



INTRO TO

HTML

HYPertext MARKUP LANGUAGE

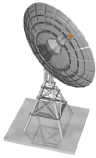


World Class Provider of Web Hosting E-COMMERCE AND CUSTOM INTERNET SOLUTIONS

Quality Web Hosting ✓

is the foundation for a Successful and effective internet presence.

At OLM, we define quality web hosting by :



- *Internet Connection Speed*
- *Connection Reliability*
- *Custom Service*

Web Hosting STARTING @ \$13.95/MONTH

- FTP & FrontPage 2000 Access
- Full Web Control System
- MySQL & mSQL Database Access
- Redundant DS3 High Speed Connections
- 24 X 7 Monitoring & Technical Support
- 15 cgi-scripts Provided

E-COMMERCE STARTING @ \$29.95/MONTH

- On-line Payment Service Available
- Secure Server (SSL) Encryption
- Browser-Based Store Creation
- Merchant Accounts Available On-Line
- Multiple Plans Available

<http://olm.net>

Telephone: 1-877-265-6638

Copyrights and Trademarks

No part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means – electronic, mechanical, recording, or otherwise – without the prior written consent of the publisher.

Netscape Navigator is a trademark of Netscape Communications Corp.
Windows 3.1, Windows 95, Windows NT, and Internet Explorer are trademarks of Microsoft Corporation.

All trademarks and brand names are acknowledged as belonging to their respective owners.

Published by

XtraNet

180 Attwell Dr., Suite 130 Toronto, Ontario, Canada M9W 6A9
Phone: 416-675-1881 Fax: 416-675-9217 E-mail: info@xnu.com

Copyright © 1999 by XtraNet

All Rights Reserved

January 1999

First Edition

1 2 3 4 5 6 7 8

You'll Never Outgrow HoTMetal PRO!

HoTMetal PRO™ 6.0 offers the ability to build sites quickly, the power to do more and the tools to learn how.

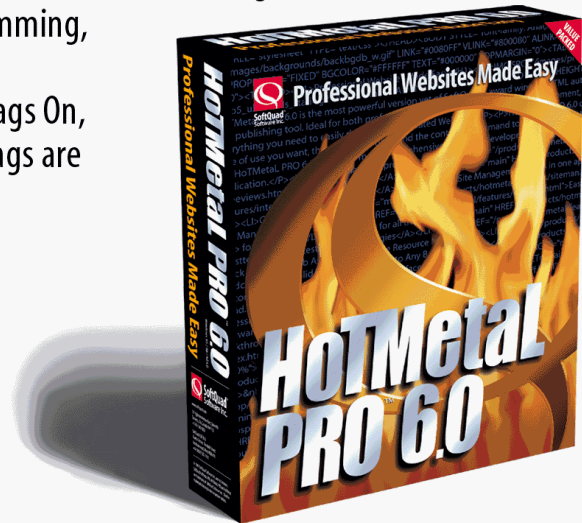
- more support for new users than any other professional Website development tool
- create your site quickly with HoTMetal Site Maker and WYSIWYG editing tools
- add dazzling effects to your pages, without programming, with the Asset Manager
- switch between three editing views – WYSIWYG, Tags On, HTML Source – to quickly see and learn how your tags are turned into the pages you see in your browser
- check your code with HoTMetal PRO's validator to pinpoint any errors

Buy your copy today.

Web: www.hotmetalpro.com

Email: sales@softquad.com

Tel: 416-544-9000 Toll Free: 1-800-387-2777



Share these FREE Courses!

Why stuff your friend's mailbox with a copy of this when we can do it for you!

Just e-mail them the link info – <http://www.trainingtools.com>

Make sure that you visit the site as well:

- MORE FREE COURSES
- Weekly Tool Tips
- Updated course versions
- New courses added regularly

So don't copy files or photocopy - Share!

End User License Agreement

Use of this package is governed by the following terms:

A. License

TrainingTools.com Inc, ("we", "us" or "our"), provides the Licensee ("you" or "your") with a set of digital files in electronic format (together called "the Package") and grants to you a license to use the Package in accordance with the terms of this Agreement. Use of the package includes the right to print a single copy for personal use.

B. Intellectual Property

Ownership of the copyright, trademark and all other rights, title and interest in the Package, as well as any copies, derivative works (if any are permitted) or merged portions made from the Package shall at all times remain with us or licensors to us. This Package is protected by local and international intellectual property laws, which apply but are not limited to our copyright and trademark rights, and by international treaty provisions.

C. Single-User License Restrictions

1. You may not make copies of the files provided in the Package
2. You may not translate and/or reproduce the files in digital or print format
3. You may not rent, lease, assign or transfer the Package or any portion thereof
4. You may not modify the courseware

Table of Contents

Chapter 1 - HTML Introduction	1
Linear Media	2
Hypermedia	2
What is HTML Markup.....	3
Document Structure.....	4
A Basic Document	4
Review Questions.....	6
Summary	7
Chapter 2 - Overview of HTML Page Creation	8
Choosing a Text Editor,	9
Starting NotePad,	9
Creating a Basic Starting Document,	10
Setting Document Properties,.....	11
Color Codes,	11
The BODY Element,	12
BODY Text Color,	12
LINK, VLINK and ALINK,	13
Body Image Backgrounds,.....	13
HTML Tip: Image Backgrounds.....	14
Previewing Your Work,	15
Edit, Save & View Cycle	15
Using Netscape Navigator	15
Using Microsoft Internet Explorer.....	16
Exercise	17
Review Questions.....	18
Summary	19
Chapter 3 - Headings, Paragraphs and Breaks	20
Headings, <Hx> </Hx>	21
Paragraph, <P> </P>.....	22
HTML Tip: Multiple Spaces	22
Break, </BR>.....	23
Horizontal Rule, <HR>.....	24
Exercise – Headings.....	25
Exercise – Horizontal Rules	26
Exercise – Paragraphs & Breaks.....	27
Review Questions.....	28
Summary	29
Chapter 4 - Character Formatting	30
Bold & Italic and other Character Formatting	31
Alignment.....	32
Special Characters & Symbols	32
Additional Character Formatting Elements.....	33
Strike Through Text	33
Big Text	33
Small Text	33
Subscript Text.....	33
Superscript Text	33
Defining instance of enclosed term	33
Formatting extracts of program code	33
Formatting sample output from programs	33
Formatting Keyboard text.....	33
Formatting variables	33
Basefont	34
HTML Tip: BlockQuotes	34
Exercise 1	35
Exercise 2.....	36
Optional Exercise.....	36
Review Questions.....	37

Table of Contents

Summary	38
Chapter 5 - Lists	39
List Elements	40
Unordered List	40
Ordered List	41
Definition List	42
Nesting Lists	42
Exercise – Ordered List	43
Optional Exercises	44
Review Questions	45
Summary	46
Chapter 6 - Images	47
Supported Image Formats	48
GIF	48
GIF Interlacing	48
GIF Transparency	48
GIF Animated	48
JPEG	49
Limited Support or Non-Supported Image Formats	49
Inserting Images 	50
Additional Image Attributes	50
Horizontal Space HSPACE	50
Vertical Space VSPACE	50
HTML Tip: Images	50
Exercise 1	51
Optional Exercise	51
Review Questions	52
Summary	53
Chapter 7 - Anchors, URLs and Image Maps	54
Link Elements 	55
Link Types	55
Internal Links	55
Local Links	55
External Links	55
URLs – Uniform Resourced Locators	56
HTTP – HyperText Transport Protocol	56
FTP – File Transfer Protocol	56
News – News Groups	56
Gopher	57
E-mail	57
Telnet – Remote Terminal Emulation	57
Internal Links	58
HTML Tip: Links	58
Image Maps	59
Server-side Image Maps	59
Client-side Image Maps	60
Exercise – Mailto and HTTP Link	61
Review Questions	62
Summary	63
Chapter 8 - Tables	64
Tables <TABLE> </TABLE>	65
Table Attributes	66
Table Captions	67
Table Header	67
HTML Tip: Centering Tables	69
Exercise – Add and customize a table	70
Review Questions	71
Summary	72

Table of Contents

Chapter 9 - Frames	73
Frames.....	74
Frames Architecture	75
Creating a Frames Page	76
Frameset.....	76
Other Attributes of the FRAMESET Element.....	77
Frame.....	78
Noframes.....	79
Targets.....	81
Special Targets.....	82
Exercise – Using a Framed page	83
Exercise – Creating a link in a framed environment.....	83
Review Questions.....	84
Summary	85
Chapter 10 - Forms	86
Forms <FORM> </FORM>.....	87
Form Attributes	88
HTML Tip: Forms.....	88
Form Elements	89
Textboxes.....	89
Password boxes	90
Check boxes.....	91
Radio/Option Button	91
File Upload	92
Push button	92
Submit button	93
Image submit button	93
Reset Button.....	93
Text area	94
Select	95
Drop down list.....	95
List box.....	95
Options.....	95
Common Gateway Interface - CGI	96
CGI Scripts	96
Section 3.2.a: Simple CGI Program	97
Section 3.2.b: Three-Tier Web Application Development	98
Section 3.2.c: Four-Tier Web Application Development.....	99
Exercise – Creating an Order Form.....	100
Review Questions.....	113
Summary	114
Chapter 11 - JAVA and JAVAScript Introduction	115
JAVA.....	116
Applet.....	116
Applet Attributes	116
<PARAM> Element.....	116
JAVAScript.....	118
Exercise – Inserting a JAVA Applet.....	120
Optional Exercise.....	121
Review Questions.....	122
Summary	123
Chapter 13 - ActiveX Objects and VBScript introduction	124
ActiveX Support.....	125
<OBJECT>	125
VBScript.....	128
Exercise – Adding an ActiveX Object to a page.....	129
Optional Exercise – Adding VBScript to a page.....	130
Review Questions.....	131

Table of Contents

Summary	132
Chapter 14 - Special Elements	133
Uses of the Meta Element	134
Client Pull, Server Push	134
Description	134
Keywords.....	134
Author.....	134
Company	135
Copyright	135
Adding sound to an HTML page.....	136
Embed.....	136
BGSound.....	136
Marquee.....	137
Blink	137
Exercise – Adding Client Pull, Server Push	138
Optional Exercise.....	139
Review Questions.....	140
Summary	141
Chapter 15 - Page Layout and Design Considerations	142
Technical Design Considerations.....	143
Screen Resolution	143
Color Depth	143
Document Size vs. Download Time	144
Page Loading – HTTP 1.0 vs. HTTP 1.1.....	145
Browser Compatibility & Quirks.....	145
Page layout.....	146
Page Layout Guidelines.....	146
Site Design Factors and Criteria.....	148
Site Layout and Navigation.....	148
Review Questions.....	149
Summary	150
Chapter 16 - Cascading Style Sheets	151
Introduction to Cascading Style Sheets	152
Inline Styles	153
Embedded Style Sheets	154
Linked Style sheets	156
Classes	157
IDs.....	158
DIV and SPAN	159
Cascading and Inheritance.....	160
Exercise – Inline Styles	162
Exercise – Embedded Styles.....	163
Exercise – Linking in a Cascading Style Sheet	164
Exercise – Editing a Cascading Style Sheet	165
Review Questions.....	166
Summary	167

HTML Introduction

In this chapter you will be introduced to the concepts of linear media and hypermedia. You will learn about HTML and the basics of document structure.

Objectives

Upon completing this section, you should be able to

1. Explain hypermedia vs. linear media
2. Define HTML
3. Describe the Basic Document Structure
4. Identify the sub-elements of the header

Linear Media

Linear media is a term used to describe any media where there is a defined beginning and a linear progression to the end. Forms of linear media such as movies, audio and videotapes, and most books are organized with this expectation. The World Wide Web, however, is organized very differently.

Hypermedia

Hypermedia is where the user simply selects the next item of interest and is immediately transported to that new location. A good example is an audio CD where you can choose song 5 and listen to it almost immediately. Contrast this with an audiocassette where you would have to scan through from your current location on the tape to the beginning of the song.

When this concept is applied to text you get **hypertext**, where by *{Clicking}* on a link or hotspot (hyperlink) you are immediately transported to a new location within the same page or to a new page altogether.

When you interlink a large number of pages of text on different computers all over the world, you get a spider web-like system of links and pages. This is known as the World Wide Web – a system whereby pages stored on many different web servers, connected to the Internet, are linked together.

The system is useful because all of the pages are created in the same format. This format or “language” is called HTML, (Hypertext Markup Language) a subset of an international standard for electronic document exchanged called SGML (Standard Generalized Markup Language).

In this class you will be introduced to the format of an HTML page, you will learn about the components that make up HTML, and how to create pages that can be published on the World Wide Web.

What is HTML Markup?

HTML is a set of logical codes (markup) in parentheses that constitute the appearance of a web document and the information it contains.

E.g. ****This text would appear bold in the browser****

The codes are enclosed by less than “<”, and greater than “>” brackets. These bracketed codes of the markup are commonly referred to as tags. HTML codes are always enclosed between brackets and are not case-sensitive; meaning, it does not matter whether you type them in upper case or lower case. However, tags are easier to recognize in a web document if they are capitalized.

Most elements have an opening element (tag) and a closing element (tag) distinguished by the “/” inside the “<” opening bracket.

The first word or character that appears inside the “<” opening bracket is called the **element**. An element is a command that tells the browser to do something, such as ****.

Words that follow the element and are contained inside the “>” bracket of the opening tag are called **attributes**. Attributes are not repeated or contained in the closing element (tag). Confused? Attributes are another way of describing the element’s properties. For instance, a t-shirt can have several properties or “attributes” – the t-shirt’s color, size, material etc. are all attributes.

Attributes that appear to the right of the element are separated by a space, and followed by an equal sign. The **value** of the attribute is contained in quotes. In the following example the element is FONT, the attribute is COLOR, and the value of that attribute is BLUE.

E.g. ****This text would be blue****

Most HTML elements have more than one attribute.

E.g. ****This text would be blue and one size larger than normal****

Document Structure

The way a document is marked up with elements and their attributes is according to a Document Type Definition (DTD). These are the rules that govern the way in which a document can be marked up.

The authoritative source for information about HTML and the HTML DTD is the World Wide Web Consortium (W3C) at <http://www.w3.org>. The World Wide Web Consortium is a not-for-profit organization that coordinates the evolution of the Web. It includes the Internet Engineering Task Force, the group of people who make recommendations for new markup.

A Basic Document

An element called **HTML** surrounds the whole document. This element contains two sub-elements, **HEAD** and **BODY**. These elements are required to form any HTML document.

```
<html>
<head>
<title>Document Title</title></head>
<body></body>
</html>|
```

<HTML>

<HEAD> has sub-elements that define header material:

<TITLE> document title. The title of your document is what appears in a web browser's Favorite or Bookmark list. Your document's title should be as descriptive as possible. Search engines on the Internet use the document's title for indexing purposes. **</TITLE>**

<BASE> can be used to record the document's location in the form of a URL. The URL recorded here may be used to resolve a relative URL (necessary if the document is not accessed in its original location). **</BASE>**

<ISINDEX> indicates to the browser that the document is an index document. This is used only if the document is on a server that does indexing. **</ISINDEX>**

<LINK> indicates a relationship between this document and some other object on the Web. **</LINK>**

<META> provides information such as the page's keywords and description that appears in HTTP headers. **</META>**

<SCRIPT> contains either JAVA Script or VB Script **</SCRIPT>**

<STYLE> contains information used by cascading style sheets
</STYLE>

</HEAD>

<BODY> the remaining HTML elements are contained within these tags.
</BODY>

</HTML>

Note: a framed document is formatted differently than a basic document and is discussed in the advanced section of this course.

Review Questions

1. What does HTML stand for?
2. What is an Element?
3. What are the Attributes of an Element?
4. What are the three basic elements of an HTML document?
5. What are the elements that can be contained in the header of the document?

Summary

As a result of this chapter, you should be able to

- Explain hypermedia vs. linear media
- Define HTML
- Describe the Basic Document Structure
- Identify the sub-elements of the header

2

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.

Choosing a Text Editor

There are many different programs that you can use to create web documents. Text editors are basic word processing programs without all of the “bells and whistles” of full-blown word processors. The advantage of using a text editor is that the files are created and saved with few if any invisible formatting codes, which could drastically effect your document when, saved as a web page and displayed in a browser. For this reason, text editors can be used quite effectively 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. 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.

The current versions of both Microsoft Word and Corel WordPerfect also have the abilities to create web pages.

For this course, we will focus on using the standard Microsoft Windows text editor, NotePad. You can apply the same concepts using any text editor on any platform.

Starting NotePad

NotePad is the standard text editor that comes with both 16 and 32-bit versions of the Microsoft Windows operating system. To start NotePad in Windows 95 follow the steps below:

1. Click on the “Start” button located on your Windows task bar.
2. Click on “Programs” and then click on the directory menu labeled “Accessories”.
3. Locate the shortcut called “NotePad” and click the shortcut once.

Creating a Basic Starting Document

As stated in Chapter One, there are certain elements that are required in a web document's structure. The easiest way to create and develop a document is to type the required elements in as a starting point. This way you can add to your document by inserting elements and adding content between the starting and ending tags of existing elements and you won't have to try to remember if you have typed in the closing tag or not.

In NotePad you would start with:

```
<HTML>

<HEAD>
<TITLE></TITLE>
</HEAD>

<BODY>

</BODY>
</HTML>
```

At this point your page has a HEAD and a BODY section inside the HTML tags. You also have a TITLE element, inside the HEAD element, which you should fill in. The text in the TITLE element is used by the surfer's browser and also by search engines.

The TITLE of your document appears in the very top line of the user's browser. If the user chooses to "Bookmark" your page or save as a "Favorite"; it is the TITLE that is added to the list.

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

The following is an example of a document title:

```
<HTML>

<HEAD>
<TITLE>XtraNet University: HTML, JavaScript, Netscape,
Microsoft and SoftQuad Training</TITLE>
</HEAD>

<BODY>

</BODY>
</HTML>
```

Setting Document Properties

Document properties are controlled by attributes of the BODY element. For example, there are color settings for the background color of the page, the document's text and different states of links.

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, an RGB value with 00 as the first two digits has no red in the color. See the chart below for a listing of some of the commonly used colors:

RGB Color	Hexadecimal Value
White	#FFFFFF
Black	#000000
Red	#FF0000
Green	#00FF00
Blue	#0000FF
Magenta	#FF00FF
Cyan	#00FFFF
Yellow	#FFFF00
Aquamarine	#70DB93
Baker's Chocolate	#5C3317
Violet	#9F5F9F
Brass	#B5A642
Copper	#B87333
Pink	#FF6EC7
Orange	#FF7F00

There are several resources available on the Internet that chart colors and their hexadecimal values. If you require more information about color values, there is an excellent site entitled "VGDesign's Interactive Color Cube" that displays the background color code when you put your cursor over a small color sample. The web address is:
<http://www.vgdesign.com/color.html>

The BODY Element

The BODY Element of a web page is an important element in regards to the page's appearance. This element contains information about the page's background color, the background image, as well as the text and link colors.

If the BODY Element is left blank, web browsers will revert to their default colors. In older browsers, if the BODY element is left blank, the page's background color will be a light gray. The newer browsers, IE 4+ and Netscape 4+, default to using the client's Windows colors settings.

It is very common to see web pages with their background color set to white or some other color. To set your document's background color, you need to edit the <BODY> element by adding the BGCOLOR attribute. The following example would 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. This will affect all of the text within the document that is not being colored by some other element, such as a link. 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. As suggested earlier, it is important to ensure that your document's text is a color that will stand out from your background color.

LINK, VLINK, and ALINK

These attributes control the colors of the different link states:

- LINK - initial appearance – default = Blue
- VLINK - visited link – default = Purple
- ALINK - active link being clicked – default = Red

Many web developers will set the link colors of their documents to flow with the color scheme of the site. The format for setting these attributes is:

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

The results of the above BODY element would be a white background with links being blue, visited links as magenta and active links colored in yellow.

Using Image Backgrounds

The BODY element also gives you the ability of setting an image as the document's background. Background images are "tiled" in the web browser; which means that they are replicated and positioned below and beside each other until the browser screen is filled. To create a professional look, images must be 'seamless'. Meaning that when the copies are placed below and beside each other the seams are invisible.

Using background images can be very effective if used properly. For instance, you may want your company's logo as your background or you could also create a border background so that it appears as though your page is divided into two columns. A background image must be either in the form of a .gif or .jpg file. There will be more information on image files provided in Chapter 6.

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

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

In this example, we have set the document's background image to 'logo.gif'. We have also added the BGCOLOR attribute as well so that the browser window will have a white background during the process of loading the background image.

HTML Tip

When using a background image for your document, keep in mind that your text will appear on top of the image. Background images should be a light color so that your text is readable.

Before you set the color properties of your document, keep in mind that not every user who visits your page will have their display colors set to the same as yours. Some users still work on computers that can only display 16 colors. Use caution when setting your document's properties if you are concerned about how the page will appear.

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. Since you are publishing for the Internet, you must save your files in a format that web browsers can interpret. A standard web page's file extension is usually ".htm". Some developers name their files with the extension ".html" but this format is not compliant on all platforms. In our examples we will use the .htm file extension.

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 .html). i.e. index.html

In NotePad you have to type the entire filename including its extension. If you do not type in the file's extension, NotePad will assume that you want your document to be a text document with the extension .txt.

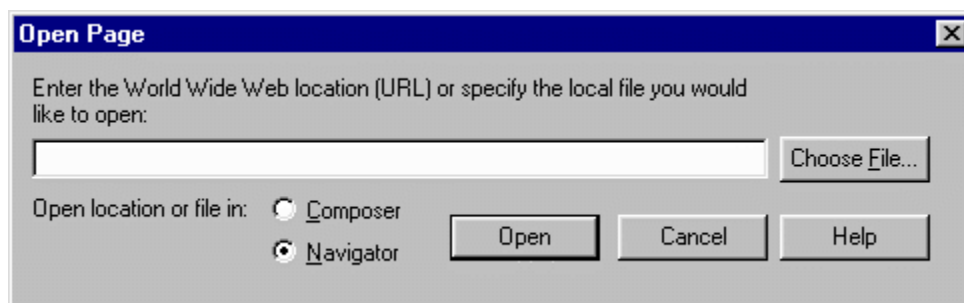
Edit, Save & View Cycle

As you continue to add to your web document(s), it is important to preview your work in a browser such as Netscape Navigator or Microsoft Internet Explorer.

To preview your work, open a web browser and do the following:

In Netscape Navigator:

1. Click on the menu labeled "File ..."
2. Locate the menu option, "Open Page ..."



3. In the “Open Page” dialog box, click on the “Choose File ...” button and locate your web document. Ensure that the “Navigator” button option is clicked as you want to preview the work in the browser window.
4. Once you have chosen the file click on “Open”

In Microsoft Internet Explorer:

1. Click on the menu labeled “File ...”
2. Locate the menu option, “Open ...”



3. In the “Open” dialog box, click on the “Browse ...” button and locate your web document. Click “OK” once you have selected your file.

Once you have opened and previewed your work in a web browser, you can continue working by adding to and editing the html file, saving the file with the changes and then viewing the changed file.

You will be more productive if you do not close your web browsers; simply minimize them and continue working. When you want to preview your work again:

- save your html file’s changes
- switch to one of your browsers
- hold down the [SHIFT] key
- in Netscape, click on the button labeled “RELOAD”
- in IE click on the button labeled “REFRESH”

The web browser will load the same document but with the new revisions. This process is the Edit, Save and View cycle.

Chapter 2 – Exercise 1

The goal of this exercise is to familiarize you with using the concepts discussed in Chapter 2.

Procedure

1. Start Notepad.
2. Create the following HTML page:

```
<HTML>
<HEAD>
<TITLE>XYZ Corporation</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
</BODY>
</HTML>
```

This will set your documents background color to white. (Older browsers default to gray)

3. Save your file – call it index.htm. Save it in the C:\HTMLFILES\Exercises directory.
4. Preview your file. View your file in Navigator, IE and any other browser that you have installed and are using.

Review Questions

1. What are the advantages of using a text editor instead of a full-blown word processor?
2. What does the TITLE element do?
3. What attribute controls the document's background color?
4. How are colors expressed as attribute values?
5. What does the BACKGROUND attribute do?

Summary

As a result of this chapter, you should be able to

- Choose a text editor
- Create a basic document
- Set the document's properties
- View the results of your work

3

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, Paragraphs, Breaks and Horizontal Rules

Headings, <Hx> </Hx>

Inside the BODY element, heading elements H1 through H6 are generally used for major divisions of the document. Headings are not mandatory. Headings are permitted to appear in any order, but you will obtain the best results when your documents are displayed in a browser if you follow these guidelines:

- H1: should be used as the highest level of heading, H2 as the next highest, and so forth.
- You should not skip heading levels: e.g., an H3 should not appear after an H1, unless there is an H2 between them.

The size of the text surrounded by a heading element varies from very large in an <H1> tag to very small in an <H6> tag.

```
<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>
```

As displayed by the browser.

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Paragraph, <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 suit the window size of the browser in which it is being displayed. Each line of text will stretch the entire length of the window.

```
<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>
```

As displayed by the browser.

Heading 1

Paragraph 1, ...

Heading 2

Paragraph 2, ...

Heading 3

Paragraph 3, ...

Heading 4

Paragraph 4, ...

Heading 5

Paragraph 5, ...

Heading 6

HTML Tip

If you insert multiple empty paragraphs in succession, you will notice the browsers display these with only one blank line between them.

If you try to type two words separated by five spaces, only one space will display in the browser

Browsers ignore multiple spaces between words, within paragraphs and within tables. If you have areas where you would like extra spaces or you want your empty paragraphs to give you extra line spacing, you will want to insert a NBSP (non-breaking space). A NBSP is an invisible character that takes up one space.

To insert a NBSP type

Break,

Line breaks allow you to decide where the text will break on a line or continue to the end of the window. There may be instances where you want the text to appear on the next line. Example: a return address - contained in one Paragraph but on multiple lines or text in a Header, such as a title for the document.

You can insert a Forced Line Break


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

As displayed by the browser.

Heading 1

Paragraph 1,
Line 2
Line 3
...

Heading 2

Paragraph 2, ...

A
 is an Empty Element, meaning that it may contain attributes but it does not contain content. The
 element does not have a closing tag. The closing tag, </BR>, is not required as this element marks a position and does not contain content.

The
 element uses one attribute – CLEAR which can have a value of LEFT, RIGHT or ALL. The CLEAR attribute forces a line down; if an image is located at the page's left margin, the clear LEFT attribute will move the cursor down to the first line that is cleared at the left margin. The same applies with the option of the clear RIGHT attribute. If an image is at either the left or right margins, the clear ALL attribute will force a line break until both margins are cleared.

Horizontal Rule, <HR>

The <HR> element causes the browser to display a horizontal line (rule) in your document. This element does not use a closing tag, </HR>. Simply type <HR> to insert a horizontal line (rule) element into your document.

Key attributes for use with the <HR> element:

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

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

As displayed by the browser.

Heading 1

Paragraph 1,
Line 2

Line 3

...

Heading 2

Paragraph 2, ...

Chapter 3 – Exercise 1 - Headings

In this exercise you will add headings to your document.

Procedure

1. Add an H1 to your page. Inside the tags key in **{Your Company Name}**.
2. Add an H2 after the H1 on your page. Inside the tags key in **{Your Division or Dept. Name}**.
3. Add an H3 after the H2 on your page. Inside the tags key in **{A Project Name}** e.g. Intranet Project.
4. Add an H4 after the H3 on your page. Inside the tags key in **{Project Name}** Plan.
5. Add an H3 after the H4 on your page. Inside the tags key in **{A different Project Name}** e.g. Internet Project.
6. Add an H4 after the H3 on your page. Inside the tags key in **{Project B Name}** Plan.
7. Your document should look something like this at this point:

```
<HTML>
<HEAD>
<TITLE>XYZ Corporation</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1>XYZ Corporation</H1>
<H2>IS Department</H2>
<H3>Intranet Project</H3>
<H4>Intranet Project Plan</H4>
<H3>Internet Project</H3>
<H4>Internet Project Plan</H4>
</BODY>
</HTML>
```

As displayed by the browser.

XYZ Corporation

IS Department

Intranet Project

Intranet Project Plan

Internet Project

Internet Project Plan

Chapter 3 – Exercise 2 – Horizontal Rules

In this exercise you will add HR's to your document.

Procedure

1. Add a Horizontal Rule after the H1 and before the H2.
2. Make the Horizontal Rule only 50% of the width of the screen.
(Remember elements have attributes.)
3. Add a Horizontal Rule after the last H4 on the page.
4. Your document should look similar to this now:

```
<HTML>
<HEAD>
<TITLE>XYZ Corporation</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1>XYZ Corporation</H1>
<HR WIDTH="50%">
<H2>IS Department</H2>
<H3>Intranet Project</H3>
<H4>Intranet Project Plan</H4>
<H3>Internet Project</H3>
<H4>Internet Project Plan</H4>
<HR>
</BODY>
</HTML>
```

As displayed by the browser.

XYZ Corporation

IS Department

Intranet Project

Intranet Project Plan

Internet Project

Internet Project Plan

Chapter 3 – Exercise 3 – Paragraphs & Breaks

In this exercise you will add paragraphs and a break to your document.

Procedure

1. Add a Paragraph between each one of the Headings. We will add content between the paragraph tags later.
2. Add a BR in the first Paragraph so your fill looks similar to this one:

```
<HTML>
<HEAD>
<TITLE>XYZ Corporation</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H1>XYZ Corporation</H1>
<HR WIDTH="50%">
<H2>IS Department</H2>
<P><BR></P>
<H3>Intranet Project</H3>
<P></P>
<H4>Intranet Project Plan</H4>
<P></P>
<H3>Internet Project</H3>
<P></P>
<H4>Internet Project Plan</H4>
<P></P>
<HR>
</BODY>
</HTML>
```

As displayed by the browser.

XYZ Corporation

IS Department

Intranet Project

Intranet Project Plan

Internet Project

Internet Project Plan

Review Questions

1. How do browsers handle multiple spaces?
2. How do paragraphs format text?
3. How many Headings are there?
4. When would you use a Break?
5. How would you insert a HR that is centered and half a page wide?

Summary

As a result of this chapter, you should be able to

- List and describe the different Heading elements.
- Use paragraphs to add text to a document.
- Insert breaks where necessary.
- Add a Horizontal Rule.

4

Character 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

- ****Two sizes bigger**** Text size can be modified with the font element and the size attribute. 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 is size 3.
- ** Bold **
- **<I> Italic </I>**
- **<U> Underline </U>** (Not recommended, as links are underlined.)
- Color = #RRGGBB" The COLOR attribute of the FONT element. E.g. ****this text has color****
- **<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.
- ** Emphasis ** browsers usually display this as italics.
- ** STRONG ** browsers display this as bold.
- **<TT> TELETYPE </TT>** Text is displayed in a mono-spaced font.
- **<CITE>** Citation **</CITE>** represents a document citation.

Sample Coding:

```
<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>  
<CITE>Citation</CITE></P>
```

Netscape Navigator Sample

One Size Larger - Normal - One Size Smaller
Bold - *Italics* - Underlined - Colored
Emphasized - **Strong** - Tele Type
Citation

Alignment

Some elements have attributes for alignment (ALIGN) e.g. Headings, Paragraphs and Horizontal Rules. The three alignment values are: LEFT, RIGHT, CENTER.

Alignment of many other elements is not well supported, meaning that you can't rely on getting the desired results. Proper positioning of all of the components in a web page can be achieved by using other elements to control alignment:

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

Special Characters& Symbols

Special Characters and Symbols not found on the average keyboard can be inserted using special character entities. These special characters are specified in an internationally accepted character set known as the ISO-Latin-1 (ISO-8859-1).

These characters are recognized in HTML as they begin with an ampersand and end with a semi-colon e.g. **&value;** The value will either be an entity name or a standard ASCII character number.

The following 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

Special Character	Entity Name	Special Character	Entity Name
ampersand	&	greater-than sign	>
asterisk	∗	less-than sign	<
cent sign	¢	non-breaking space	
copyright	©	quotation mark	"
fraction one qtr	¼	registration mark	®
fraction one half	½	trademark sign	™

Additional Character Formatting Elements

<STRIKE> strike-through text **</STRIKE>**

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

<SMALL> places text in a small font **<SMALL>**

_{ places text in subscript position **}**

^{ places text in superscript style position **}**

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>
```

Results: (viewed in Navigator)

~~strike through text~~
places text in a big font
places text in a small font
places text in subscript position Normal places text in superscript style position

These special formatting elements are used for technical documentation applications:

<DFN> defining instance of the enclosed term **</DFN>**

<CODE> used for extracts of program code **</CODE>**

<SAMP> used for sample output from programs, scripts, etc. **</SAMP>**

<KBD> used for text to be typed by the user **</KBD>**

<VAR> used for variables or arguments to commands **</VAR>**

Example:

```
<P><DFN> defining instance of the enclosed term </DFN><BR>
<CODE> used for extracts of program code </CODE><BR>
<SAMP> used for sample output from programs, scripts, etc. </SAMP><BR>
<KBD> used for text to be typed by the user </KBD><BR>
<VAR> used for variables or arguments to commands </VAR><BR></P>
```

Results: (viewed in Navigator)

defining instance of the enclosed term
used for extracts of program code
used for sample output from programs, scripts, etc.
used for text to be typed by the user
used for variables or arguments to commands

<BASEFONT>

Used to set the base font size. The size attribute is an integer value ranging from 1 to 7. The base font applies to the normal and preformatted text but not to headings, except where these are modified using the FONT element with a relative font size.

</BASEFONT>

<BLOCKQUOTE>

This element is used to enclose block quotations from other works or to indent sections of a document. (It indents from both sides in a browser)

This element can also contain most other formatting elements such as Headings, paragraphs, tables, etc.

</BLOCKQUOTE>

<ADDRESS>

The address element specifies information such as authorship and contact details for the current document. There is no support for an alignment attribute with the <ADDRESS> element. You will need to use some other method if you want the address displayed differently from left aligned.

</ADDRESS>

```
<P><BLOCKQUOTE>This is the first day of the rest of your life, make it count.</BLOCKQUOTE>
```

```
<CENTER><ADDRESS>180 Attwell Dr. Suite 130<BR>Toronto,  
ON<BR>M9W 6A9</ADDRESS></CENTER></P>
```

```
<H6>Heading 6</H6>
```

```
<P>Paragraph 6, ... </P>
```

Results:

Heading 5

This is the first day of the rest of your life, make it count.

*180 Attwell Dr. Suite 130
Toronto, ON
M9W 6A9*

Heading 6

Paragraph 6, ...

HTML Tip

BlockQuotes can be “nested” – inserted multiple times in succession – to indent a section significantly. This technique also helps you to create more white space on the sides of your pages.

Chapter 4 – Exercise 1

In this exercise you will add special characters and formatting elements.

Procedure

1. Below the bottom Horizontal Rule, add © XYZ Corporation, and center it on the page.
2. At the end of the line you just added, insert a trade mark symbol using `™`.
3. After the information you just added, add an address element and enter in your company's address.
4. Your document should look similar to this sample:

```
<P ALIGN="CENTER">&copy; XYZ Corporation &#153;</P>
<ADDRESS>
180 Attwell Dr. Suite 130<BR>Toronto, ON<BR>M9W 6A9
</ADDRESS>
```

© XYZ Corporation TM

*180 Attwell Dr. Suite 130
Toronto, ON
M9W 6A9*

5. Save your file and then Preview your document in a browser.

Chapter 4 – Exercise 2

In this exercise you will work with more formatting elements.

Procedure

1. Add a Block Quote under each H3 and describe a bit about the project.
2. Change the color of the text in the Block Quote to blue.
3. Decrease the size of the text in the block quote.
4. Your document should look similar to this:

```
<H3>Intranet Project</H3>
```

```
<BLOCKQUOTE><FONT COLOR="#0000FF" SIZE="-1">The Intranet  
Project plays a  
very important role in increasing the communication between  
departments  
and employees here at XYZ Corp.</FONT></BLOCKQUOTE>
```

```
<H4>Intranet Project Plan</H4>
```

Intranet Project

The Intranet Project plays a very important role in increasing the communication between departments and employees here at XYZ Corp.

Intranet Project Plan

Chapter 4 – Optional Exercise

This exercise provides additional practice.

Procedure

1. Add some text about the Division or Department in the Paragraph below the H2, and try some of the other character formatting elements.
2. Center the address at the bottom of the page.

Review Questions

1. What are six elements for formatting characters?
2. What are two methods for centering text?
3. How would you add the copyright symbol to a page?
4. What are the steps required to change text color?

Summary

As a result of this chapter, you should be able to

- Change the color and size of your text.
- Use common character formatting elements.
- Align your text.
- Add special characters and symbols.
- Use additional character formatting elements.

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 definition list
4. Nest Lists

List Elements

HTML supplies several list elements. Most list elements are composed of one or more (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>
<LI>List item...</LI>
</UL>
```

- List item...
- List item...
- List item...

You have the choice of three bullet types: Disc (default), Circle, Square. These are controlled in Netscape Navigator by the "TYPE" attribute for the element.

```
<UL TYPE="SQUARE">
<LI>List item...</LI>
<LI>List item...</LI>
<LI>List item...</LI>
</UL>
```

- List item...
- List item...
- List item...

Note: Changing the bullet type is not supported in Internet Explorer 3.0 or below.

- 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.

TYPE	Numbering Style	
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, ...

You can also specify a starting number for an ordered list. The value of the START Attribute is always an Arabic number (1,2,3,4...). Changing the starting number may be useful in a situation where your list items are separated by an image or by text unrelated to the list.

```
<OL TYPE="i">
<LI>List item...</LI>
<LI>List item...</LI>
</OL>
<P>Unrelated text ... </P>
<OL TYPE="i" START="3">
<LI>List item...</LI>
</OL>
```

- i. List item...
- ii. List item...

Unrelated text ...

- iii. List item...

- 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 Descriptions (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!
```

Note the format - the definition is always placed indented on the next line to emphasize the relationship.

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>
</OL>
</LI>
<LI>List item...</LI>
</UL>
```

Results:

- List item...
- List item...
 - iii. List item...
- List item...

Chapter 5 – Exercise 1

In this exercise you will add a nested list to your document.

Procedure

1. Add an Ordered List below your first Project Plan Heading.
2. Enter two or three list items.
3. Insert another Ordered List below the second item in the first list.
4. Change the numbering scheme to lower case roman numerals.
5. Add two items.
6. Add a paragraph after your list with some text.
7. Insert another ordered list after your paragraph, and start numbering where your first list left off.
8. Your work should look something like the following:

```
<UL TYPE="SQUARE">  
<LI>Purchase the "Implementing an Intranet - The 8 Step Guide"</LI>  
<LI>Choose a Web Server Platform  
<OL TYPE="i">  
<LI>Unix</LI>  
<LI>Windows NT</LI>  
</OL>  
</LI>  
<LI>Build an alpha Intranet site.</LI>  
</UL>
```

Intranet Project Plan

- Purchase the "Implementing an Intranet - The 8 Step Guide"
 - Choose a Web Server Platform
 - i. Unix
 - ii. Windows NT
 - Build a alpha Intranet site.
9. Save and Preview your work. It would be a shame to loose it, and you have to take a look at the great page you are creating.

Chapter 5 – Optional Exercises

In this exercise you can practice creating and editing another list.

Procedure

1. Create another list this time under the second Project Plan.

Review Questions

1. Can you nest lists?
2. When would you use an ordered list instead of an unordered list?
3. Why would you use a definition list?

Summary

As a result of this chapter, you should be able to

- Create an unordered list
- Create an ordered list
- Create a definition list
- Nest your lists

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. Distinguish between supported graphics formats and unsupported graphics formats.
2. Determine which graphic format for the type of image.
3. Add images to your pages.

Supported Image Formats

GIF: Graphic Interchange Format uses a maximum of 256 colors, and uses combinations of these to create colors beyond that number. The GIF format is best for displaying images that have been designed using a graphics program, like logos, icons, and buttons.

GIF images come in two different versions and have some extra functionality that JPEGs do not. You can save GIF images in GIF 87 or GIF 89a format. GIF 89a is newer and has the following features that GIF 87 and JPEG files do not:

- **Interlacing:** if you save a GIF 89a image as interlaced, the browser will be able to display the image as it loads, getting gradually crisper and clearer until it is finished. Interlaced GIFs have slightly larger file sizes than non-interlaced GIFs, so you will have to decide whether the interlacing effect is worth the extra download time for your images.
- **Transparency:** with GIF 89a format images you can set a single color to be transparent, that is, it will allow the background color or image to show through it. Transparency is most commonly used to make the rectangular background canvas of an image invisible; this feature can be very effective in Web page design.
- **Animated GIFs:** GIF 89a images can also be animated using special software. Animated GIF images are simply a number of GIF images saved into a single file and looped. Netscape Navigator and Microsoft Internet Explorer can both display animated GIFs, but many other browsers cannot, and may not be able to display even the first image in the loop. Use animated GIFs with caution.

JPG, JPEG: Joint Photographic Expert Group, is a good format for photographs because JPEG files can contain millions of colors.

JPEG images don't give you the option of including transparency or of interlacing images, but they do allow you to specify the degree of file compression so that you can create a balance between image quality and file size.

A new addition to the JPEG format is Progressive JPEG. Pro-JPEGs boast superior compression to regular JPEGs. They also give you a wider range of quality settings. Pro-JPEGs also support interlacing.

Limited Support or Non-Supported Image Formats

The following image formats are supported by some client computers but not all, and as such are not recommended to be included in your HTML pages. You should first convert them to a supported format using a graphics editor, such as Corel Photo Paint or Adobe Photoshop.

PNG: Portable Network Graphics, is good for combinations of text and graphics within one image. PNG permits truecolor images, variable transparency, platform-independent display, and a fast 2D interlacing scheme. Currently only supported by Internet Explorer.

BMP: MS Windows BitMaP

TIFF: Tagged Image File Format

PCX: Originally developed by ZSOFT for its PC Paintbrush program, PCX is a graphics file format for graphics programs running on PCs.

Image creation is beyond the scope of this course. However, there are some excellent tutorials on creating images for the web at a web site called "Andy's Art Attack" (<http://www.andyart.com/>).

Inserting Images

**** This element defines a graphic image on the page. It is typically used for inline images

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

Alternate Text (ALT): This is a text field that describes an image or acts as a label. It is displayed when people turn the graphics off in their browsers, or when they position the cursor over a graphic image (now supported in most browsers). It is also helpful as an accessibility feature.

Alignment (ALIGN): This allows you to align the image on your page. The options include Bottom, Middle, Top, Left, Right, TextTop, ABSMiddle, Baseline, and ABSBottom.

Width (WIDTH): is the width of the image in pixels. This value can be obtained from a graphics program or can be left unspecified.

Height (HEIGHT): is the height of the image in pixels. This value can be obtained from a graphics program or can be left unspecified.

Border (BORDER): is for a border around the image, specified in pixels. If you use an image in an anchor you will want to set the border value to zero so that it is not outlined.

Additional Image Attributes

HSPACE: is for Horizontal Space on both sides of the image specified in pixels. A setting of 5 will put five pixels of invisible space on both sides of the image.

VSPACE: is for Vertical Space on the top and bottom of the image specified in pixels. A setting of 5 will put five pixels of invisible space above and below the image.

HTML Tip:

- Try to keep your image file sizes small.
- Don't rely exclusively on images to convey your information.
- Use alternate text for images.
- Some background images make it difficult to read the text of the page.

Chapter 6 – Exercise 1

You will need the files located on the diskette shipped with your materials for this exercise.

In this exercise you will add an image to your document.

Procedure

1. In your page add the following
``
The contents your SRC may be different depending on where you placed your image file. The value of the SRC must be a “web accessible” address.

The example above uses a format that is NOT acceptable for the web but will work on your local workstation.

2. Try changing the VSpace and HSpace attribute to see the affect.
3. Try adding a border.

Chapter 6 – Optional Exercise

More practice.

Procedure

1. Try adding an animated graphic image, from the graphics located in C:\HTMLfiles\Images\ directory. (“mad_hack.gif” or “teethani.gif”)

Review Questions

1. Why should you describe the image in the ALT parameter?
2. Why do you use transparency for your graphics?
3. Why would you reduce the number of colors in a graphic image?

Summary

As a result of this chapter, you should be able to

- Distinguish between supported graphics formats and unsupported graphics formats.
- Determine which graphic format for the type of image.
- Add images to your pages.

7

Anchors, URLs and Image Maps

In this chapter you will learn about Uniform Resource Locators, and how to add them as Anchors 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

Link Elements

It is normal for HTML documents to contain links to other documents, which can be located anywhere on the Web. These links are provided by URLs (Uniform Resource Locators) , which give the location and filename of a document, and the method used to access it.

The following elements represent links to other documents:

** **: The HREF attribute of the anchor element specifies a URL. If this attribute has a value, the contents of the **<A> ** element will be highlighted when the document is displayed in a browser window, and *{Clicking}* on this content will cause the browser to attempt to open the file specified by the URL.

E.g. `XtraNet University`. In this example the text "XtraNet University" represents the contents that would be highlighted as a link to the file named as the value of the HREF.

Link Types

There are three major types of links:

Internal Links: are links within a document. They help in the navigation of large documents.

Local Links: are links to documents on the local web server. Local links can be the full URL (Complete e.g. `http://www.yourdomain.com/sales/report.htm`) or partial (Relative to your current directory e.g. `/sales/report.htm`).

External Links: links to pages on other web servers. External links are always the full URL.

NOTE: Remember that the colors of the links to be displayed in the browser are controlled by attributes of the body element. E.g.

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

URLs - Uniform Resource Locators

A URL is a standard way of referencing a Service, a Host name, a Port, and a Directory-path.

Service:// Host name : Port / Directory-path

E.g. <http://www.1stvirtual.com:80/default.htm>

Not all of these components appear in each URL, as you will see when you learn about the different types of URLs for different services. But the preceding example is a good general guide.

HTTP – HyperText Transport Protocol

E.g. <http://www.1stvirtual.com/default.htm>

- The type of service is identified as http. It is the standard protocol for linking to web pages with multimedia content.
- The Host name is www, in the domain 1stvirtual, in the zone com (that identifies it as a US commercial site.)
- default.htm is the actual web page.

FTP – File Transfer Protocol

E.g. <ftp://ftp.1stvirtual.com/prizes/brkbrk.wav>

- The type of service is identified as ftp. This service is used to create links to files to be downloaded from ftp servers.
- The Host name is ftp.1stvirtual.com.
- The subdirectory is prizes.
- The file is brkbrk.wav

News – News Groups

E.g. <news:alt.internet.services>

- The type of service is identified as news, for the Usenet newsgroup service.
- The newsgroup is alt.internet.services.

This will invoke a clients newsreader program.

Gopher

E.g. `gopher://owl.trc.purdue.edu/`

- The type of service is identified as gopher. (a predecessor to the World Wide Web)
- The Host name is owl.trc.purdue.edu.

E-mail – Electronic Mail

E.g. `mailto:webmaster@1stvirtual.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 webmaster@1stvirtual.com

Telnet – Remote Terminal Emulation

E.g. `telnet://media.mit.edu`

- The type of service is identified as the telnet client program.
- The Host name is media.mit.edu.

Internal Links

Links can also be created inside large documents to simplify navigation. Today's world wants to be able to get to the information quickly, and move around easily. 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:
`Book Mark`
The NAME attribute of an anchor element specifies a location in the document that we will 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. Add the following anchor around the text link.
`Goto Book Mark`

The surfer will only see the link created in step two, they will not see the anchor created in step one.

HTML Tip

You can combine a link to another page with an internal link inside that page. `` This will link to a specified location on that page. This is commonly used to reference a particular area of a large document.

Image Maps

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

There are two types of image maps:

1. client-side
2. server-side

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

Server-side Image Maps

Server-side image maps (called ISMAP) involve a separate map file that is linked to the image by a program running on a Web server. Server-side image maps are one of two flavors NCSA - National Center for Supercomputer Applications or CERN - Conseil European pour la Recherche NuCLaire.

```
<A HREF="http://sitename/picture.map">  
<IMG ISMAP SRC="Picture.gif" BORDER=0>  
</A>
```

CERN Style: picture.map file

```
default filename.htm  
circle (123,160) 20 next.htm
```

NCSA Style: picture.map file

```
default filename.htm  
circle next.htm 123,160 123,180
```

Supported Shapes:

- Circle – Center-point radius or, for NCSA center, edge point
- Poly – List of vertices, max. 100, that define an irregular space
- Rect – Upper-left, and Lower-right
- Point – single spot, coordinates are x,y

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 browsers.

```
<IMG SRC="Picture.gif" USEMAP="#map1">  
  
<MAP NAME="map1">  
<AREA SHAPE="RECT" COORDS="9,120, 56,149"  
  HREF="whatever.htm">  
<AREA SHAPE="RECT" COORDS="100,200, 156,249"  
  HREF="wherever.htm">  
</MAP>
```

Shapes:

- RECT, or RECTANGLE: require four coordinates: X1, Y1, X2, and Y2
- CIRC, or CIRCLE: requires three coordinates centerx, centery, and radius
- POLY, or POLYGON: requires three or more pairs of coordinates

Note: There are a number of image map creation tools available for download from the web. They can greatly simplify the job of creating image maps by calculating the COORDS attributes (coordinates) for the AREA elements.

Chapter 7 – Exercise 1

In this exercise you will create a link in your document.

Procedure

1. Below the address element add the e-mail address:
webmaster@xyzcorp.com.
2. Put the e-mail address inside an anchor tag.
3. For the URL type in “mailto:**{your e-mail address}**”.
4. Center the link.

```
<CENTER> <A HREF=mailto:your@email.address>  
webmaster@xyzcorp.com </A> </CENTER>
```

Review Questions

1. What are the components of a URL?
2. What protocols are supported by URLs?
3. What is the difference between an internal and external link?
4. What is the difference between complete and relative URLs?
5. How do you set the link colors?

Summary

As a result of this chapter, you should be able to

- Insert links into documents
- Define Link Types
- Define URL
- List some commonly used URLs
- Work with Image Maps

T

ables

In this chapter you will learn that tables have many uses in HTML. The most obvious is a table of data, but tables can also be used to place graphics on a page at just the right spot or to format text and form input boxes, or simulate columns.

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; Table Row <TR></TR>, Table Header <TH></TH>, Table Data <TD> </TD>, and Caption <CAPTION> </CAPTION>. The Table Row elements usually contain Table Header elements or Table Data elements. The Table Header and Table Data elements can contain several of the body elements, which allows for rich formatting of the data in the table. A brief HTML code sample follows:

```
<TABLE BORDER="1">
<TR>
<TH>Column 1 Header</TH>
<TH>Column 2 Header</TH>
</TR>
<TR>
<TD>Row 1 - Col 1 </TD>
<TD>Row 1 - Col 2 </TD>
</TR>
<TR>
<TD>Row 2 - Col 1 </TD>
<TD>Row 2 - Col 2 </TD>
</TR>
<TR>
<TD>Row 3 - Col 1 </TD>
<TD>Row 3 - Col 2 </TD>
</TR>
</TABLE>
```

Results in Navigator:

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

Tables are used a great deal in the creation of web pages. They allow you to create boundaries that make positioning easier. They are great for formatting forms, a topic you will study later in the course.

Table Attributes

BGColor: Some browsers support background colors in a table. The color you select will be expressed as a hexadecimal red-green-blue value. You can enter this value directly or you can enter one of the standard Windows color names. (these color names are currently supported only by Microsoft Internet Explorer).

Width: you can specify the table width as an absolute number of pixels or a percentage of the document width. The width corresponds to the WIDTH attribute of the TABLE element. You can set the width for table cells as well.

Border: the lines that form the boundary of each table cell when the file is displayed in a browser. You can choose a numerical value for the border width, which specifies the border in pixels, or 'BORDER' (causing the browser to draw the default border). The table border corresponds to the BORDER attribute of the TABLE element. A setting of BORDER="0" will make the border disappear.

Some browsers do not draw borders around empty table cells. You may want to insert a NBSP in each cell when you create the table so that each cell will display border and background color.

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

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. The alignment attribute of the <TABLE> is not well supported and you will want to use <DIV> or <CENTER> to control the position of your table in your document.

Background: Background Image, will be tiled in Internet Explorer 3.0 and above.

BorderColor: The color of the border around the table. This is supported properly by Navigator 4.0, and Internet Explorer 3.0 and above.

BorderColorLight: Light color used to outline two sides of a cell or the table, only supported in Internet Explorer 2.0 and above.

BorderColorDark: Dark color used to outline two sides of a cell or the table, only supported in Internet Explorer 2.0 and above.

Table Caption

A Table caption allows you to specify a line of text that will appear centered above or below the table. This can act like a title for the table.

```
<TABLE BORDER="1" CELLPADDING="2">  
<CAPTION ALIGN="BOTTOM">Label for my Table</CAPTION>  
<TR>
```

The caption element has one attribute ALIGN that can be either TOP (above the table) or BOTTOM (below the table). You can use standard character formatting codes inside the CAPTION element.

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

Label for my 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.

If you simply wish to bold the text, we suggest that you insert a B or STRONG element inside the table cell(s) instead of changing the table cell element.

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.

Background: Background Image, will be tiled in Internet Explorer 3.0 and above.

BGColor: Some browsers support background colors. The color you select will be expressed as a hexadecimal red-green-blue value.

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.

A table example using ROWSPAN and COLSPAN. Notice that in the case of the COLSPAN that there is only one TD tag set. In the case of the ROWSPAN, the row below is less one TD tag set.

```
<TABLE BORDER="1" CELLPADDING="2">
<CAPTION ALIGN="BOTTOM">Label for my Table</CAPTION>
<TR>
<TH>Column 1 Header</TH>
<TH>Column 2 Header</TH>
</TR>
<TR>
<TD COLSPAN="2">Row 1 - Col 1 </TD>
</TR>
<TR>
<TD ROWSPAN="2">Row 2 - Col 1 </TD>
<TD>Row 2 - Col 2 </TD>
</TR>
<TR>
<TD>Row 3 - Col 2 </TD>
</TR>
</TABLE>
```

The results in Navigator:

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

Label for my Table

HTML Tip

The following method of centering a table is supported by many browsers. First you insert the **<CENTER> </CENTER>** element, then drag the table inside the **<CENTER> </CENTER>** element.

Chapter 8 – Exercise 1

In this exercise you will work with a Table element.

Procedure

1. Create a table above the bottom Horizontal Rule.
2. Change its background color to yellow.
3. Increase its border to 5 pixels.
4. Change the Top Row to a table Heading, and add some headings.
5. Add some text/data to your table.
6. Center the table using the Center element.
7. Center the address using the Center element (if not already done).
8. Save and Preview.

- Go to dinner with Dell
- Play golf with IBM

This is a table	Column 2
Row 2	Column 2 Row 2

© XYZ Corporation™
3-1750 The Queenway
Etobicoke, ON
Canada
M9C 5H5

Review Questions

1. How would you create a TABLE with no border?
2. What is Cell Spacing?
3. What is Cell Padding?
4. How would a row of Table Header cells be formatted?

Summary

As a result of this chapter, you should be able to

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

Frames

Frames are a relatively new addition to the HTML standard, having been included in release 3.0 and therefore not all browsers support this group of elements. Popular browsers, such as Netscape and Internet Explorer releases 3.0 and higher do provide full frames support.

Objectives

Upon completing this section, you should be able to

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

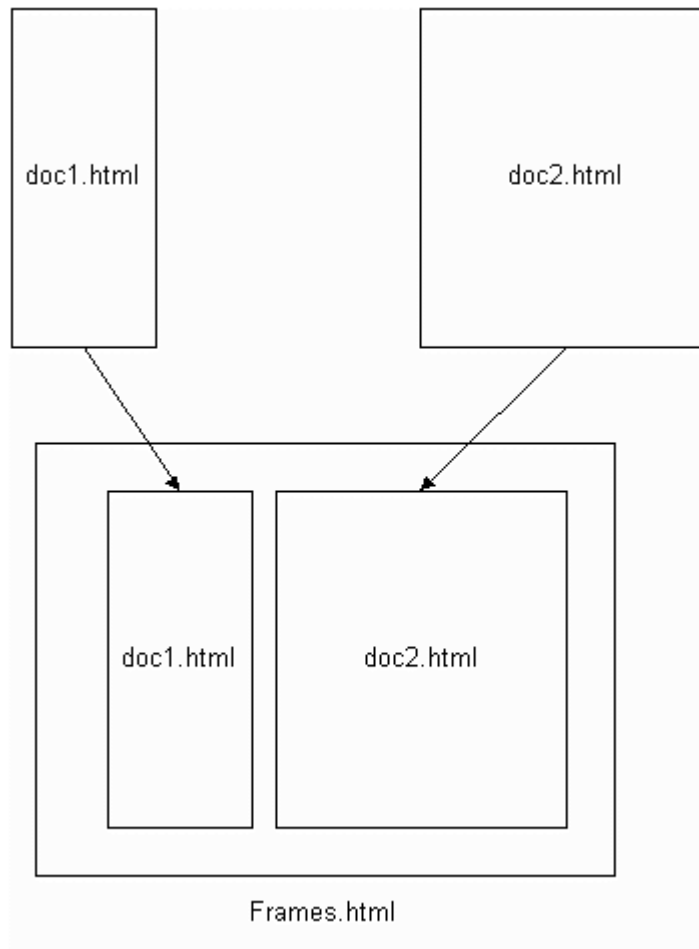
Prerequisites

- Internet Explorer 3.0 or greater
- Navigator 2.0 or greater
- Frames are Platform Independent

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.



In this chapter we will describe the tags that are used to create the Frames.html page that will break up the browser window into multiple windowpanes.

Frame Page Architecture

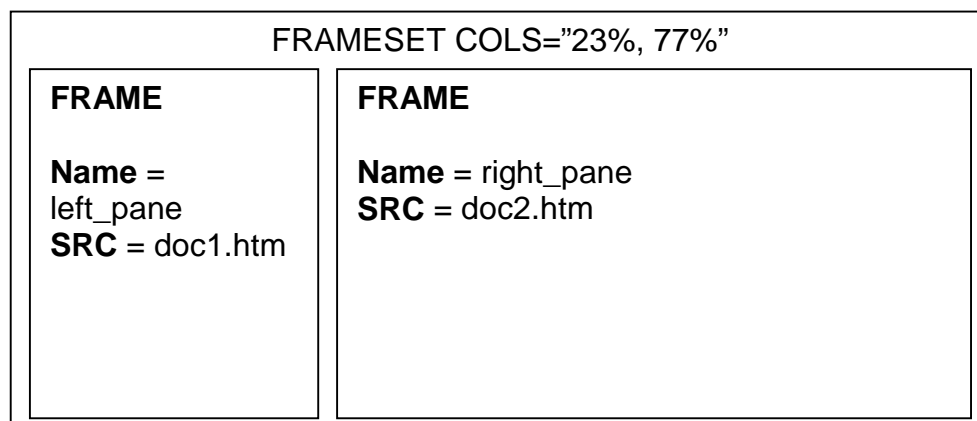
A `<FRAMESET>` element is placed in the HTML document before the `<BODY>` element. The `<FRAMESET>` describes the amount of screen real estate given to each windowpane by dividing the screen into ROWS or COLS. The `<FRAMESET>` will then contain `<FRAME>` elements, one per division of the browser window.

```
<HTML>
<HEAD>
<TITLE>Framed Page</TITLE>
</HEAD>

<FRAMESET COLS="23%,77%">
<FRAME SRC="doc1.htm" NAME="left_pane" SCROLLING="NO">
<FRAME SRC="doc2.htm" NAME="right_pane" SCROLLING="YES">
<NOFRAMES>
<BODY>
<!-- Document for browsers that do not support Frames -->
</BODY>
</NOFRAMES>
</FRAMESET>
</HTML>
```

Older browsers are written to ignore tags/elements that they don't understand so they will ignore the `<FRAMESET>`, `<FRAME>`, and `<NOFRAMES>` tags. As a result older browsers will be able to display the contents placed between the `<BODY></BODY>` tags. Many authors use a message indicating that the site is built using frames and that the surfer should upgrade their browser. You can actually put content in the `<BODY>` tags such as navigation to a non-frames version of your site.

The diagram below is a graphical view of the document described above.



Creating a Frames Page

<FRAMESET>

The FRAMESET element creates divisions in the browser window in a single direction. This allows you to define the 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,*"

The latter enables the developer to allocate all unassigned vertical space proportionately. Values are given within quotation marks and separated by commas in HTML source.

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 height, i.e. "75%,25%"
- proportional values using the asterisk (*). This is often combined with a value in pixels e.g. "480,*"

The latter enables the developer to allocate all unassigned vertical space proportionately. Values are given within quotation marks and separated by commas in HTML source.

Other Key Attributes of the FRAMESET Element:

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:

Possible values 0, 1. A setting of zero will create a borderless frame.

BORDERCOLOR:

This attribute allows you to choose a color for your border. This attribute is rarely used.

<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. In order for a link to display the retrieved file in a frame other than the frame in which the link was {Clicked} in, the targeted frame must be referenced by its name.

Works in conjunction with the TARGET attribute of the <A>, <AREA>, <BASE>, and <FORM> tags. All names must begin with an alphanumeric value and not the underscore character. The exception is the special target names illustrated later in this chapter.

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:

- yes – always display scroll bar(s)
- no – never display scroll bar(s)
- auto – browser will decide based on frame contents

The default value for this attribute is “auto”.

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>

Frame-capable browsers ignore all HTML within this tag including the contents of the BODY element. Browsers that don't support frames will ignore all frame elements, and interpret the <NOFRAMES> content, beginning with the <BODY> element.

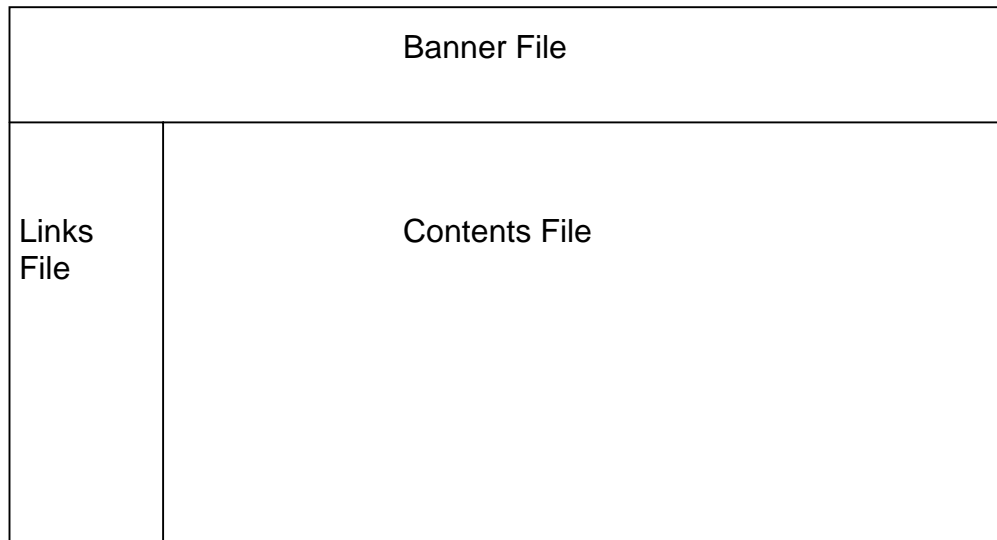
This element does not have any attributes.

```
<HTML>
<HEAD>
<TITLE>Framed Page</TITLE>
</HEAD>

<FRAMESET COLS="23%,77%">
<FRAME SRC="doc1.htm" NAME="left_pane" SCROLLING="NO">
<FRAME SRC="doc2.htm" NAME="right_pane" SCROLLING="YES">
<NOFRAMES>
<BODY>
<P>This is a Framed page. Upgrade your browser to support frames.</P>
</BODY>
</NOFRAMES>
</FRAMESET>
</HTML>
```

Compound FRAMESET Divisions

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



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 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.

```
<HTML>
<HEAD>
<TITLE>Compound Frames Page<TITLE>
</HEAD>

<FRAMESET ROWS="120,*">
<FRAME SRC="banner_file.htm">
<FRAMESET COLS="120,*">
<FRAME SRC="links_file.htm">
<FRAME SRC="content_file.htm" NAME="content">
<NOFRAMES><BODY>default message</BODY></NOFRAMES>
</FRAMESET>
</FRAMESET>
</HTML>
```

Targets

When you are creating 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.

For example if we were to place a link in doc1.htm that linked to doc3.htm and we wanted doc3.htm to be displayed in the right windowpane; the HTML code would appear in doc1.htm as follows:

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

Warning. If you fail to specify a TARGET frame name the linked document will be displayed in the current frame.

Warning. Frame NAMES and TARGET values are case sensitive and must match exactly to work. If they don't match it will create a new browser window like with the special target "_blank". (see special targets)

The following elements have TARGET attributes:

<A> sample - ``

<AREA> sample - `<AREA SHAPE="RECT" COORDS="5,6,2,4" HREF="file.htm" TARGET="Frame-window-name">`

<BASE> sample - `<BASE TARGET="Frame-window-name">`

<FORM> sample - `<FORM ACTION="cgi-bin/script" TARGET="Frame-window-name">`

Special Targets

There are 4 special target names that cannot be assigned by the NAME attribute of the FRAME tag. Each of these reserved names serves a special function when used with the TARGET attribute. These special names and functions are:

TARGET="_top"

This loads the linked document into the full browser window with the URL specified by the HREF attribute. This is particularly useful for moving between a frames environment and a non-frames environment.

TARGET="_blank"

Opens a new browser window and loads the document specified in the URL attribute into that new window. The window is not named.

TARGET="_self"

Loads the document in the same window where the anchor was *{Clicked}*. This is the default setting for linking elements. This attribute would generally be used to override the TARGET attribute of the <BASE> tag.

TARGET="_parent"

The _parent frame is a prior frameset that the current frameset was "spawned" from. If there isn't one then it is the browser window.

```
<HTML>
<HEAD>
<TITLE>Document 1, doc1.htm</TITLE>
</HEAD>
<BODY>
<P>
Links<BR>
<A HREF="doc3.htm" TARGET="right_pane">Document 3 in the right
pane</A><BR>
<A HREF="doc4.htm" TARGET="_top">Document 4 in top</A><BR>
<A HREF="doc4.htm" TARGET="_blank">Document 4 in blank</A><BR>
<A HREF="doc4.htm" TARGET="_self">Document 4 in self</A><BR>
</P>
</BODY>
</HTML>
```

Chapter 9 – Exercise 1

In this exercise you will work with a frames based page.

Procedure

In your **browser** open the file called frames.htm, it is located in the c:\HTMLfiles\Frames\ directory.

1. Click on the links on in the frame on the left, notice how the content on the right changes.

Chapter 9 – Exercise 2

In this exercise you will add a link to the page in the frame on the left.

Procedure

In **Notepad**, open the Links.htm page in the c:\HTMLfiles\Frames\ directory.

1. Place the following anchor tag set around the word Success.
` `
2. Save the file, and **reload** the frames.htm document in your browser. *{Click}* on the word Success and observe what happens.

Review Questions

1. What is the TARGET attribute used for?
2. What are two attributes of the FRAMESET element?
3. What is the NOFRAMES element used for?
4. What does the SRC attribute of the FRAME element specify?

Summary

As a result of this chapter, you should be able to

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

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 that information.

Objectives

Upon completing this section, you should be able to

1. Create a FORM.
2. Add elements to a FORM.
3. Define CGI.
4. Describe the purpose of a CGI Application.
5. Specify an action for the FORM.

Prerequisites

- 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.

NOTE: The FORM element has no formatting attributes.

For example:

```
<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"><BR>
Last Name: <INPUT TYPE="TEXT" NAME="lname"
MAXLENGTH="50"></P>
<P><INPUT TYPE="SUBMIT" NAME="Submit1" VALUE="Send
Info"></P>
</FORM>
</BODY>
</HTML>
```

The attributes of the <FORM> element are:

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.

ENCTYPE: determines the mechanism used to encode the form contents. You can leave this attribute as default (unspecified) unless you are creating a file upload field.

NAME: is a form name used by VB or JAVA scripts.

TARGET: is the target frame where the response page will show up.

The rest of the form elements are used to create the form seen in the browser and collect the data to be sent to the application.

HTML Tip

A good way to format a FORM and its sub elements is to use a table. With a table you have more control over the layout of your page. The table can be created inside the <FORM></FORM> tags.

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>`.

<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 by default
SIZE =	Number of characters in text field
MAXLENGTH =	Maximum number of characters accepted

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
MAXSIZE: 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.

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.

MAXSIZE: 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: 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 to a value expected by the CGI application.

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 buttons allow 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.

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:

NAME: is the name of the variable to be sent to the CGI application.

SIZE: is the size of the text box in characters.


MAXSIZE: is the maximum size of the input in the textbox in characters.

ENCTYPE: is a list of (MIME) types that the field can be used to upload.

This type of input is a valid HTML 3.2 construct; however, it is not widely used. You will also have to change the ENCTYPE attribute of the `<FORM>` element to "multipart/form-data"

Push Button: This element would be used with either JavaScript or VBScript 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.

Note: You will see an application of a Push Button in some of the following chapters.

Submit: Every set of Form tags requires a Submit button. This is the element that 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.

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


```
<INPUT TYPE="IMAGE" SRC="URL">
```

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.

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.

Other Elements used in Forms

Text Area

`<TEXTAREA>` `</TEXTAREA>` is an element that allows for free form text entry.



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.
- WRAP:** defaults to OFF. You should set wrap to "VIRTUAL" or "PHYSICAL" so that the text wraps in the browser display as the surfer types.

Select

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

The Select element's 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.

Drop Down List:



NAME: is the name of the variable to be sent to the CGI application.

SIZE: 1

List Box:



NAME: is the name of the variable to be sent to the CGI application.

SIZE: is greater than one.

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.

```
<P><SELECT SIZE="1">
<OPTION VALUE="1" SELECTED="SELECTED">Item
1</OPTION>
<OPTION VALUE="2">Item 2</OPTION></SELECT></P>
```

Common Gateway Interface (CGI)

When the user has finished filling in the form and presses "Submit" the data is sent to the application on the server specified in the ACTION attribute of the form element.

The application is commonly referred to as a CGI application/program. It resides and runs on the web server. It is typically but not always stored in a directory called cgi-bin.

The application can be written in any language; however, it must be one supported by your webserver's operating system and webserver software. A very popular language for creating CGI applications is PERL (Practical Extraction Report Language).

PERL is an interpreted language with rich text manipulation characteristics. Because it is interpreted its performance compared to a compiled CGI application is slower, thus negatively affecting server performance. It has been ported to most popular operating systems and is currently available for most versions of UNIX, Windows NT, and Windows 95. The actual application is commonly referred to as a script.

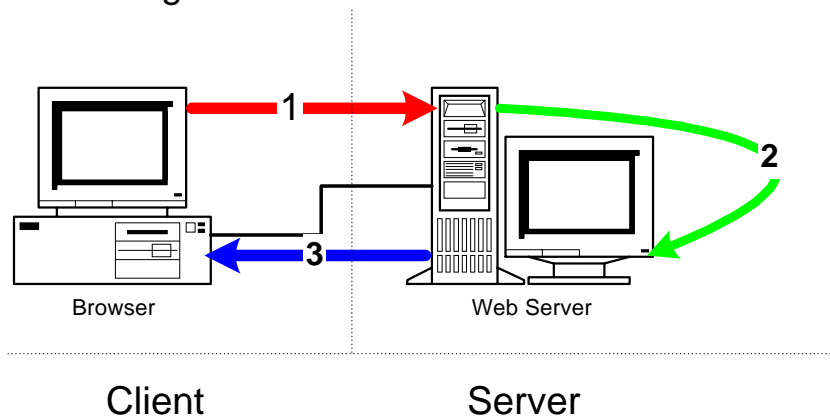
CGI Scripts

CGI Scripts process the form's data, and send a response back to the user.

They can be written to calculate numbers as in a sales order. They can format data and put it into a database such as a mailing list or guest book. Depending on the situation, a CGI Script can be written to do almost anything.

Intranet Application Development Architectures

1. Simple CGI Program



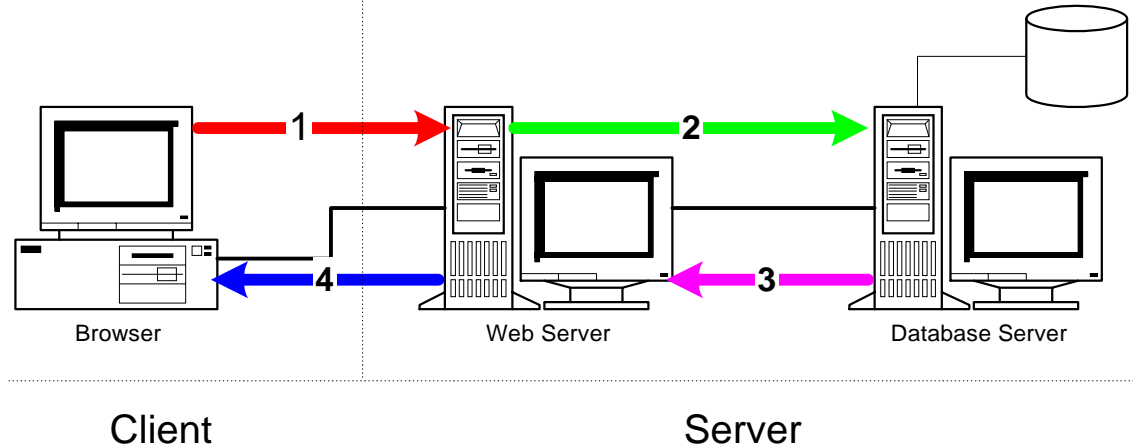
1. The client fills out an HTML document containing the <FORM> element and presses the submit button. This sends the data the user entered to the web server.

2. The web server receives the data and passes it to the program specified in the ACTION attribute of the <FORM> element. This program is called a CGI (Common Gateway Interface) program. It can be written in a number of languages; VBScript, JavaScript, PERL, C++, etc. It can perform a number of different functions.

3. The CGI program creates a new HTML document on the fly and sends it to the client.

Browser	Web Server	CGI Program
Netscape Navigator	Netscape Enterprise Server	- NSAPI -C++ - PERL - Server side JavaScript
Microsoft Internet Explorer	Microsoft Internet Information Server	- ISAPI -C++ - PERL - Active Server Pages (ASP) JScript or VBScript

2. Three-Tier Web Application Development



1. The client fills out an HTML document containing the <FORM> element and presses the submit button. This sends the data the user entered to the web server.

2. The web server receives the data and passes it to the program specified in the ACTION attribute of the <FORM> element. This program is called a CGI (Common Gateway Interface) program. It can be written in a number of languages; VBScript, JavaScript, PERL, C++, etc.

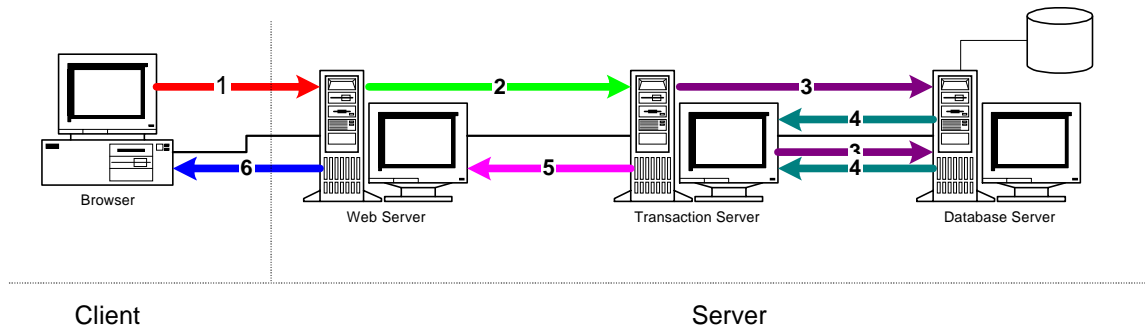
The program in this case creates a command and issues it to the database server.

3. The Database Server receives the command and issues it against the database. The data returned from the command is then sent back to the CGI application running on the web server.

4. The CGI program creates a new HTML document on the fly with the data returned from the database server and sends it to the client.

Browser	Web Server	CGI Program	Database Server
Netscape Navigator	Netscape Enterprise Server	- NSAPI -C++ - PERL - Server side JavaScript	- Oracle - Informix - Sybase - Microsoft SQL - Other ODBC compliant
Microsoft Internet Explorer	Microsoft Internet Information Server	- ISAPI -C++ - PERL - Active Server Pages (ASP) - JScript or VBScript	

3. Four-Tier Web Application Development



1. The client fills out an HTML document containing the <FORM> element and presses the submit button. This sends the data the user entered to the web server.

2. The web server receives the data and passes it to the program specified in the ACTION attribute of the <FORM> element. This program is called a CGI (Common Gateway Interface) program.

In this case the program invokes a transaction on the transaction server.

3. The Database Server receives a command from the transaction server and issues it against the database.

4. The data returned from the command is then sent back to the transaction, this cycle will continue until the transaction fails or completes.

5. When the transaction completes successfully or unsuccessfully it passes the result to the web server.

6. The CGI program running on the web server then creates a new HTML document on the fly with the data returned from the transaction server and sends it to the client.

Transaction Server Choices:

- Microsoft Transaction Server
- Netscape Application Server
- Oracle Application Server

Chapter 10 - Exercise 1 - Creating an Order Form

In this exercise, you will create a partial order form:

Procedure

In steps 1 through 5 you will add the form element, and put in a hidden field.

1. In Notepad, open a new file.
2. Insert the basic elements – HTML, HEAD and BODY.
3. Insert a FORM element inside the BODY element
4. Insert a “Hidden” INPUT element
5. In the “Hidden” INPUT set the NAME attribute as “SecretPassword” and the VALUE as “OpenSesame”.

Your code should look like this:

```
<HTML>
<HEAD>
<TITLE>Partial Order Form</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<FORM>
<P><INPUT TYPE="HIDDEN" NAME="SecretPassword"
VALUE="OpenSesame"></P>
</FORM>
</BODY>
</HTML>
```

In steps 6 through 9 you will add a select element, and add the option elements that make up a drop down list of Greetings.

6. Type "Greeting:" after the end of the hidden Input tag.
7. Put a space after the colon and insert a "Drop Down Box" (Select Element with SIZE set at 1) Give your Select element a NAME of "Greeting"
8. Insert OPTION elements to specify the list items as follows:

List Item: **Mr.** Value: **Mr.**
List Item: **Mrs.** Value: **Mrs.**
List Item: **Ms.** Value: **Ms.**
List Item: **Dr.** Value: **Dr.**
9. Insert a Break after the Select element.

Sample Code:

```
<P>  
<INPUT TYPE="HIDDEN" NAME="SecretPassword"  
VALUE="OpenSesame">Greeting:  
<SELECT NAME="Greeting" SIZE="1">  
<OPTION VALUE="Mr.">Mr.</OPTION>  
<OPTION VALUE="Mrs.">Mrs.</OPTION>  
<OPTION VALUE="Ms.">Ms.</OPTION>  
<OPTION VALUE="Dr.">Dr.</OPTION>  
</SELECT>  
<BR>  
</P>
```

In steps 10 through 13 you will add a Text Box, for the user to enter their First Name.

10. After the BR, type in "First Name:"

11. Insert a Text box

12. Set the Text Box attributes:

Name: **FirstName**

Size: **20**

Max. Length: **25**

13. Insert a Break after the Text Box

Sample Code:

```
<P>  
<INPUT TYPE="HIDDEN" NAME="SecretPassword"  
VALUE="OpenSesame">Greeting:  
<SELECT NAME="Greeting" SIZE="1">  
<OPTION VALUE="Mr.">Mr.</OPTION>  
<OPTION VALUE="Mrs.">Mrs.</OPTION>  
<OPTION VALUE="Ms.">Ms.</OPTION>  
<OPTION VALUE="Dr.">Dr.</OPTION>  
</SELECT>  
<BR>  
First Name:<INPUT TYPE="TEXT" NAME="FirstName" SIZE="20"  
MAXLENGTH="25">  
<BR>  
</P>
```

In steps 14 through 17 you will add a Text Box, for the user to enter their Last Name.

14. After the BR, type in "Last Name:"

15. Insert a Text box

16. Set the Text Box attributes:

Name: **LastName**

Size: **30**

Max. Length: **35**

17. Insert a Break after the Text Box.

Sample Code:

```
<P>
<INPUT TYPE="HIDDEN" NAME="SecretPassword"
VALUE="OpenSesame">Greeting:
<SELECT NAME="Greeting" SIZE="1">
<OPTION VALUE="Mr.">Mr.</OPTION>
<OPTION VALUE="Mrs.">Mrs.</OPTION>
<OPTION VALUE="Ms.">Ms.</OPTION>
<OPTION VALUE="Dr.">Dr.</OPTION>
</SELECT>
<BR>
First Name:<INPUT TYPE="TEXT" NAME="FirstName" SIZE="20"
MAXLENGTH="25"><BR>
Last Name:<INPUT TYPE="TEXT" NAME="LastName" SIZE="30"
MAXLENGTH="35">
<BR>
</P>
```

In steps 18 through 21 you will add a Text Box, for the user to enter their e-mail address.

18. After the BR, type in "E-mail Address:"

19. Insert a Text box

20. Set the Text Box attributes:

Name: **Email**

Size: **30**

Max. Length: **50**

21. Insert a Break after the Text Box.

22. Insert another Break.

Sample Code:

```
<P>
<INPUT TYPE="HIDDEN" NAME="SecretPassword"
VALUE="OpenSesame">Greeting:
<SELECT NAME="Greeting" SIZE="1">
<OPTION VALUE="Mr.">Mr.</OPTION>
<OPTION VALUE="Mrs.">Mrs.</OPTION>
<OPTION VALUE="Ms.">Ms.</OPTION>
<OPTION VALUE="Dr.">Dr.</OPTION>
</SELECT>
<BR>
First Name:<INPUT TYPE="TEXT" NAME="FirstName" SIZE="20"
MAXLENGTH="25">
<BR>
Last Name:<INPUT TYPE="TEXT" NAME="LastName" SIZE="30"
MAXLENGTH="35">
<BR>
E-mail Address:<INPUT TYPE="TEXT" NAME="Email" SIZE="30"
MAXLENGTH="50">
<BR>
<BR>
</P>
```

In steps 23 through 34 you will add Radio Buttons, for the user to select a preferred method of follow up.

23. Type in “What type of response would you prefer?”

24. Insert a Radio Button.

25. Set the Radio Button attributes:

Name: **Response**
Value: **Phone**
Checked: **Checked**

26. Type “Phone”

27. Insert another Radio Button

28. Set the Radio Button attributes:

Name: **Response**
Value: **Fax**

29. Type “Fax”

30. Insert another Radio Button

31. Set the Radio Button attributes:

Name: **Response**
Value: **Email**

32. Type “E-mail”

33. Insert a Break.

34. Insert another Break.

Sample Code:

```
<HTML>
<HEAD>
<TITLE>Partial Order Form</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<FORM>
<P>
<INPUT TYPE="HIDDEN" NAME="SecretPassword"
VALUE="OpenSesame">Greeting:
<SELECT NAME="Greeting" SIZE="1">
<OPTION VALUE="Mr.">Mr.</OPTION>
<OPTION VALUE="Mrs.">Mrs.</OPTION>
<OPTION VALUE="Ms.">Ms.</OPTION>
<OPTION VALUE="Dr.">Dr.</OPTION>
</SELECT>
<BR>
First Name:<INPUT TYPE="TEXT" NAME="FirstName" SIZE="20"
MAXLENGTH="25">
<BR>
Last Name:<INPUT TYPE="TEXT" NAME="LastName" SIZE="30"
MAXLENGTH="35">
<BR>
E-mail Address:<INPUT TYPE="TEXT" NAME="Email" SIZE="30"
MAXLENGTH="50">
<BR>
<BR>
What type of response would you prefer?<INPUT TYPE="RADIO"
NAME="Response"
VALUE="Phone" CHECKED="CHECKED"> Phone
<INPUT TYPE="RADIO" NAME="Response" VALUE="Fax">Fax
<INPUT TYPE="RADIO" NAME="Response" VALUE="Email">E-
mail<BR>
<BR>
</P>
</FORM>
</BODY>
</HTML>
```

In steps 35 through 48 you will add Check Boxes, for the user to select the options they would like to purchase.

35. Type "I would like the following options:"

36. Insert a Break.

37. Type "Leather interior"

38. Insert a Check Box

39. Set the Check Box attributes:

Name: **Leather**

40. Insert a Break.

41. Type "CD Player"

42. Set the Check Box attributes:

Name: **CD**

43. Insert a Break.

44. Type "Sunroof"

45. Insert a Check Box

46. Set the Check Box attributes:

Name: **Sunroof**

47. Insert a Break.

48. Insert another Break.

Sample Code:

```
<HTML>
<HEAD>
<TITLE>Partial Order Form</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<FORM>
<P><INPUT TYPE="HIDDEN" NAME="SecretPassword"
VALUE="OpenSesame">Greeting:
<SELECT NAME="Greeting" SIZE="1"><OPTION
VALUE="Mr.">Mr.</OPTION>
<OPTION VALUE="Mrs.">Mrs.</OPTION><OPTION
VALUE="Ms.">Ms.</OPTION>
<OPTION VALUE="Dr.">Dr.</OPTION></SELECT>
<BR>
First Name:<INPUT TYPE="TEXT" NAME="FirstName" SIZE="20"
MAXLENGTH="25"><BR>
Last Name:<INPUT TYPE="TEXT" NAME="LastName" SIZE="30"
MAXLENGTH="35"><BR>
E-mail Address:<INPUT TYPE="TEXT" NAME="Email" SIZE="30"
MAXLENGTH="50"><BR>
<BR>
What type of response would you prefer?<INPUT TYPE="RADIO"
NAME="Response"
VALUE="Phone" CHECKED="CHECKED"> Phone
<INPUT TYPE="RADIO" NAME="Response" VALUE="Fax">Fax
<INPUT TYPE="RADIO" NAME="Response" VALUE="Email">E-
mail<BR>
<BR>
I would like the following options:<BR>
Leather interior <INPUT TYPE="CHECKBOX" NAME="Leather"> <BR>
CD Player <INPUT TYPE="CHECKBOX" NAME="CD"><BR>
Sunroof <INPUT TYPE="CHECKBOX" NAME="Sunroof"><BR>
<BR>
</P>
</FORM>
</BODY>
</HTML>
```

In steps 49 through 53 you will add a Text Area, for the user to make comments about their purchase.

49. Type "Comments:"

50. Insert a Break.

51. Insert a Textarea element

52. Set the attributes of the Textarea element:

Name: **Comments**

Width: **30**

Rows: **4**

Wrap: **virtual**

53. Insert a Break.

Sample Code:

```
<BR>
I would like the following options:<BR>
Leather interior <INPUT TYPE="CHECKBOX" NAME="Leather"> <BR>
CD Player <INPUT TYPE="CHECKBOX" NAME="CD"><BR>
Sunroof <INPUT TYPE="CHECKBOX" NAME="Sunroof"><BR>
<BR>
Comments:<BR>
<TEXTAREA NAME="Comments" ROWS="4" COLS="30"
WRAP="Virtual"></TEXTAREA>
<BR>
```

In steps 54 through 55 you add the Submit Button.

54. Insert a Submit Button

55. Set the attributes of the Submit Button:

Value: **Submit Request**

Sample Code:

```
<TEXTAREA NAME="Comments" ROWS="4" COLS="30"  
WRAP="Virtual"></TEXTAREA>  
<BR>  
<INPUT TYPE="SUBMIT" NAME="Submit Request">
```

In steps 56 through 59 you will specify the Action for the <FORM>, and test it.

56. Now set up where the form will go.

57. Set the attributes of the FORM element:

58. For testing purposes, you may use the "Form Test" CGI Script located on XtraNet University Web Server. In the ACTION Attribute type:
<http://www.xnu.com/formtest.asp>

59. Save the file.

Sample Code:

```
<BODY BGCOLOR="#FFFFFF">
<FORM ACTION="http://www.xnu.com/formtest.asp">
<P><INPUT TYPE="HIDDEN" NAME="SecretPassword"
VALUE="OpenSesame">Greeting:
<SELECT NAME="Greeting" SIZE="1"><OPTION
VALUE="Mr.">Mr.</OPTION>
<OPTION VALUE="Mrs.">Mrs.</OPTION><OPTION
VALUE="Ms.">Ms.</OPTION>
<OPTION VALUE="Dr.">Dr.</OPTION></SELECT>
```

60. Preview your document in the browser and try filling out and submitting the order form.

Greeting:

First Name:

Last Name:

E-mail Addr:

What type of response would you prefer? Phone Fax E-mail

I would like the following options:

Leather interior

CD Player

Sunroof

Comments:

Results:

SecretPassword = OpenSesame
Select1 = Mr.
FirstName = Sample
LastName = Student
Email = student@xnu.com
Response = Email
Leather = on
CD = on
Sunroof = on
Comments = Please Quote me on all of the options
Submit1 = Submit Request

Note: When using forms on your Intranet, a CGI script is required to process the information submitted.

Review Questions

1. When would you use Check boxes over Option/Radio Buttons?
2. Do you need a submit button?
3. Where does the data entered in the FORM go?

Summary

As a result of this chapter, you should be able to

- Create a FORM.
- Add elements to a FORM.
- Define CGI.
- Describe the purpose of a CGI Application.
- Specify an action for the FORM.

JAVA and JAVAScript Introduction

In this chapter you will be introduced to what JAVA is, and how to add JAVA Applets to your web pages. You will also be introduced to JAVA Script, and how to add it to your web pages.

Objectives

Upon completing this section, you should be able to

1. Describe JAVA.
2. Add a JAVA Applet to a page.
3. Describe JAVAScript.
4. Add JAVAScript to an HTML page.

Prerequisites

- Internet Explorer 3.0 or greater (IE 3.0 Supports JAVA with limited support for JAVAScript)
- Navigator 2.0 or greater
- Java is Platform Independent

JAVA

The Java programming language was developed by Sun Microsystems. It was developed to be cross-platform and device-independent, meaning the compiled program should run on Macs, PCs, and Unix machines. It is a powerful, full-featured object oriented programming language. The browser creates an environment for JAVA applets to run in called a JAVA Virtual Machine. This provides platform independence for JAVA Applets.

An applet, for those not familiar with the term, is the name of the mini-applications created with the JAVA programming language. The creation of applets is beyond the scope of this course. Applets are separate compiled programs with the file extension of “class”. They are downloaded with the page through the use of the <APPLET> tag. The HTML syntax for the APPLETTAG, and its associated PARAM tag is as follows:

<APPLET>

Designates a Java applet. Instead of containing the code necessary to perform a task, this tag contains a reference to the code needed to perform the task.

The Applet element has one sub-element, the <PARAM> element, which passes values to the applet that affect the operation of the program.

Applet has the following attributes: >

CODEBASE: Directory or folder where the applet(s) are located.

CODE: Defines the compiled applet that is to be loaded.

WIDTH: Determines the width of the area, in pixels, reserved to display the applet in the browser.

HEIGHT: Determines the height of the area, in pixels, reserved to display the applet in the browser.

<PARAM>

This tag passes a parameter to the applet. It has two attributes:

NAME: Names the parameter for recognition by the applet.

VALUE: Defines a value for the parameter specified by the NAME attribute. All data is passed to the applet as a string variable.

</PARAM>

</APPLET>

When using Java in your pages, it is important to be aware that its support is not universal. If, like many authors, you are showcasing your use of JAVA on the page, it is advisable to include text in the document that alerts viewers to this fact. This is especially true when the applet itself is the centerpiece of the document.

Since JAVA applets must be downloaded with your HTML page they increase download time. Once downloaded, the JAVA Virtual Machine must be started and the applet will then start to run. This is usually indicated in the status line of most browsers.

JAVAScript

JavaScript is a programming language that allows scripting of events, objects and actions to create Internet applications. It uses two elements the <SCRIPT> </SCRIPT> and <!-- the comment - ->. The script element lets the browser know what type of script it is with its LANGUAGE attribute. The comment tags enclose the actual JavaScript code so that browsers will ignore the code if they don't support scripting.

Since JavaScript is embedded in the HTML page it is interpreted at the client when the page is loaded. This is in contrast to JAVA applets that are compiled programs that are downloaded from the web server to the browser and run inside the browser.

```
<HTML>
<HEAD>
<TITLE>JavaScript Example</TITLE>
<SCRIPT LANGUAGE="JAVASCRIPT">
<!-- hide JavaScript code from browsers that are not JavaScript enabled
function getanswer() {
    var x = parseInt(document.FORM1.FirstNum.value);
    var y = parseInt(document.FORM1.SecondNum.value);
    var sum = x + y;

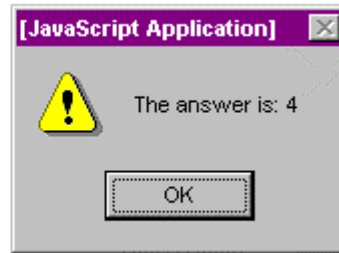
    alert("The answer is: " + sum);
}
//end hiding of JavaScript code -->
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="FORM1">
    <INPUT TYPE="TEXT" NAME="FirstNum"><BR>
    <INPUT TYPE="TEXT" NAME="SecondNum"><BR>
    <INPUT TYPE="BUTTON" VALUE="Answer"
ONCLICK="getanswer();">
</FORM>
</BODY>
</HTML>
```

This code takes two numbers from the textboxes, adds them together and pops up the result.

Result in browser:

A =

B =



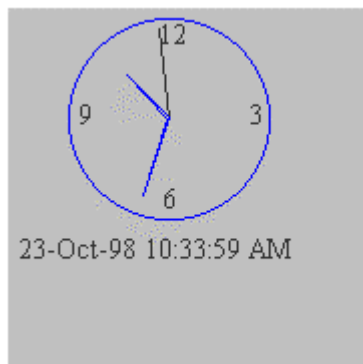
Note: Actual JavaScript programming is outside the scope of this course.

Chapter 11 - Exercise 1 - Inserting a Java Applet

In this exercise, you will insert a Java applet that displays the time, on an analog clock, in the browser.

Procedure

1. Create a new file with the basic elements. Type "JAVA Clock Example" into your TITLE element
2. Insert a set of APPLET tags inside the BODY: <APPLET></APPLET>
3. Add attributes to the APPLET element:
 - CODEBASE="../java"
 - CODE="clock2.class"
 - ID="CLOCK2"
 - HEIGHT="180"
 - WIDTH="180"
 - ALT="The Current Time Is"
4. Save the file as Jclock.htm in the C:\HTMLFILES\exercises directory.
5. Preview the file in different browsers. It should look similar to this:



Sample Code:

```
<HTML>
<HEAD><TITLE>JAVA Clock Example</TITLE>
</HEAD>
<BODY>
<APPLET CODE="Clock2.class" CODEBASE="../JAVA" ID="Clock2"
  ALT="The Current Time" WIDTH="180" HEIGHT="180"></APPLET>
</BODY>
</HTML>
```

Chapter 11 – Optional Exercise

In this exercise you will create a file as illustrated in the example, and test in both browsers:

Sample Code:

```
<HTML>
<HEAD>
<TITLE>JavaScript Example</TITLE>
<SCRIPT LANGUAGE="JAVASCRIPT">
<!-- hide JavaScript code from browsers that are not JavaScript enabled
function getanswer() {
    var x = parseInt(document.FORM1.FirstNum.value);
    var y = parseInt(document.FORM1.SecondNum.value);
    var sum = x + y;

    alert("The answer is: " + sum);
}
//end hiding of JavaScript code -->
</SCRIPT>
</HEAD>
<BODY>
<FORM NAME="FORM1">
<INPUT TYPE="TEXT" NAME="FirstNum"><BR>
<INPUT TYPE="TEXT" NAME="SecondNum"><BR>
<INPUT TYPE="BUTTON" VALUE="Answer" ONCLICK="getanswer();">
</FORM>
</BODY>
</HTML>
```


Review Questions

1. Who created the JAVA programming language?
2. What does the PARAM element do?
3. Why is the actual JAVAScript code placed within a comment element?

Summary

As a result of this chapter, you should be able to

- Describe JAVA.
- Add an APPLELET to a web page.
- Describe JAVAScript.
- Add JAVAScript to an HTML page.

12

ACTIVE X Objects, and VBScript Introduction

In this chapter you will learn about ActiveX, and add an object to a web page.

Objectives

Upon completing this section, you should be able to

1. Describe ActiveX.
2. Add an ActiveX to a web page.
3. Describe VBScript.
4. Add some VBScript to an HTML page.

Prerequisites

- Internet Explorer 3.0 or greater
- Navigator 3.0 or greater with Plug-in (For ActiveX support, Navigator does not support VBScript)

ActiveX Support

ActiveX controls/components can be added to your HTML pages using the OBJECT element. Like the APPLET element the PARAM element is a sub-element of OBJECT that controls the way the ActiveX control functions.

ActiveX controls are compiled programs. They have been derived from the OLE (Object Linking and Embedding) standard. There are a large number of controls currently available from Microsoft and other vendors.

The big advantage over JAVA Applets is that ActiveX controls are only downloaded once. The controls integrate themselves with the operating system, making their execution extremely fast, and their display characteristics sharp and crisp. This makes them extremely valuable on Intranets for applications such as graphing.

The big disadvantage of ActiveX controls is their lack of portability compared to JAVA Applets that will run on any platform.

ActiveX control creation is beyond the scope of this course and requires knowledge of C++ or Visual Basic programming.

<OBJECT>

The <OBJECT> tag has a number of attributes that define the object, its location, and how it is presented on the HTML page.

Class ID

ActiveX controls, which are binary files, are installed on a user's computer and registered in the system registry with a unique class ID. The CLASSID attribute of the <OBJECT> tag contains a URL that uniquely identifies the object. Referencing the class ID of the registered control in the <OBJECT> tag inserts it into the HTML page, and then the control displays itself.

ActiveX controls are identified by a URL that uses a class ID identifier. The format for this URL is:

```
"clsid:12345678-1234-1234-1234-123456789012"
```

This example shows the <OBJECT> tag for the Calendar control. The CLASSID is the only attribute that is required by the <OBJECT> tag.

```
<OBJECT classid="clsid:8E27C92B-1264-101C-8A2F-040224009C02">  
</OBJECT>
```

The class ID for an ActiveX control is placed in the registry when a control is installed. There are a number of tools you can use to retrieve this value.

- ® **Registry Editor (RegEdit)** This tool enables you to change settings in your system registry. You can also use it to view and copy the class ID for an object.
- ® **OLE Viewer** This utility is installed with the Win32 Software Development Kit (SDK). You can select the Copy HTML <OBJECT> Tag to Clipboard command from the Object menu to copy the tag and the class ID to the Clipboard.
- ® **ActiveX Control Insertion Device** This utility can also build the <OBJECT> tag for you and copy it to the Clipboard.

ID

The ID attribute in the <OBJECT> tag is synonymous with the NAME attribute for standard controls. You can use the ID to refer to the object from VBScript.

```
<OBJECT
  classid="clsid:8E27C92B-1264-101C-8A2F-
040224009C02"
  id=cldCalendar2>
</OBJECT>
```

CodeBase:

This attribute is a URL that points to a file containing the implementation of an object. (It is explained in detail in the next topic.)

Name:

This parameter is required when the object is on a form and needs to be included with other form fields when information is sent to the server.

Width, Height, Align, HSpace, and VSpace

These attributes affect the way an object is positioned and sized within an HTML page.

CodeBase

If the object specified by the CLASSID parameter is not available on the user's computer, the browser will use the CODEBASE parameter to determine where the object is located on the Internet server, and download the files required to render the object. As with all files that can be downloaded, you should provide the user with the option of downloading the object and an alternative to the object, if possible.

For example, the browser will try to locate an ActiveX control on the local computer by first searching the registry. If the control is not registered, the browser will download the object from the location specified in the CODEBASE parameter.

Specifying the CODEBASE URL

The CODEBASE parameter for the <OBJECT> tag contains a URL pointing to the implementation files for a given object. This URL is critical for downloading components, because it must specify all files necessary to implement a particular object.

This example downloads an ActiveX control by referencing the .ocx file:

```
<OBJECT
  ID="NewControl"
  CLASSID="clsid:123-1234-123-123-"
  CODEBASE="http://server/control.ocx">
</OBJECT>
```

HTML authors can set the CODEBASE attribute to point to one of three file types, which are typically supplied by the developer of the control.

- ® **Portable executable file** This is a single executable file, such as an .ocx file or DLL, that is downloaded, installed, and registered by the browser.
- ® **.CAB (cabinet) file** This file contains one or more files, all of which are downloaded together in a compressed cabinet.
- ® **.INF file** This file specifies various files that need to be downloaded and set up for the .ocx to run.

VBScript

VBScript is a programming language that allows scripting of events, objects and actions to create Internet applications. It uses two elements the <SCRIPT> <SCRIPT> and <!-- the comment - ->. The script element lets the browser know what type of script it is with its LANGUAGE attribute. The comment tags enclose the actual VBScript code so that browsers that don't support scripting will ignore the code.

Since the VBScript is embedded in the HTML page it is interpreted at the client when the page is loaded, and activated by an event like pressing a button. It can also be used in the creation of Active Server Pages, for developers using Microsoft's Internet Information Server as their web server platform.

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

<SCRIPT LANGUAGE="VBScript">
<!-- beginning of script
Sub add_2_num()
  a_str = document.Form1.Text1.value
  b_str = document.Form1.Text2.value
  a_int = CInt(a_str)
  b_int = CInt(b_str)
  c = 0
  c = a_int + b_int
  MsgBox "The answer is: " & c
End Sub
<!-- end of script --></SCRIPT>
<FORM NAME="Form1">
  <P> A = <INPUT TYPE="TEXT" NAME="Text1"><BR>
    B = <INPUT TYPE="TEXT" NAME="Text2"><BR>
    <INPUT TYPE="BUTTON" NAME="Button1" VALUE="A + B =" ONCLICK="add_2_num()">
  </P>
</FORM>
</BODY>
```

The above code takes two numbers from the textboxes, adds them together and pops up the result.



Actual VBScript programming is outside the scope of this course.

Note: VBScript is currently not supported in Navigator.

Chapter 12 – Exercise 1 – Adding an ActiveX Object

In this exercise you will add the Calendar Control to a page and view it in Microsoft's Internet Explorer.

Procedure

1. Using MS Internet Explorer browser - Open the webpage "ActiveX.htm" (located in the C:\HTMLfiles\ ActiveX directory). You may need to adjust your browser settings to a lower security level in order to get the page to open in the browser.
2. In Notepad, start a new document and add the basic tags.
3. Insert a set of OBJECT tags inside the BODY: <OBJECT></OBJECT>
4. Add attributes to the OBJECT element:
 - CLASSID= "CLSID:8E27C92B-1264-101C-8A2F-040224009C02"
(The Class ID value can be found using a tool such as RegEdit.)
 - HEIGHT="400"
 - WIDTH="400"
 - ALT="Graphical Calendar"
5. Save your file and view it in Internet Explorer.

NOTE: This code will only work on a system that has the Calendar Control in its registry. To create a page that will allow the control to download to a machine that doesn't have the control resident, you must also use the CODEBASE attribute where the value of CODEBASE is a URL, specifying where the object is located on the Internet server. (View the source of the file "activex.htm")

Chapter 12 – Optional Exercise

In this exercise you will create a file as illustrated in the example, and test in both browsers:

Sample Code:

```
<SCRIPT LANGUAGE="VBScript">
<!-- beginning of script
Sub add_2_num()
    a_str = document.Form1.Text1.value
    b_str = document.Form1.Text2.value
    a_int = CInt(a_str)
    b_int = CInt(b_str)
    c = 0
    c = a_int + b_int
    MsgBox "The answer is: " & c
End Sub
<!-- end of script -->
</SCRIPT>

<FORM NAME="Form1">
<P> A=<INPUT TYPE="TEXT" NAME="Text1"><BR>
B=<INPUT TYPE="TEXT" NAME="Text2"><BR>
<INPUT TYPE="BUTTON" NAME="Button1" VALUE="A + B="
ONCLICK="add_2_num()"> </P>
</FORM>
```

Review Questions

1. What element is used to support ActiveX controls?
2. Why should you specify the CodeBase attribute?

Summary

As a result of this chapter, you should be able to

- Describe ActiveX.
- Add an ActiveX object to a web page.
- Describe VBScript.
- Add some VBScript to an HTML page.

Special Elements

In this chapter you will learn how to create animated pages using a Meta element and how to use the Meta element to help index your website on the Internet. You will also learn how to use some additional elements that allow you to add background sound, scrolling text, and blinking text.

Objectives

Upon completing this section, you should be able to

1. Create animated pages using the Meta element.
2. Use the Meta element to add information relevant to Internet search engines and people viewing your source code.
3. Add background sound to HTML pages.
4. Add a Marquee.
5. Add Blinking text.

Prerequisites

- The requirements are described with each element.

Uses of the Meta Element

1. Client Pull & Page Animation

Client pull describes a scenario where the HTML page has a META element inside the HEAD element of the document, which instructs the browser to REFRESH the view of that page. Other attributes of the refresh instruction specify the time interval and the URL to be used. In this way, a series of pages can be presented without the user requesting the new pages to be loaded.

```
<META HTTP-EQUIV="Refresh"  
CONTENT="1;URL=../directory/filename.htm">
```

2. Description

You can use the Meta element to embed information in the Head of your page that the search engines will use to display a description of your site.

```
<META NAME="description" CONTENT="This is my company's web site.  
The site contains information on our products and services.">
```

3. Keywords

You can use the Meta element to embed information in the Head of your page that the search engines will use for keywords when people are searching for your site.

```
<META NAME="keywords" CONTENT="product name, service name,  
company name, country, province/state, industry, etc.">
```

4. Author

You can embed the author's name in a Meta element

```
<META NAME="author" CONTENT="Your Name">
```

5. Company

Similarly you can embed the company name.

```
<META NAME="company" CONTENT="Company Name">
```

6. Copyright

```
<META NAME="copyright" CONTENT="Copyright © 1997 Your Name or  
Company name. All rights reserved.">
```

Example:

```
<HTML>  
<HEAD>  
<TITLE>My HTML Document</TITLE>  
  
<META NAME="description" CONTENT="This is my company's web site.  
The site contains information on our products and services.">  
<META NAME="keywords" CONTENT="product name, service name,  
company name, country, province/state, industry, etc.">  
  
<META NAME="author" CONTENT="Your Name">  
<META NAME="company" CONTENT="Company Name">  
<META NAME="copyright" CONTENT="Copyright © 1997 Your Name or  
Company name. All rights reserved.">  
  
</HEAD>  
<BODY>  
  
</BODY>  
</HEAD>
```

Adding Sound to an HTML Page

When sound files are included in web pages the browser must first download the audio file and then start the player software; this can seem to take an extremely long time over some connections. Depending on the browser that the client is using either the Embed element or the BGSound element will be supported.

Embed

The <EMBED></EMBED> element is used to place audio in a web page. It is supported by Netscape Navigator, and some ActiveX extensions to Internet Explorer. The Embed element has the following attributes:

SRC: defines the URL of the sound file.

CONTROLS: You have your choice of several controls including console, small console, play button, etc.

AUTOSTART: when set to true the sound will start once the sound file is downloaded.

HIDDEN: when set to true it will hide the controls, default is false.

LOOP: defines how many times the sound file will play.

VOLUME: sets the default volume when playing first starts.

HEIGHT: in pixels depends on choice of controls.

WIDTH: in pixels depends on choice of controls.

```
<EMBED SRC=file:///C:/WINDOWS/MEDIA/Jungle%20Exit.wav  
WIDTH="145" HEIGHT="60" AUTOSTART="TRUE" VOLUME="100"  
CONTROLS="CONSOLE" HIDDEN="TRUE">
```

The following sound formats are supported: .WAV or .AU audio files, or .MID midi files.

BGSound

The <BGSOUND></BGSOUND> element is used to place background audio in a page, it is supported by Internet Explorer. BGSound has the following the following attributes:

SRC: defines the URL of the sound file.

LOOP: defines how many times the sound file will play.

DELAY: defines the delay between loops.

TITLE: the text that describes the sound.

```
<BGSOUND SRC=file:///C:/WINDOWS/MEDIA/Jungle%20Start.wav  
LOOP="2" TITLE="Jungle theme" DELAY="5">
```

The following sound formats are supported: .WAV or .AU audio files, or .MID midi files.

Animation Elements

Marquee

The `<MARQUEE></MARQUEE>` element defines text that is displayed in an animated region of the browser.

WIDTH: of the marquee in pixels.

HEIGHT: of the marquee in pixels.

LOOP: defines how many times the text will travel through the space.

BGCOLOR: defines the background color of the marquee.

BEHAVIOR: defines how the text should move in the marquee.

SCROLLDELAY: sets the number of milliseconds before refreshes.

```
<MARQUEE BEHAVIOR="Alternate" BGCOLOR="#FFFF80"  
LOOP="5">This Text is Animated!</MARQUEE>
```

The Marquee element is currently only supported by Internet Explorer, Navigator just displays the text between the `<MARQUEE>` and `</MARQUEE>` tags.

Blink

The `<BLINK></BLINK>` element is supported by Navigator. It will cause text to blink. You can combine it with most text formatting for the desired effect. The element has no attributes.

```
<BLINK>This Text will blink</BLINK>
```

A word of caution this element should be used very cautiously as it will become annoying to repeat visitors.

Chapter 13 – Exercise 1 – Adding Client Pull

In this exercise, you are going to modify a series of html pages so that each page contains a META element that causes the browser to load another page in a specific period of time.

Procedure

1. In Notepad open the file called c:\HTMLfiles\Special\h.htm
2. Insert a META tag inside the HEAD: <META>
3. Add attributes to the META element:
 - HTTP-EQUIV="REFRESH"
 - CONTENT="1;URL=ht.htm"

Note: Notice that the document h.htm points to ht.htm.

4. Save the file.
5. Insert a META tag in ht.htm that points to htm.htm and save the file.
6. Insert a META tag in htm.htm that points to html.htm and save the file.
7. Insert a META tag in html.htm that points to web.htm and save the file.
8. Preview the file h.htm.

Chapter 13 – Optional Exercise

In this exercise you can experiment with animation elements and their attribute settings.

Procedure

1. Create a new file to test MARQUEE and BLINK.
2. Add some of the other META tags to your page.
3. Add a MARQUEE element
4. Add a BLINK element
5. Make the text in both larger and colored
6. Save your file and view in both Navigator & Internet Explorer
7. Experiment with some of the other attributes of the MARQUEE element

Sample Code:

```
<HTML>
<HEAD>
<TITLE>Marquee and Blink</TITLE>
<META NAME="COMPANY"
CONTENT="XYZ Corporation web site with samples of Marquee and
Blink as animation.">
<META NAME="KEYWORDS" CONTENT="Text Animation, Marquee,
Blink">
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<P><MARQUEE BEHAVIOR="ALTERNATE" BGCOLOR="#80FF00"
LOOP="10" HEIGHT="40"><FONT
SIZE="+3" COLOR="#FF0000">This Text is
Animated!</FONT></MARQUEE>
</P>
<P><BLINK><FONT SIZE="+3" COLOR="#FF0000">This Text is
Animated!</FONT></BLINK>
</P>
</BODY>
</HTML>
```

Review Questions

1. Where do the Meta tags go in the HTML Document?
2. Name three uses for the Meta element information.
3. How can you add sound to an HTML page?
4. How do you set the attributes of the Blink element?

Summary

As a result of this chapter, you should be able to

- Create animated pages using the Meta element.
- Use the Meta element to add information to your pages that is relevant to Internet search engines and people viewing your source code.
- Add background sound to an HTML page.
- Add a Marquee.
- Add Blinking text.

Page Layout, and Design Considerations

This chapter will teach you to use HTML elements in an effective manner to control page layout. You will also be introduced to the different design considerations that you will have to make.

Objectives

Upon completing this section, you should be able to

1. Use Technical Design Considerations in planning an HTML page or project
2. Discuss Effective Page Layout
3. Plan a successful web site
4. Create a web site that is easily navigated

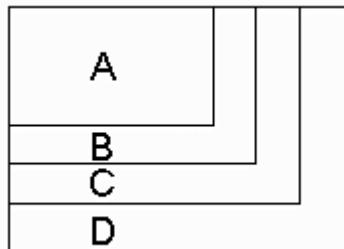
Prerequisites

- Multiple browsers to ensure that your page looks the same in different browsers.

Technical Design Considerations

Screen Resolution

Screen resolution is dependent on the hardware and the settings on the computer that is going to view your page. It is measured in pixels. When designing your pages you should make sure that they look good at the lowest resolution. The maximum **guaranteed** viewable area on a PC is 600 X 280. This takes into account that the browser is actually using some of the viewable space.



- A. 640 X 480
- B. 800 X 600
- C. 1024 X 768
- D. 1280 X 1024
- E. Others, Mac and Web TV

Color Depth

Another factor that is determined by the hardware of the surfer's computer, is the number of colors supported; the video card or the current video settings can limit this. On a PC, the video driver defaults to 16 colors and in many cases, older video cards only support 16 colors. Designers should develop their graphics with a 216-color palette or the Netscape supported palette, and all graphics should be viewed at 16 colors to ensure that they look good.

- 16 colors
- 256 colors
- 216 colors

Reducing the colors used in graphics has the additional benefit of reducing the file size, which allows faster downloading.

Document Size vs. Download Time

Probably the biggest technical design consideration is the overall size of the page versus the time it takes to load on the surfer's computer. Several things affect the speed at which a browser receives your page. The primary one is the speed of the user's connection to the Internet; usually controlled by the surfer's modem speed. This is another factor that is out of the control of the designer.

As a designer you should add up the following components to calculate the overall size of your web page:

- A. The size of the html page in bytes.
- B. The size of any graphics in bytes.
- C. The size of any embedded audio files in bytes.
- D. The size of any ActiveX objects or JAVA applets

Page Size = HTML page + Graphics + audio + other (ActiveX and/or Java)

14.4 Kbps represents Kilobits per second or 14400 bps or (14400/8) 1800 Bps (bytes per second). A 50 KB (Kilo Byte) graphics file will take (50000/1800) 27 seconds to download. At 28.8 Kbps it will take half that time 13-14 seconds. At 33.3 Kbps it will take approximately 12 seconds.

Given that modem speed is a big limitation for most surfers you should not overload your pages. You should give the surfer the option to go to pages that are heavy with graphics, sound or JAVA applets. Recognizing that many people surf with their graphics off you will also want to ensure that you effectively use the Alt Text attribute for graphics that you are using.

Technical Design Guidelines for Internet Applications

- Use plug-ins and applets sparingly
- Keep total page size less than 75KB (45 seconds @ 14.4)
- Let users choose to view graphics intensive pages
- Keep graphics less than 50KB usually
- Use interlaced graphics (don't use for backgrounds)

Page Loading – HTTP 1.0 Differences with HTTP 1.1

Many questions come up over how a page is loaded, and a lot of that can be answered by the way the protocol that transports the page from the server to the web client works.

1. The HTML page is retrieved. This is done through one connection to the server. The content of the page is then parsed, by the browser, for other elements that need to be downloaded like graphics, sound, ActiveX objects, or JAVA applets.
2. Separate connections to the server are created for the other elements in the page, and each element is downloaded individually. Each one of these connections would be considered a hit by the web server, and entered into the web server log.
3. Since the TCP/IP protocol can have several connections open at once between the server and the web client, more than one element is downloaded at a time. This is why the status bar at the bottom of the browser shows the percentage of the file downloaded. The status alternates between different sizes as the information for each connection reaches the browser.

Future web servers and web client software will support HTTP 1.1, which has several enhancements over HTTP 1.0. The major enhancement is that only one connection will be established for the page and all the elements that need to be downloaded. This will reduce the number of connections that must be opened and maintained, at the server and will decrease download time.

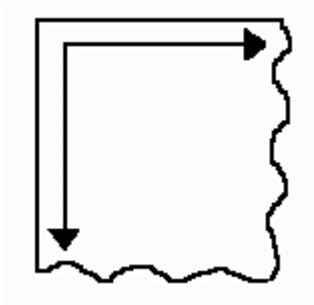
Browser Compatibility & Quirks

Web client software is written by a number of companies and as such there are a number of differences between the browsers in the way that they interpret standard HTML code.

One difference is the way in which the browsers draw tables. Internet Explorer will redisplay the table as new elements fill in the cells. Navigator 4 downloads all the elements before displaying the table and so does not perform multiple redraws as the table fills with data.

Page layout

An HTML page starts in the top left corner and grows to the right with no fixed boundary, and grows down with no fixed boundary. This means a page can be infinitely long and wide.



Keep in mind that people reading English are trained to start in the top left of a page and finish at the bottom right. It is a good idea to start the theme of the page in the top left or center at the top of the page for this reason. Page navigation controls should be located at the bottom right hand side of the page to make it easy to select the next page, or return to the previous page.

Page Layout Guidelines

1. In most cases your pages should not require the viewer to scroll to the right. The maximum width of a page should be 600 pixels in order to be displayed properly, regardless of the display resolution.
2. Recommended components of a home page:
 - i. E-mail address (i.e. `webmaster@whatever.com`)
 - ii. Mailing address
 - iii. Phone numbers (800 numbers plus local area code and number. Remember not everyone one in the world has access to your 800 number)
 - iv. Interactive component that encourages repeat visits by your target audience
 - v. Look, feel and message should reflect the organization
 - vi. Information that is relevant to the target audience
 - vii. Directions to what your target audience wants to know or do (PRE and/or Post Sale)
3. To control a page you may want to create a table with a maximum size of 600 pixels and contain the page inside the table.

4. When deciding to use a Background Image that has a distinct left side you should make sure it is 1280 pixels wide, because the browser tiles the background image, and anything smaller will appear again on the right of the page when viewed at high resolutions.
5. Use the DIV and CENTER element to control layout. (Div creates divisions within the document that do not have preformatted spacing associated with them, like paragraphs) You can control the alignment of objects with the alignment attribute of the DIV element.
6. To create attractive looking forms you should use a table to control the layout and alignment of the form elements.
7. Web pages should use white space at the borders. This makes the pages easier to look at and to read. Printed pages usually have white space at the left and right hand sides; as well, at the top and bottom of the page. This effect can be achieved by centering the contents of the page.
8. Frames are good for keeping people in your site. Careful consideration must be given to using Frames, as they require Frames-capable browsers. Always remember to provide valuable information in the NoFrames section of the Frameset.
9. You should be consistent in your layout across multiple pages, people become comfortable with the format, and begin to expect it.
10. Bigger is better. If things are too small they will strain the viewer's eyes and it will not be comfortable for them. General text areas should be 10 pt. or larger as this approximates a similar reading experience as a newspaper.
11. You should vary sizes for texture. When everything is the same size it becomes very boring.
12. Blinking text should only be used in situations where the information is changing frequently.
13. Be sure to use case properly. Mix upper and lower case as required or for effect. Documents done in all UPPER case are hard to read.

Site Design Factors and Criteria

When developing a web site from scratch you should first answer the questions related to why people will be visiting the site in the first place. Is it for pre or post sale support, to purchase something, or find out more about the company. Your web site should be part of your business plan or marketing plan and there should be established goals and objectives for what it is to accomplish. Once you know what problem you are trying to solve with the web site you can then start your design.

In general a site should be intuitive for the person visiting it. It should be very easy for them to contact your organization at any time and they should be able to locate the information they are looking for in three to five mouse clicks.

1. The Home page should be simple and fast.
2. The Home page should answer some basic questions for the surfer:
 - i. Is this the organization I was looking for?
 - ii. Do they have what I am looking for?
 - iii. How can I contact them?
 - iv. How do I get what I want now?

Site Layout and Navigation

1. Navigation controls should be located across the top or the right hand side of the page. This will locate your navigation controls close to the browser's controls for movement and up and down scrolling.
2. As previously mentioned you should only have 3 to 5 mouse clicks to move from start to destination. People do not like to hunt for an answer.
3. Create sub sites to simplify navigation, and for marketability. E.g. education.novell.com, insight.netscape.com

Review Questions

1. What is the maximum guaranteed viewable area on a PC?
2. Give two reasons why you would want to reduce colors in graphics?
3. What components do you add together to find your page size?
4. How many Bytes per second can a 14.4 modem receive?
5. Name three methods of controlling the positioning of text and objects on a page?
6. What must you be check for when using a background image?
7. How many mouse clicks are reasonable for surfers to make to get to the information they are after?

Summary

As a result of this chapter, you should be able to

- Use Technical Design Considerations in planning an HTML page or project
- Discuss Effective Page Layout
- Plan a successful web site
- Create a web site that is easily navigated

Cascading Style Sheets

This chapter will help you make use of a powerful feature in HTML page creation called Cascading Style Sheets.

Objectives

Upon completing this section, you should be able to

1. Describe how to use a Cascading Style Sheet.
2. Add an Inline Style.
3. Add an Embedded Style Sheet.
4. Add Linked Style Sheets.
5. Use the DIV and SPAN Elements to add style.
6. Define and use a Class.
7. Define and use an ID.

Prerequisites

Cascading Style Sheets are currently supported by:

- Netscape Navigator 4.0 or greater
- Microsoft Internet Explorer 3.0 or greater.

Introduction to Cascading Style Sheets

Cascading style sheets are a recent addition to HTML pages. Web page designers have long been frustrated at the lack of control they have, over how a page will appear. Cascading Style Sheets are a way of controlling the way the browser displays the page. For the experienced designer, Cascading Style Sheets will be a welcome feature that will save a lot of time normally spent on repetitive style formatting commands. You can specify a particular font size in points or pixels and control the spacing and indentation, etc. Cascading Style Sheets will give you much more control over the text on the page.

There are three ways to add Style information to you web page:

1. Inline styles
2. Embedded style sheets
3. Linked style sheets

We will cover the concepts behind all three.

Microsoft Internet Explorer (3.0 and above) and Netscape Communicator 4.0 provide the most extensive support for cascading style sheets at this time.

The specifications for the evolving CSS standard can be found at the W3C Consortium's Web page at <http://www.w3.org/>. You will need to use the CSS standard section of the site or some other CSS reference to see all of the details for writing style commands. This chapter will focus on showing you how to apply style commands to your documents.

Inline Styles

The first method of adding Style to a web page is to use the STYLE attribute of a selected element. To use an inline style, you add a STYLE attribute to a specific instance of an element, using the following syntax:

```
<ELEMENT STYLE="property:value; property:value; ..."></ELEMENT>
```

For example: `<B STYLE="color:navy;">In the navy.` This text would appear bold and navy in a browser.

An inline style may be applied to any HTML element, from `<A>` to `<VAR>`, and modifies only the specific instance (occurrence in the document) of the element to which you apply it.

Sample Code:

```
<HTML>
<HEAD>
<TITLE>Style Samples</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H2 STYLE="COLOR:RED">This Heading has Style!</H2>
<H2>This Heading has Style!</H2>
<H2>Introduction</H2>
<H2>Sales and Marketing</H2>
<P>This is normal text.</P>
<H2>Operations</H2>
<H2>Summary</H2>
</BODY>
</HTML>
```

When viewed in a browser you can see that in this file the STYLE command only affects the element that it is applied to.

Inline Styles are used rarely because they are as much work as using the `` tag. They are sometimes used to override styles from embedded style sheets or from a linked cascading style sheet; this will be demonstrated later.

Embedded Style Sheets

To use an embedded style sheet, you define a style block (delimited by the `<STYLE>` and `</STYLE>` tags), which should be placed in the `<HEAD>` section of the document. This block consists of a set of style rules, where each rule defines a style for an HTML element or group of elements.

```
<HEAD>
<TITLE> Embedded Style Sheet Example </TITLE>
<STYLE TYPE="text/css">
</STYLE>
<HEAD>
<BODY>
```

A required attribute of the Style element is `TYPE` and it should be set to `text/css`. A disadvantage of Embedded Style sheets is that the style commands only apply to the document they are embedded in.

Some browsers are not CSS-compliant and will display the contents of the `STYLE` element in the document window. For this reason it is a good idea to surround an embedded style sheet with an HTML comment. This will cause the content to be ignored by non-CSS browsers, but still interpreted by CSS compliant browsers.

- Type `<!--` just after the `STYLE` start-tag, but before the style sheet information.
- Type `-->` just before the `STYLE` end-tag, but after the style sheet information.

A style rule has two parts:

A selector that identifies an HTML element or group of elements

A declaration of the style properties to be applied to that selector

The generic syntax for a style rule is as follows:

```
selector {property:value; property:value; ... }
```

Each style rule must start with a selector or group of selectors, followed by an open brace (`{`), followed by a set of declarations. Each of these declarations ends with a semi-colon and each property is separated from its value(s) by a colon (`:`), ending with a close brace (`}`).

Add the following STYLE element to the previous code sample:

```
<STYLE TYPE="text/css"><!--  
H2 {color: blue;}  
--></STYLE>
```

Using the Embedded Style Sheet will make all of the <H2> elements in the document display in blue, except for the <H2> that still contains the inline style command. This illustrates that the inline style command is a higher priority and takes precedence over the embedded commands if they are in conflict.

Linked Style sheets

The easiest method for adding style to multiple HTML documents is to use Linked Style Sheets. A Linked Style Sheet is a separate file that contains all of the style information.

This is an example of a Cascading Style Sheet.

```
H2 {
  color: fuchsia;
  text-decoration: line-through;
}
```

To link the style sheet to the HTML page you use a LINK element in the head of your document. The required attributes are HREF="URL" and REL="STYLESHEET"; both the TITLE and TYPE are optional. But TITLE must match the Title: property in the Style Sheet), and TYPE should be set to "text/css".

To create a linked style sheet you can use Notepad and save the file with the file extension .css E.g. h2_styles.css

To have the style sheet work with your HTML file you need to link the file using a LINK element.

Sample Code added to previous file:

```
<HEAD>
<TITLE>Style Samples</TITLE>
<LINK REL="STYLESHEET" HREF="h2_styles.css">
<STYLE TYPE="text/css"><!--
H2 {
  color: blue;
}
--></STYLE>
</HEAD>
```

View the file in your browsers and you will see that the colors are still controlled by the higher priority inline and embedded style commands but because the text decoration command does not conflict with anything, it is applied to all of the <H2> elements in the document.

Classes

If you expect to have formatting variations for different instances of a single element, or you would like to have different elements share the same format, using a class name as a selector is a good approach. This is often referred to as "sub-classing" an element. CLASS is an HTML attribute that has no display characteristics and that you can apply to any element, like this:

```
<H2 CLASS="sales">Styles applied to the sales section</H2>
```

The style rule for sales could be declared as follows:

```
<STYLE> <!--  
.sales { color:lime;  
        text-decoration:none;  
}  
--> </STYLE>
```

Note: that the selector begins with a period (.), which is the required syntax for class names as selectors.

A class is simply a group of elements, possibly scattered throughout a document, all of which have the same value for their CLASS attribute.

To apply a class to a group of elements, set the CLASS attribute for all that apply:

```
<H2 CLASS="sales"> Sales and Marketing</H2>  
<P CLASS="sales">This is normal text.</P>
```

In the browser, notice how the styles associated with the CLASS value are applied to the tagged elements and also notice that they override the other conflicting style commands.

IDs

Like CLASS, ID is an HTML attribute that does not affect the display of an element and can be applied to any element. In general, while class names are usually given to groups of element instances sharing some common function or format (relative importance, context, and so on), ID is used to identify one specific instance of an element. Style rules selected by the ID attribute are declared using the pound sign (#) to precede the selector, as follows:

```
<STYLE> <!--  
#mainhead {color:yellow; text-decoration:italic;}  
--> </STYLE>
```

Note: While many elements can have the same CLASS attribute value, only one element in any document can have a particular ID attribute value.

IDs are intended to be unique identifiers for elements in a document. Cascading style sheets let you associate an ID attribute with a set of style properties and values, so that you can format a particular element instance in a certain way.

Note: Class and ID attribute values must start with a letter and can contain only letters, numbers, and '-' (hyphen).

DIV and SPAN

Just as CLASS and ID appear to have little use beyond setting styles (and scripting), the two HTML elements <DIV> and are almost exclusively used as containers for CSS properties. <DIV> and , like CLASS and ID -- but unlike other HTML elements -- have no inherent display characteristics, with one exception each. <DIV> defines a block consisting of text and HTML tags, and separates this block from surrounding content by line breaks, while is an inline element which "flows" in with surrounding content.

```
<STYLE>
<!--
DIV { background-color: black; color:red; font-weight:bold; }
SPAN { background-color: royalblue; color:white; }
-->
</STYLE>
```

```
<P>Some text about to run into a big DIV tag</P>
<DIV>I am a DIV</DIV>
<P>and narrowly escape.</P>
<P>Some text about to flow seamlessly into a SPAN tag <SPAN>I am a
SPAN</SPAN> and make a smooth getaway.</P>
```

Some text about to run into a big DIV tag

I am a DIV

and narrowly escape.

Some text about to flow seamlessly into a SPAN tag I am a SPAN and make a smooth getaway.

Cascading and Inheritance

The style sheets we are discussing are called Cascading Style Sheets. Simply stated, "cascading" in CSS specifies how an element instance may properly be affected by styles of different types (inline style, embedded style sheet, linked style sheet, imported style sheet) and selected in a number of ways (element tag, class, ID). The logic is simple: CSS cascades from general to specific, and from top to bottom.

```
<HEAD>
<STYLE>
<!--
P { margin-left:20px; }
.clsCode { font-family:"Comic Sans MS"; font-size:10pt; color:navy;}
#idP1 { text-align:left; font-weight:bold; }
-->
</STYLE>
</HEAD>

<BODY>
<P ID="idP1" CLASS="clsCode">Multiple styles, no conflicts.</P>
</BODY>
```

idP1's formatting is affected by the style rules for <BODY>, <P>, clsCode, and idP1. These potentially conflicting styles are resolved through the laws of cascading and inheritance.

This means that first, the <BODY>-selected style (background-color) is applied, followed by the <P>-selected style, clsCode, and finally, idP1, with each style taking precedence over the previous one. If we had an inline style defined as well, it would have been applied last, overriding all others. In the cascade, for the same element, a rule with ID as the selector takes precedence over a rule with CLASS as the selector. CLASS, in turn, takes precedence over the HTML element name as selector. This is referred to as "specificity" of the selector. The order of specificity, from greatest to least, is:

1. inline styles
2. ID
3. CLASS
4. HTML element

There were no conflicting style assignments -- for example, the background color for the paragraph was set only in the <BODY> rule, and the font size was set only in the clsCode style rule — so the styles "trickled down" to the next selector unaltered.

In the case of conflicting assignments with identical selectors, a style selected by a selector of greater specificity takes precedence. For example, let's apply a color value (white) to the #idP1 declaration. This will "conflict" with the color value specified in the .clsCode specification. Since #idP1 has greater specificity, its color (white) will win out over the previously set color (navy).

Chapter 15 – Exercise 1 – Inline Styles

In this exercise, you will add an Inline Style to a document.

Procedure

1. Open C:\HTMLfiles\css\Doc1.htm.
2. Add a STYLE attribute to the <H1> element
3. Set the value of the STYLE attribute: type **"color: yellow"**
4. Save and preview your document in Internet Explorer or Navigator v4.

Chapter 15 – Exercise 2 – Embedded Styles

In this exercise, you will add Embedded Styles to a document.

Procedure

1. Open C:\HTMLfiles\css\Doc2.htm.
2. Insert a STYLE element in the head of your document. Set the TYPE attribute of the STLYE element to **“text/css”**
3. Inside the STYLE tags type:

```
<!--  
H2 {color: green;}  
P {color: red; font-size: 12pt; font-family: Arial;}  
-->
```

4. Save and preview your document in Internet Explorer or Navigator v4.

Sample Code:

```
<HEAD>  
<TITLE>Embedded Style Sheet</TITLE>  
<STYLE TYPE="text/css">  
<!--  
H2 {  
  color: green;  
}  
  
P {  
  color: red;  
  font-size: 12pt;  
  font-family: Arial;  
}  
-->  
</STYLE>  
  
</HEAD>
```

Chapter 15 - Exercise 3 – Linking a Cascading Style Sheet

In this exercise, you will create and link a style sheet to a document.

Procedure

1. Create a new file in Notepad
2. Type the following style commands:

```
.red {color: red;}  
.blue {color: blue;}
```
3. Save the document as stylesht.css in the C:\htmlfiles\css directory.
4. Open the file C:\htmlfiles\css\doc3.htm
5. Insert SPAN element tags around the word “red” E.g.
red
6. Set the CLASS attribute of the SPAN element to “red” E.g. red
7. Insert a SPAN element around the word “blue” and set the CLASS attribute to “blue”
8. Insert a LINK element in the HEAD element and set the following attributes:
 - REL=“stylesheet”
 - TYPE=“text/css”
 - HREF=“stylesht.css”
9. Save and preview your document in Internet Explorer or Navigator v4. You have successfully created an external style sheet that uses classes and linked it to your document that uses the SPAN element to apply those classes.


Chapter 15 - Exercise 4 – Editing the Style Sheet

In this exercise, you will edit the style sheet you created in Exercise 3.

Procedure

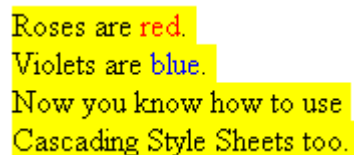
1. Open the file “stylesht.css”
2. Add the style commands:
 - `#Border {border-color:black; border-style:solid; border-width:thin; background-color:yellow;}`
3. Save the file with the changes
4. Open the file C:\htmlfiles\css\doc3.htm
5. Change the PARAGRAPH tags to DIV tags
6. Set the ID attribute of the DIV to “Border”
7. Save and preview your document in Internet Explorer and Navigator v4. You have successfully edited an external style sheet that uses IDs and linked it to your document that uses the DIV element to leverage those IDs. You will also notice that the support for the border style commands is different in Navigator and IE as is the way that the 2 browsers apply the background to the DIV element.

Microsoft Internet Explorer 4.x



Roses are red.
Violets are blue.
Now you know how to use
Cascading Style Sheets too.

Netscape Navigator 4.x



Roses are red.
Violets are blue.
Now you know how to use
Cascading Style Sheets too.

Optional Exercises

Experiment with the additional .css files in the sample files directories.

Review Questions

1. What are three ways to add style information to your web pages?
2. Give two disadvantages of Embedded Style Sheets.
3. How is a Style Sheet linked to an HTML document?
4. Describe two applications for Style Sheets.

Summary

As a result of this chapter, you should be able to

- Describe how to use a Cascading Style Sheet.
- Add an Inline Style.
- Add an Embedded Style Sheet.
- Add Linked Style Sheets.
- Define and use a Class.
- Define and use an ID.

Index

A

ActiveX 125
Address <ADDRESS>
 </ADDRESS> 34
Alignment 32
Alignment, <CENTER> 32
Alignment, <DIV> 32
Alignment, <TABLE> 32
Anchor, <A> 55
Applet, <APPLET> </APPLET> .. 116
Area Element <AREA> 60

B

Base, <BASE> 78
Big <BIG> </BIG> 33
Blockquote <BLOCKQUOTE>
 </BLOCKQUOTE> 34
BMP 49
BODY <BODY> Element..... 12
BODY Element, Text Attribute..... 12
Body, <BODY> </BODY> 4, 5
Bold 31
Bookmarks 58
Break, Line
 23
Browsers 145

C

Caption Element <CAPTION>..... 67
CENTER, see Alignment
 <CENTER></CENTER> 32
CGI..... 96
Citation <CITE> </CITE> 31
Class ID, ActiveX 125
Classes 157
Color Depth 143
Comments..... 118, 128
CSS – Cascading Style Sheets... 152
CSS, Cascading and Inheritance 160
CSS, Classes 157
CSS, DIV and SPAN 159
CSS, Embedded Style Sheets 154
CSS, IDs 158
CSS, Inline Styles 153
CSS, Linked Style sheets..... 156

D

DD <DD> Definition Descriptions,
 See Lists 42
DIV 159
DL <DL> Definition List, See Lists 42
Document Properties 11
Document Properties, Color Codes
 11
Document Size 144
Download Time..... 144
DT <DT> Definition Term, See Lists
 42

E

Element 3
E-mail 57
Embedded Style Sheets 154
Emphasis 31

F

Font 31
Form, attributes 88
Form, Elements 89
Forms, <FORM> </FORM> 87
Forms, Check boxes..... 91
Forms, File Upload 92
Forms, Image Submit button 93
Forms, Option button..... 91
Forms, Password boxes 90
Forms, Push button 92
Forms, Reset button 93
Forms, Select 95
Forms, Select options..... 95
Forms, Submit button 93
Forms, Text area 94
Frames 74
Frames, <FRAME> </FRAME> 78
Frames, <FRAMESET>
 </FRAMESET> 76
Frames, <NOFRAMES>
 </NOFRAMES> 79
Frames, Architecture 75
Frames, Creating..... 76
Frames, Other Attributes 77

Index

Frames, Other Attributes, BORDER	77	Image Maps, <AREA>	60
Frames, Other Attributes, BORDERCOLOR	77	Image Maps, <MAP> </MAP>	60
Frames, Other Attributes, FRAMEBORDER	77	Image Maps, Client-side	60, 124
Frames, Other Attributes, FRAMESPACING	77	Image Maps, Server-side	59
Frames, Special Targets	82	Images, 	50
Frames, Targets	81	Images, Alignment	50
FTP	56	Images, Alternate Text<ALT>	50
G		Images, Border Attribute	50
GIF	48	Images, Height Attribute	50
GIF, Animated	48	Images, HSPACE	50
GIF, Interlacing	48	Images, SRC attribute	50
GIF, Transparency	48	Images, VSPACE	50
Gopher	57	Images, Width Attribute	50
H		Inline Styles	153
Head, <BASE> </BASE>	5	INPUT <INPUT></INPUT>	89
Head, <HEAD> </HEAD>	4, 5	ISINDEX <ISINDEX></ISINDEX> ..	5
Head, <ISINDEX> </ISINDEX>	5	Italic <i> </i>	31
Head, <LINK> </LINK>	5	J	
Head, <META> </META>	5	Java	116
Head, <SCRIPT> </SCRIPT>	5	Java, <PARAM> </PARAM>	116
Head, <STYLE> </STYLE>	5	Java, Attributes<APPLET> </APPLET>	116
Head, <TITLE> </TITLE>	5	Javascript	118
Headings, <Hx> </Hx>	21	JPEG	49
Horizontal Rule, <HR>	24	JPG	49
HTML Tables	69	L	
HTML Tip, BlockQuotes	34	Linear Media	2
HTML Tip, Forms	88	Link Colors <LINK> <VLINK> <ALINK>	13
HTML Tip, Image Backgrounds	14	Link Element <A HREF=	55
HTML Tip, Images	50	Linked Style Sheets	156
HTML Tip, Links	58	Links, External	55
HTML Tip, Multiple Spaces	22	Links, Internal	55, 58
HTML, <HTML> </HTML>	4	Links, Local	55
HTTP	56	List Item 	40
HTTP Specification	145	Lists, Definition List <DL> </DL> ..	42
Hypermedia	2	Lists, Nesting	42
hypertext	2	Lists, Ordered List 	41
I		Lists, Unordered List ..	40
IDs	158	M	
Image Backgrounds	13	mailto	57
Image Maps	59	Marquee, <MARQUEE> </MARQUEE>	137
		Meta	134

Index

N		Sound, <EMBED> </EMBED> ...	136
NBSP	22	SPAN, 	159
News	56	Special Characters	31, 32
NOFRAMES <NOFRAMES>	79	Special Characters, <BASEFONT></BASEFONT>..	34
O		Special Characters, Code Element <CODE></CODE>	33
Option Button/Radio Button, Forms	91	Special Characters, Definition Element <DFN></DFN>	33
Ordered List, Lists 	41	Special Characters, Keyboard Text <KBD></KBD>	33
P		Special Characters, Sample Element Output <SAMP></SAMP>	33
Page Layout	146	Special Characters, Variable Element <VAR></VAR>	33
Page Layout Guidelines	146	Strike <STRIKE> </STRIKE>	33
Paragraph, <P> </P>	22	Strong .	31
PARAM <PARAM> </PARAM>	116	Style, <STYLE></STYLE>	5
PCX	49	Subscript 	33
PNG	49	Superscript 	33
PRE, <PRE></PRE>	31	Symbols	32
Preformatted Text <PRE> </PRE>	31	T	
Previewing Your Work	15	Table Data <TD></TD>	65
Using Microsoft Internet Explorer	16	Tables <TABLE></TABLE>	65
Using Netscape Navigator	15	Tables, Attributes	66, 68
S		Tables, Caption <CAPTION> </CAPTION>	67
Screen Resolution	143	Tables, Header <TH> </TH>	67
Script <SCRIPT> </SCRIPT>	128	Teletype <TT> </TT>	31
Script, <SCRIPT> </SCRIPT>	118	Telnet	57
Section 3.2		Text Color 	31
Intranet Application Development Architecture	97	Text Size 	31
Section 3.2.a		Textarea <TEXTAREA></TEXTAREA>	94
Simple CGI Program	97	TIFF	49
Section 3.2.b		Title <TITLE></TITLE>	5
Three-Tier Application Development	98	U	
Section 3.2.c		Underline <U> </U>	31
Four-Tier Web Application Development	99	Unordered List 	40
Select, FORMS <SELECT></SELECT>	95	URL	55, 56
Site Design Factors	148	V	
Site Layout and Navigation	148	value	3
Small <SMALL> </SMALL>	33	VBScript	128
Sound	136		
Sound, <BGSOUND>	136		

Toolbox Café

Web Hosting

Online Marketing

Quality Web Hosting
www.olm.net
1-877-265-6638

Access

Connect America

International Dial Up Services
www.connectamerica.com
909-778-3640

FlashNet Marketing, Inc.

Internet access products and services available for resale and promotion
www.flashopportunity.com
1-877-225-5364 x480

Internet Shows

NetTalk Live

Cool! Online and on cable!
www.nettalklive.com

Newsletters

Tricks and Trinkets

A very cool news letter!
www.tricksandtrinkets.com

E-commerce

E-Commerce Exchange

The online entrepreneur's choice for e-commerce since 1989
www.ecx.com
1-800-748-6318

InfiniteHorizon.com

Easy E-commerce setup, JavaScript libraries, cgi-bin capability, web based e-mail, auto responders, stats, 24 hour technical support and more.

www.virtualis.com/vr/ebabin/
562.433.4891

Associations

Webgrrls International

The women's tech knowledge connection
www.webgrrls.com

The HTML Writers Guide

International organization of Web authors
www.hwg.org

Training

XNU.COM

Web development training
www.xnu.com
info@xnu.com
1-877-644-3444

Software

32bit.com

Easiest software download on the Internet
www.32bit.com

The Web Developer's Virtual Library

The illustrated Encyclopedia of Web Technology
www.WDVL.com

JavaScripts, HTML, Web design & Promotion

Resources for development using JavaScript, HTML, and CGI.
www.algonet.se/~tools/index.htm

Free Site Tools

Free Website and Computer Resources – Tons of Webmaster Stuff
<http://www.freesitetools.com/>

Let THEM All Know YOU! ONLY **\$110.00** /YEAR
IN THIS TOOLBOX CAFE (PRINT VERSION)  **CLICK HERE**
TRAININGTOOLS.COM INFO@TRAININGTOOLS.COM



Make money while you sleep?

Start making more money on the Internet Today – Increase your business 30 to 100%.

TrainingTools.com special: Save \$195 setup fee!

800.274.7638



E-COMMERCE EXCHANGE

www.ecx.com