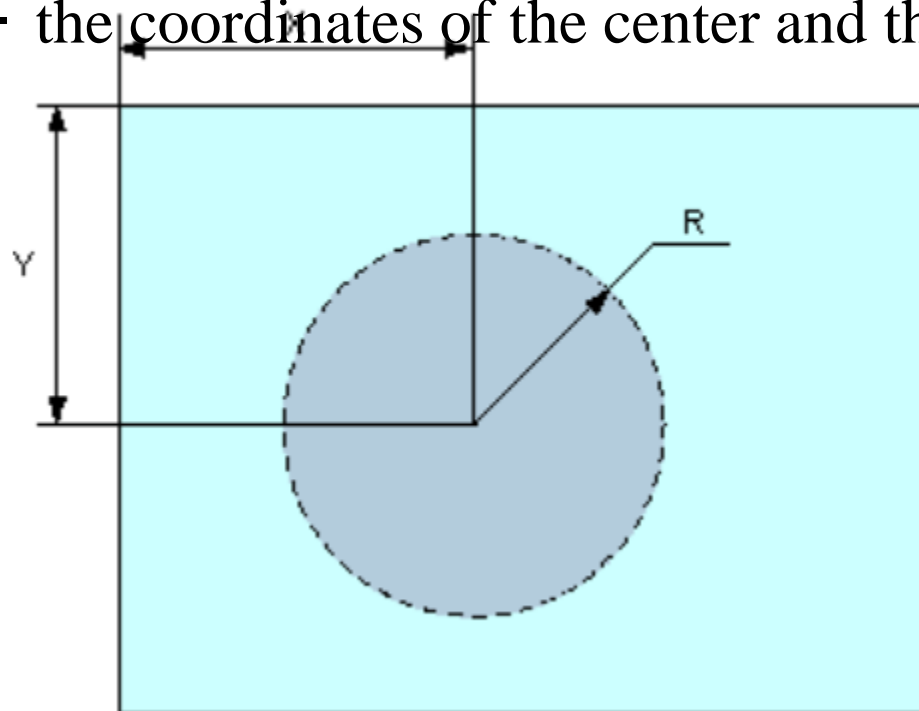
Rect - the coordinates of the upper left and right lower vertices are given



MAP

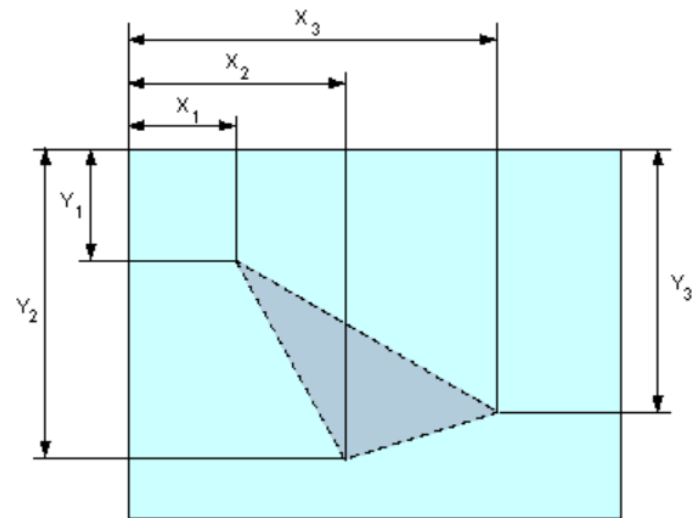
circle
given

- the coordinates of the center and the radius are



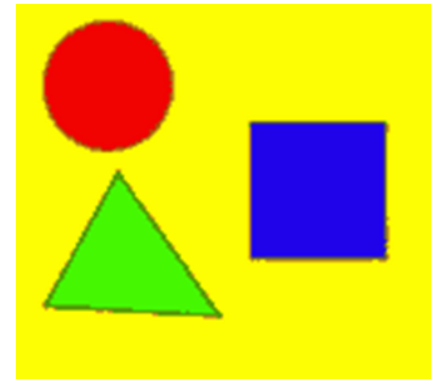
MAP

poly - the coordinates of all the points are given



CREATING SIMPLE MAP

- ``
`<MAP name="map.gif">`
`<AREA shape=circle coords="34,32,23"`
`href="page1.html" alt="ref 1">`
`<AREA shape=poly coords="12,110,37,62,72,114"`
`href="page2.html" alt=" ref 2">`
`<AREA shape=rect coords="83,44,133,94"`
`href="page3.html" alt=" ref 3">`
`</MAP>`



THE TASK

Create an image map using all types of contours.

Links are executed on any sites and files of previous tasks



INTRODUCTION TO HTML

Part 6



FORMS

- What are forms?
 - An HTML form is an area of the document that allows users to enter information into fields
 - A form may be used to collect personal information, opinions in polls, user preferences and other kinds of information



FORMS

- There are two basic components of a Web form: the shell, the part that the user fills out, and the script which processes the information
- HTML tags are used to create the form shell. Using HTML you can create text boxes, radio buttons, checkboxes, drop-down menus, and more...



EXAMPLE: FORM

First Name:

← Text Box

Last Name:

Type of Shirt:

← Drop-down Menu

Size: Large Medium Small

← Radio Buttons

Color: Red Navy Black

← Checkboxes

Comments?

← Text Area

Reset Button

Submit Button



THE FORM SHELL

- A form shell has three important parts:
 - the `<FORM>...</FORM>` tag, which includes the address of the script which will process the form
 - the form elements, like text boxes, radio buttons and others
 - the submit button which triggers the script to send the entered information to the server



CREATING THE SHELL

- To create a form shell, type

<FORM METHOD=POST ACTION="script_url">

where “**script_url**” is the address of the script

- Create the form elements
- End with a closing **</FORM>** tag



- **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

<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>
```

**Please enter the following
bioData**

First Name:

Last Name:

Nationality:

The Phone Number:

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.

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.

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.

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.

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.

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.

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.

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.

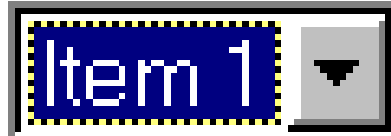
OTHER ELEMENTS USED IN FORMS

- 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.

OTHER ELEMENTS USED IN FORMS



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

```
<SELECT size=1>
```

```
<OPTION selected value=1>Select, please:</OPTION>
```

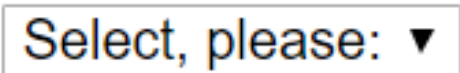
```
<OPTION value=2> 1</OPTION>
```

```
<OPTION value=3> 2 </OPTION>
```

```
<OPTION value=4> 3</OPTION>
```

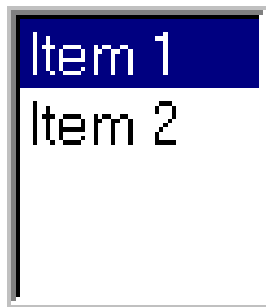
```
<OPTION value=5> 4 </OPTION>
```

```
</SELECT>
```



OTHER ELEMENTS USED IN FORMS

- **List Box:**



A list box containing two items: "Item 1" and "Item 2". "Item 1" is highlighted with a blue background, indicating it is the selected item.

- **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>What type of Computer do you have?<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

INTRODUCTION TO HTML

Part 7



FRAMES

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

Objectives:

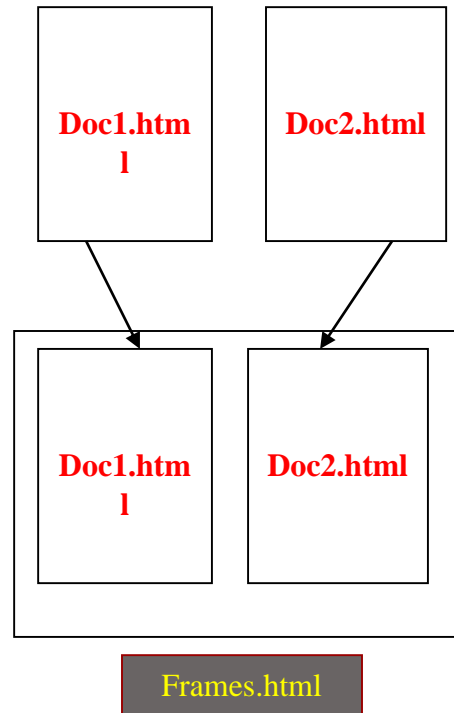
Upon completing this section, you should be able to:

- 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 COLS="23%, 77%"

FRAME

**NAME=
left_pane
SRC=Doc1.
html**

FRAME

**NAME=right_pane
SRC= Doc2.html**



<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

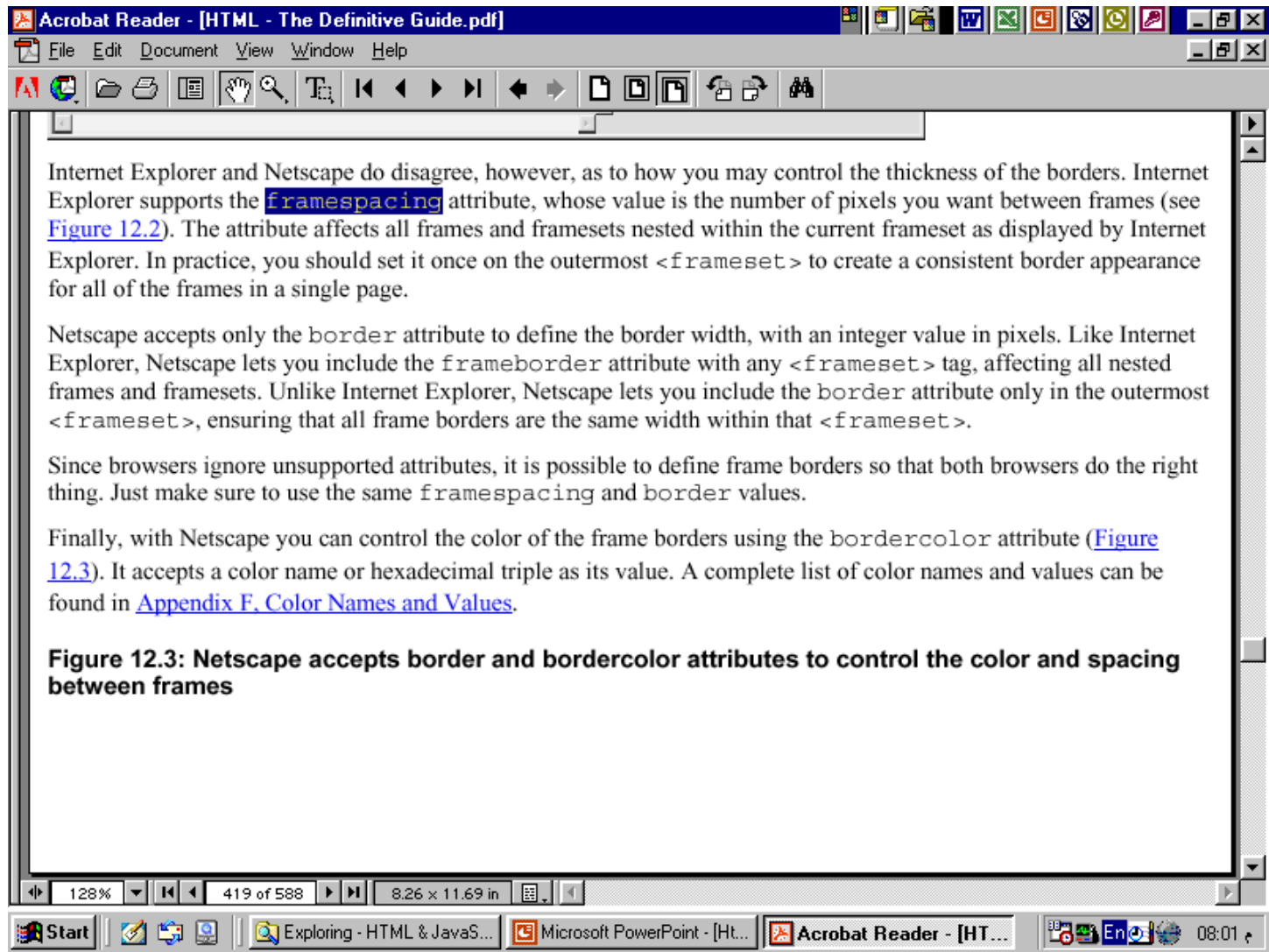
- **COLS**: Determines the size and number of rectangular columns within a <FRAMESET>. They are set from **left** to **right** of the display area.

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.



<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.

Banner File	
Links File	Contents File



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

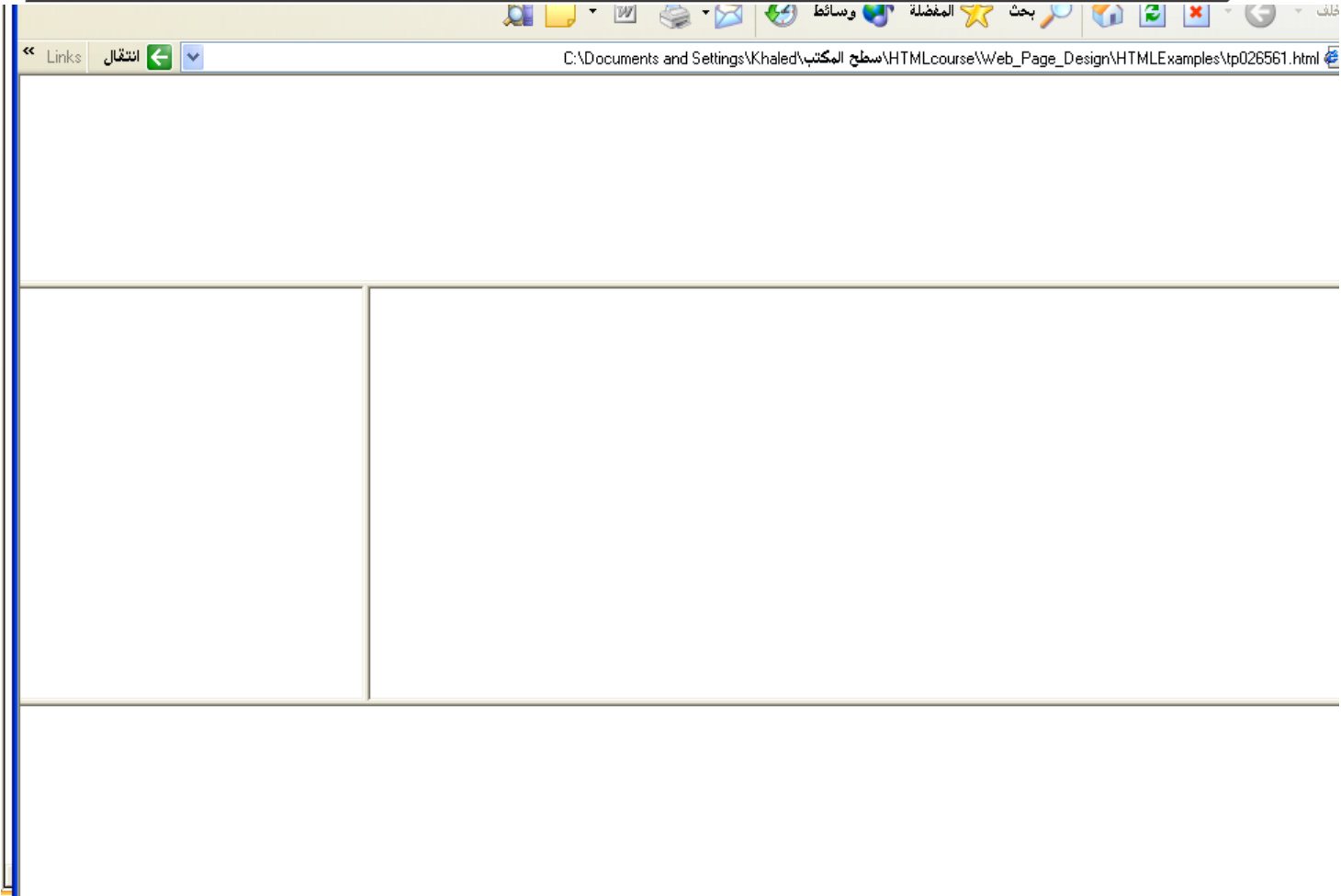




Figure 5-14: Frames created with <FRAMESET ROWS="50%, 50%">

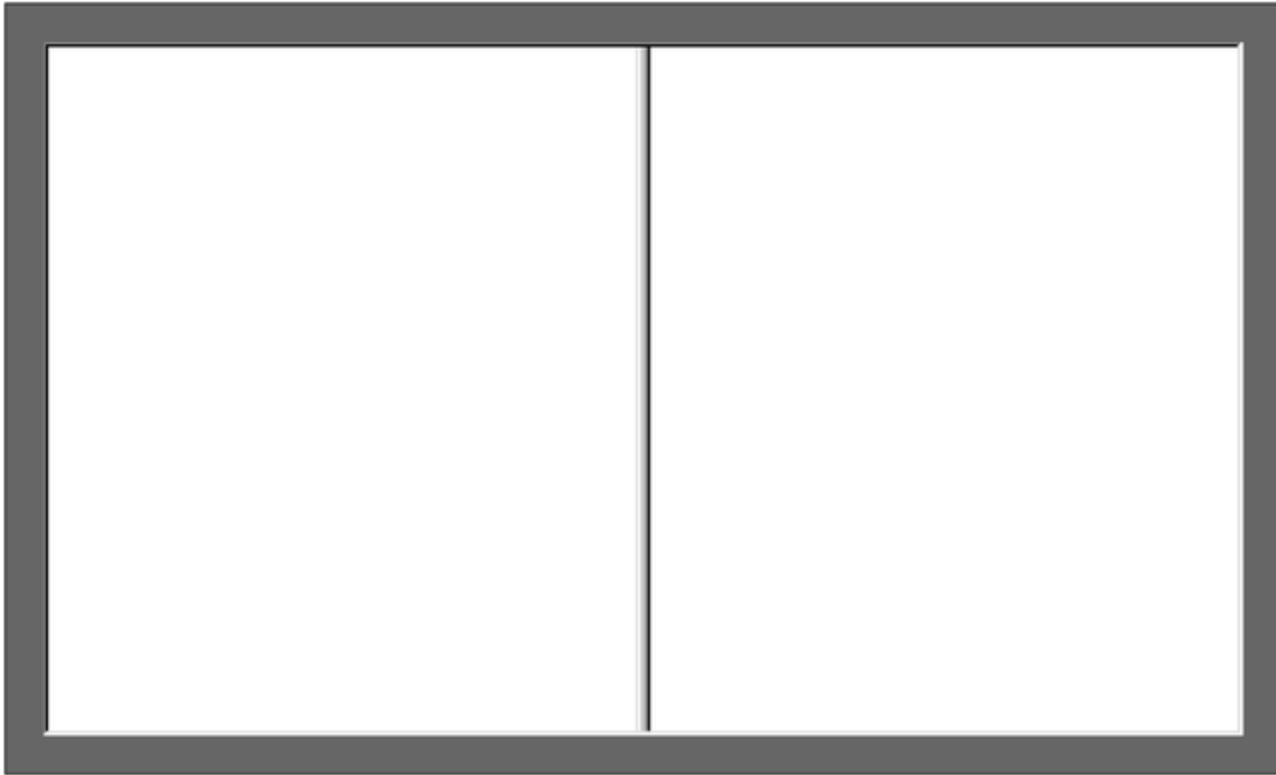


Figure 5-15: Frames created with `<FRAMESET COLS="50%, 50%">`

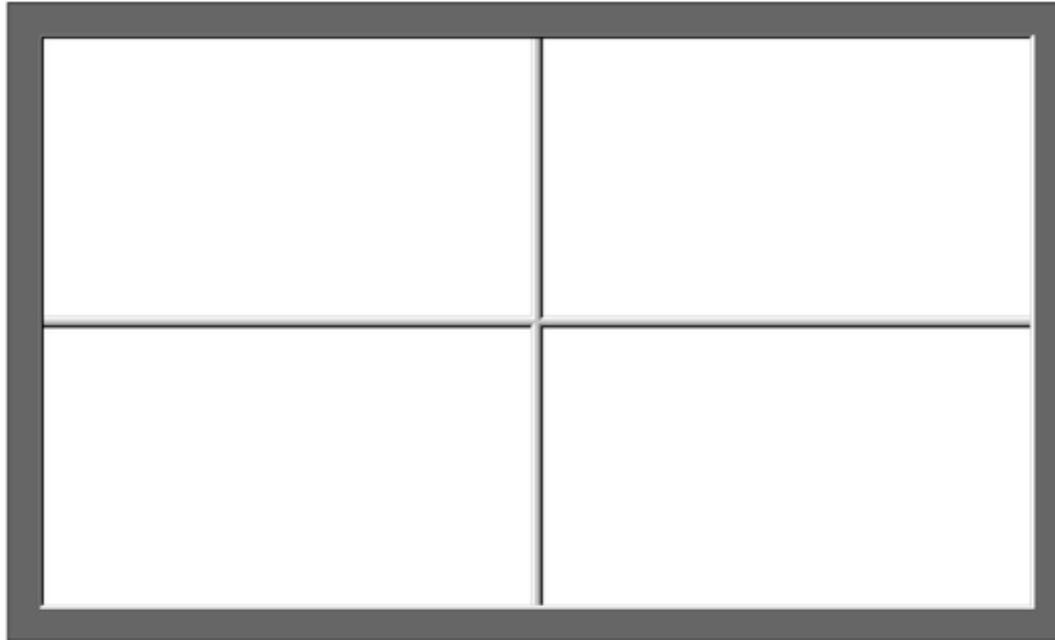


Figure 5-13: Frames created with `<FRAMESET ROWS="50%, 50%" COLS="50%, 50%">`

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>
```

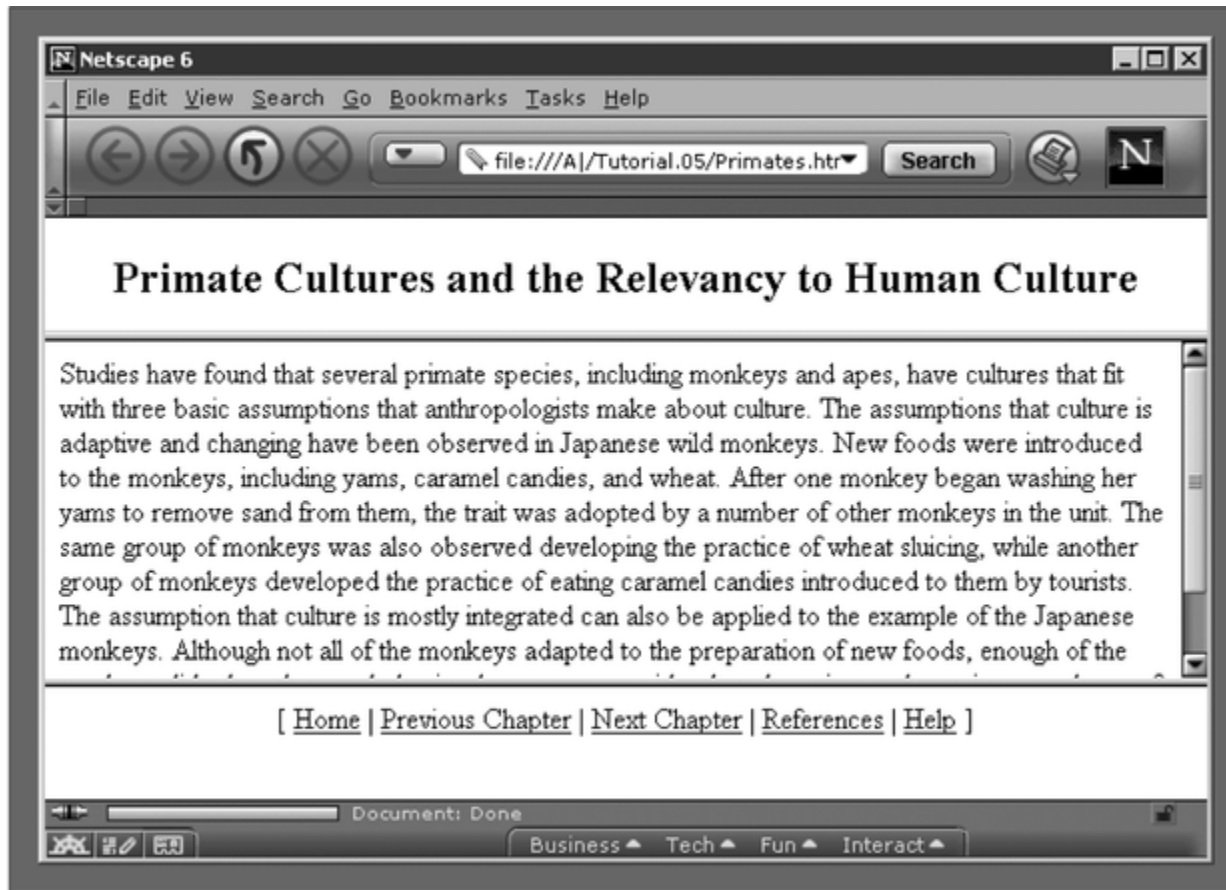


Figure 5-24: Output of program that includes NORESIZE and SCROLLING attributes

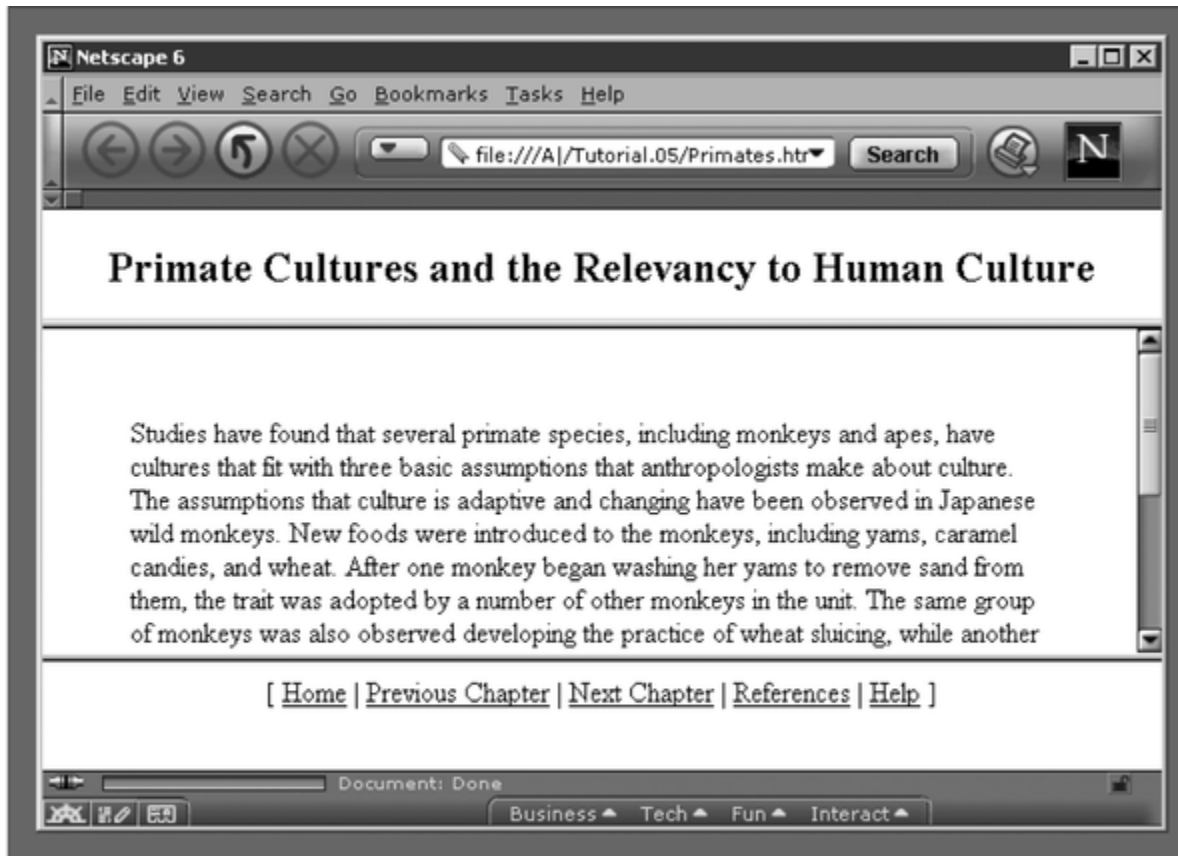


Figure 5-25: Middle frame changed to <FRAME SRC="body.html" MARGINHEIGHT=50 MARGINWIDTH=50>

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. **

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. **

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

**3. **

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

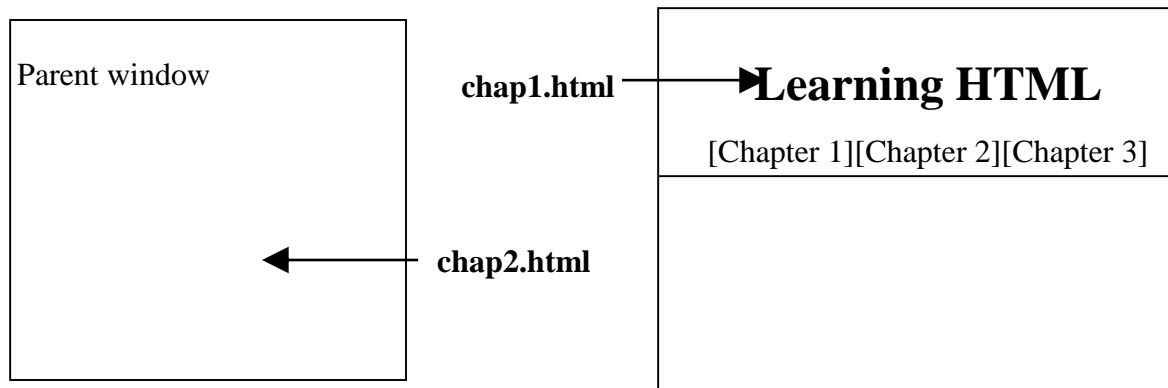
**4. **

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>
```

