



Web Basics – HTML

Copyright © 2011 iGATE Corporation. All rights reserved. No part of this publication shall be reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior written permission of iGATE Corporation.

iGATE Corporation considers information included in this document to be Confidential and Proprietary.

Document History



Date	Course Version No.	Software Version No.	Developer / SME	Change Record Remarks
1-Oct-09	1.0	1.0	Kumar Bramhadande	Initial Document
May 2011	2.0		Karthik M	Integration Refinements
20-May-2013	3.0		Vinay Gupta	Revamped according to new curriculum



Keep this as a hidden slide.

Note to coordinators: Not to be printed for the class book.

Course Goals and Non Goals



➤ Course Goals

- Understand Static web page creation.
- Create Static web pages.



➤ Course Non Goals

- DHTML (not covered in this course).



+ 3 +

Pre-requisites



- **None**

+ 4 +



Intended Audience



- Novice User
- Developer



- 5 -



Day Wise Schedule



➤ Day 1

- Lesson 1: Introduction to Internet
- Lesson 2: HTML Basics
- Lesson 3: Creating Tables
- Lesson 4: Working with Lists
- Lesson 5: Working with Links
- Lesson 6: Image Handling
- Lesson 7: HTML Forms
- Lesson 8: Working with Frames
- Lesson 9: XHTML
- Lesson 10: Cascading Style Sheets



- 6 -

Table of Contents



➤ Lesson 1: Introduction to Internet

- Introduction to Internet
- TCP/IP Protocol
- World Wide Web
- Web Servers
- Web Browsers
- Static and Dynamic Web Pages

➤ Lesson 2: HTML Basics

- HTML Introduction
- Adding Physical/Logical Character Effects
- Managing Document Spacing

- 7 -



Table of Contents



➤ Lesson 3: Creating Tables

- Attributes of a Table
- Table Headers
- Table Data
- Table Formatting
- Control Table Borders
- Grouping of Columns

➤ Lesson 4: Working with List

- Numbered List
- Bulleted List
- Directory List
- Glossary List

- 8 -



Table of Contents



➤ Lesson 5: Working with Links

- How does a hyperlink work?
- Create Links to Web documents
- Links to E-Mail
- How to add CC n BCC link
- Hyperlinks for Lists & Table data
- Providing target for a hyperlink
- Relative Path



+ 9 +

Table of Contents



- **Lesson 6: Image Handling**
 - Attributes of an Inline Image
 - Text & Image Alignment
 - Use of Image as a Hyperlink

- **Lesson 7: HTML Form for User input**
 - Data Submission using a Form
 - Types of Form Fields

- 10 -



Table of Contents



➤ Lesson 8: Working with Frames

- Frame Basics
- Frameset Tag Attributes
- Frame Tag Attributes
- Nested Frames
- Noframes Tag

➤ Lesson 9: XHTML

- Work with XHTML 1.0
- Understand DTD's for XHTML
- Understand XHTML Namespaces
- Validating XHTML with W3C Standards

- 11 -



Table of Contents



➤ Lesson 10: Cascading Style Sheet

- Cascading Style Sheet Basics
- Three primary ways to use style sheets:
 - Inline
 - Embedded
 - Linked
- Advanced Style Sheet Features

- 12 -



References



➤ Books

- HTML Unleashed
- How To HTML 4.0
- Html Pocket Reference



➤ Links

- <http://imboldc.ucc.ie/~pflynn//books/htmlcard.html>
- <http://www.w3.org/MarkUp/>
- <http://www.htmlhelp.com/reference/html40/>

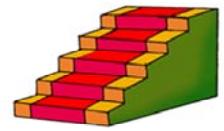
- 13 -



Next Step Courses (if applicable)



- Working with any Scripting Language
- Working with Dynamic HTML



- 14 -

Other Parallel Technology Areas



- **None**

- 15 -





Web Basics - HTML

Lesson 1. Introduction to the Internet



Lesson Objectives



- After completing this module you will be able to:
- Understand the history of Internet.
 - Understand Web terminology.
 - Understand IP addresses
 - TCP/IP Protocol
 - Domain Name System
 - HTTP Protocol
 - Servers – Web Servers
 - Web Browsers
 - Working of WWW
 - HTML – Static and Dynamic Web Pages



- 2 -

1.1: Internet

What is Internet?



➤ Internet:

- ‘Network of networks’ or “world’s largest network”.
- A concept, like the economy.
- Collection of inter-networked regional networks.
- Not owned by anyone.
- Based on TCP/IP.

- 3 -



1.2: History of Internet

History of Internet

 iGATE
ITOPS for Business Outcomes

- Need to share information.
- Advanced Research Projects Agency (ARPA)
 - ARPANET
 - Comprised individual packet switching computers interconnected by leased lines.

- 4 -



History of The Internet

Internet was established more than 25 years ago to meet research needs of the U. S. defense industry. However, it has grown into a huge global network serving universities, academic researchers, government agencies, and commercial interests, both in the United States and in more than 100 other countries. No one person runs the Internet, and no single organization pays all the costs; there is no Internet Corporation.

It exists as a result of the cooperation from people all over the world, who work in various types of organizational and computing environments. Internet never closes down, mostly because of its decentralized structure. Today, it is reliable and predictable. Individual servers may close for upgrades or hardware replacement, but the network is always available without interruption.

In 1969, Advanced Research Projects Agency (ARPA) of the U. S. Department of defense established ARPAnet, an experimental four-computer network, so that research-scientists could communicate among themselves. By 1971, ARPAnet comprised almost two dozen sites. By 1974, that number grew to 62, and by 1981, it comprised more than 200 sites.

As more and more computers using different operating systems were connected, the need for a *common communications protocol* became apparent. Theory required that any computer on the network should be able to talk to any other computer, as a peer.

1.3: Internet Basic Definition
Internet – Basic Definitions



- **Internet Service Provider (ISP)**
- **Network Information Center (NIC)**
- **Internet Address**
- **Internet Domain Name**
- **Routing**
- **Gateways**
- **Protocols**
 - **TCP/IP**
 - **HTTP**

- 5 -



Internet Service Provider (ISP): Internet Service Providers (ISPs) are companies that enable user to connect to the Internet . They offer an account on their systems and access to the Internet.

Network Information Center (NIC): Network Information Center (NIC) assigns and regulates IP addresses on the Internet. You can get one directly from the NIC, or you can ask your ISP to secure an IP address on your behalf.

Internet Address: TCP/IP requires each host on a TCP/IP network have their own unique IP address.

Internet Domain Name: A domain name maps or translates the actual numeric IP address used for your Web server into an easy-to-remember alphanumeric name. Domain refers to a collection of network host computers, known by the same name. Your domain name should reflect your organization or corporation, for example, .com , .edu , .gov , .int , .mil , .net, and so on.

Routing: Process of getting your data from point A to point B.

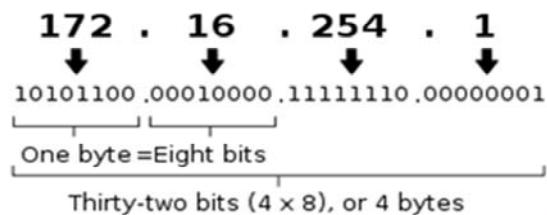
Gateway: Forwards datagrams to a destination if it knows where the destination is.

1.4: Internet Address
Internet Address



- Every device (eg: computer, printer) that participates in a computer network is assigned a numeric label called as Internet Protocol address (IP address).
- The designers of the Internet Protocol defined an IP address as a 32-bit number and this system is known as Internet Protocol Version 4 (IPv4).
- IP addresses are binary numbers, but they are usually stored in text files and displayed in human-readable notations, such as 172.16.254.1

An IPv4 address (dotted-decimal notation)



- 6 -



1.4: Internet Address

Internet Address (Contd...)



➤ Internet Address:

- 32-bit address, in the form x.x.x.x.

➤ Internet address classification

- Class A N.H.H.H Used for very large networks
- Class B N.N.H.H Used for medium sized networks
- Class C N.N.N.H Used for smaller networks
- Class D Multicast Address
- Class E Reserved for future use.

- 7 -



N: Network (Assigned by the NIC)

H: Host (Assigned by the local administrator)

1.4: Internet Address

Internet Address (Contd...)



➤ Addresses that start with a value between:

- 1 and 126 Class A (First bit value is 0)
- 128 and 191 Class B (First two bit values are 10)
- 192 and 223 Class C (First three bit values are 110)
- 224 and 239 Class D (First four bit values are 1110)
- 240 and 255 Class E (First five bit values are 11110)

- 8 -



1.5: TCP/IP Basics

TCP/IP Basics

➤ **Transmission Control Protocol (TCP):**

- Connection-oriented transport layer protocol.
- Sets up a connection between the sender and receiver.
- Uses the services of IP to send and receive data.
- Re-orders received information.

- 9 -



TCP ensures that data arrives and that it arrives in the correct order. It reorders information that is received out of order and requests the information that is not received is to be resent.

Internet uses TCP/IP to link computers. TCP/IP stands for Transmission Control Protocol/Internet Protocol, which are two significant parts of what is now also known as the Internet Protocol Suit.

The feature that makes TCP/IP different from many other networking protocols is that it was designed to link networks instead of simply linking computers in a network.

1.6: Domain Name System
Domain Name System



- **Computers work best with numbers.**
 - Synonymous to identifying people with names.
- **Domain Name:**
 - Maps or translates the actual numeric IP address into an easy-to-remember alphanumeric name.
- **Internet Network Information Center (InterNIC) Registration Service:**
 - Manages IP addresses and domain name assignment to internet users.

- 10 -



Domain Name System (DNS)

It is as easy to identify computers with numbers, as it is to identify people with names. To bridge this dichotomy, the Domain Name System (DNS), a distributed database, was invented. Domain name maps or translates an actual, numeric IP address into an easy-to-remember alphanumeric name that your Web server uses.

Before DNS can do this for you, you must register any name you want to use. Domain refers to a collection of network host computers, known by the same name.

Your domain name should reflect your organization or corporation.

InterNIC (Internet Network Information Center) Registration Service, manages the task of assigning IP addresses and domain names to Internet users.

Note: InterNIC (Internet Network Information Center) lets you apply for any domain name you like, regardless of your company name. The only restriction is that the name must be available and not already reserved by someone else.

Rightmost part of a name is called its zone. The next part is the name of the company. The part to the left of the company name is the particular machine within the company. Seven domains were established originally.

1.6: Domain Name System
Domain Name System (Contd...)



- **.com:** Commercial organization. Most companies will end up as a part of this domain.
- **.edu:** Educational establishment such as university.
- **.gov:** Branch of the U. S. government
- **.int:** International organization, such as NATO or the UN
- **.mil:** Branch of the U. S. military
- **.net:** Network organization
- **.org:** Non-profit organization



- 11 -

Five domain types are as listed in the following table:

.com	Commercial organization. Most companies end up as a part of this domain.
.edu	Educational establishment such as university
.gov	Branch of the U. S. government.
.int	international organization, such as NATO or the United Nations.
.mil	Mil is a branch of the U. S. military.
.net	Network organization.
.org	Non-profit organization.

1.7: HTTP Basics

- **HTTP is the fundamental means of communication used by WWW.**
- **It defines formal syntax that allows user agents, such as browsers, to interact with web servers.**
- **It is one of the many protocols designed to allow clients to store and retrieve files from servers.**
- **HTTP requests can specify the language the browser would like to see in a page as well as information about how the data is encoded.**

```
graph TD; WB[Web Browser] -- "HTTP Protocol" --> HS[HTTP Server]; HS <--> CP[CGI Program]; CP <--> DB[Database];
```

The diagram illustrates the communication flow in a web server architecture. A 'Web Browser' sends an 'HTTP Protocol' request to an 'HTTP Server'. The 'HTTP Server' interacts with a 'CGI Program', which in turn communicates with a 'Database'.

- 12 -

iLEARN
iGATE Learning, Education And Research Node

Diagram on above slide shows where HTTP Protocol fits in communication:

Apart from HTTP, other examples of protocols are :

- File Transfer Protocol(FTP)
- Common Internet File System (CIFS)
- Network File System (NFS)
- Simple Network Management Protocol (SNMP): Most widely used protocol for monitoring network devices such as hubs, routers, workstations, and computers. Windows NT supports SNMP.

1.7: HTTP
Client Server Interaction in HTTP



➤ **Client Server interaction in HTTP includes following four basic steps:**

- The client opens Transmission Control Protocol (TCP) connection.
- Client then sends a HTTP request such as 'GET .index.HTML'
- Then, the server sends an HTTP response, including a status and a requested object.
- Finally, a TCP connection is ended.

- 13 -



HTTP Version:

Earlier versions were 0.9 and 1.0 – Rarely used

Almost all clients and browsers support 1.1 today and it offers:

- Better performance
- Better support for proxies and cache

1.7: HTTP
HTTP – Stateless Nature



- After a request is made, the client disconnects from the server and waits for a response and the server needs to re-establish the connection after it processes the request and hence HTTP is connection less.
- HTTP being stateless is direct implication of HTTP being connectionless.
 - The server and client are aware of each other only during a request. Afterwards, each forgets the other. For this reason neither the client nor the browser can retain information between different request across the web pages.



- 14 -

In addition to being connectionless and stateless HTTP is also **media independent** which means that any type of data can be sent by HTTP as long as both the client and server know how to handle the data content. How content is handled is determined by the MIME specification.

1.7: HTTP
HTTP – Stateless Nature



➤ **Advantage and Disadvantage of being stateless**

- The stateless design simplifies the server design because there is no need to dynamically allocate storage to deal with conversations in progress. If a client dies in mid-transaction, no part of the system needs to be responsible for cleaning the present state of the server.
- A disadvantage of statelessness is that it may be necessary to include additional information in every request, and this extra information will need to be interpreted by the server

- 15 -



Examples of stateful protocol:

- TCP (Transmission Control Protocol)
- IP (Internet Protocol)
- BGP (Border Gateway Protocol)

1.7: HTTP
URL and Parts of URL



- **HTTP clients use Uniform Resource Locator (URL) to interact with resources**
- **For HTTP, URL is composed of:**
 - Scheme
 - It is implied by the fact that it is HTTP message.
 - Host
 - In HTTP 1.1, it is included in HOST header.
 - Port
 - Port is used by TCP and not HTTP. By default it is 80.
 - Path
 - Query
 - Both Path and Query are contained in request start line.



In HTTP 1.0, the host was not always include in the HTTP messages

1.7: HTTP URL and Parts of URL



- Consider following example:

<http://www.example.com:80/path/hello.jsp?k=10&h=40>

- Case Insensitive

— Scheme	http://	=	HTTP://
— Name	www.example.com	=	WWW.EXAMPLE.COM

- Case Sensitive

➤ Path	/hello.jsp	≠	/Hello.jsp
➤ Query	?H=40	≠	?h=40

- 17 -



URL parts for example given on slide:

Scheme : http
Host : www.example.com
Port : 80
Path : /path/hello.jsp
Query : k=10&h=40

1.7: HTTP
HTTP Request Methods



➤ **GET**

- It is used to retrieve a resource from server

➤ **HEAD**

- When HEAD is used, the headers in the request and response will be the same as when a GET is used, but the server will not return any data. This allows a client to check for the existence or size of a resource without actually retrieving it.

➤ **POST**

- POST is used to pass information to the server. POST allows clients to send messages to forums or update databases.

➤ **OPTIONS**

- The options method is used to determine communications options that are available between the client and server.

- 18 -



**1.7: HTTP
HTTP Request Methods****➤ PUT**

- The PUT method tells the server to replace the requested URI with the data contained in this message. If the resource does not exist, the server will create it.

➤ DELETE

- The DELETE method deletes the resource specified by the request – URI.

➤ TRACE

- The TRACE method is used to send test messages to servers. If successful, the server should respond with 200 OK.

➤ CONNECT

- It is reserved for use with proxies.

- 19 -

**HTTP Status codes**

Request received, continuing process.

This class of status code indicates a provisional response, consisting only of the Status-Line and optional headers, and is terminated by an empty line. Since HTTP/1.0 did not define any 1xx status codes, servers *must not* send a 1xx response to an HTTP/1.0 client except under experimental conditions.

1XX – Informational

This class of status codes indicates the action requested by the client was received, understood, accepted and processed successfully

2XX – SUCCESS

3XX – REDIRECTION

4XX – CLIENT ERROR

5XX – SERVER ERROR

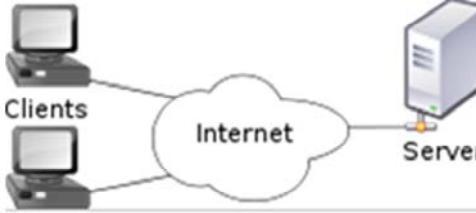
1.8: Servers

Servers

➤ A server is a system (software and suitable computer hardware) that responds to requests across a computer network to provide, or help to provide, a network service.

➤ Servers operate within a client-server architecture, servers are computer programs running to serve the requests of other programs, the clients.

➤ Thus, the server performs some task on behalf of clients. The clients typically connect to the server through the network but may run on the same computer



- 20 -

iGATE
ITOPS for Business Outcomes

LEARN
iGATE Learning, Education And Research Network

Above diagram shows clients communicating with a server via internet.

Servers often provide essential services across a network, either to private users inside a large organization or to public users via the Internet.

1.8: Servers
Servers - Example



- Application Server
- Database Server
- File Server
- Print Server
- **Web Server**

- 21 -



Application server - A server dedicated to running certain software applications

Database server – It provides database services to other computer programs or computers

File server – It provides remote access to files

Print server – It provides printer services

Web server – A server that HTTP clients connect to in order to send commands and receive responses along with data contents

1.8: Servers
Web Servers



- **Web server can refer to either the hardware (the computer) or the software (the computer application) that helps to deliver web content that can be accessed through the Internet.**
- **The primary function of a web server is to deliver web pages on the request of clients using the Hypertext Transfer Protocol (HTTP).**



- 22 -

Delivery of web pages over the web means HTML documents and any additional content that may be included by a document, such as images, style sheets and scripts

Web servers are not always used for serving the World Wide Web. They can also be found embedded in devices such as printers, routers, webcams and serving only a local network. The web server may then be used as a part of a system for monitoring and/or administering the device in question. This usually means that no additional software has to be installed on the client computer, since only a web browser is required (which now is included with most operating systems).

1.8: Servers
Web Servers - Features



- **Virtual Hosting**
 - To serve many websites using one IP address.
- **Large File Support**
 - To be able to serve files whose size is greater than 2GB or 32 bit OS.
- **Bandwidth throttling**
 - To limit the speed of responses in order to not saturate the network and to be able to serve more clients.
- **Server-side scripting**
 - To generate dynamic web pages, still keeping web server and website implementations separate from each other.



A **website** is a set of related web pages served from a single web domain and is hosted on at least one web server, accessible via a network such as the Internet or a private local area network through an Internet address known as a Uniform Resource Locator(URL).

All publicly accessible websites collectively constitute the World Wide Web.

1.8:Servers

Web Servers – Path Translation



- **Web servers are able to map the path component of URL into:**

- A local file system resource (for static requests)
- An internal or external program name (for dynamic requests)

- **Consider URL : <http://www.exam.com/path/file.html> as requested by client.**

- **The client's user agent will translate it into a connection to www.example.com with the following HTTP 1.1 request:**

```
GET /path/file.html HTTP/1.1  
Host: www.example.com
```

- **The web server on www.example.com will append the given path to the path of its root directory.**

- On an Apache server, this is commonly /home/www
- On Unix machines, usually /var/www
- The result is the local file system resource:
`/home/www/path/file.html`

- 24 -



1.8:Servers

Web Servers – Path Translation



- The Web Server then reads the file, if it exists and sends a response to the client's web browser. The response will describe the content of the file and contain the file itself or an error message will return saying that the file does not exist or is unavailable.

- 25 -



1.8: Servers

Web Servers – Examples

- Apache by Apache
- IIS by Microsoft
- nginx by NGINX Inc
- GWS by Google

Developer	Feb-13	Percent	Mar-13	Percent
Apache	101,558,682	54.92%	101,960,513	54.98%
Microsoft	21,484,151	11.62%	22,962,575	12.38%
nginx	21,970,144	11.88%	22,224,423	11.98%
Google	14,976,221	8.10%	15,016,785	8.10%

- 26 -



The above table by netcraft shows number of websites hosted on each of the webservers listed in above slide

1.9: Web Browser
Web Browser



- A web browser (commonly referred to as a browser) is a software application for retrieving, presenting and traversing information resources on the World Wide Web.
- An *information resource* is identified by a Uniform Resource Identifier (URI) and may be a web page, image, video or other piece of content
- A web browser can also be defined as an application software or program designed to enable users to access, retrieve and view documents and other resources on the Internet.

- 27 -



1.9: Web Browser
Web Browser



- Available web browsers range in features from minimal, text-based user interfaces with bare-bones support for HTML to rich user interfaces supporting a wide variety of file formats and protocols.
- Browsers which include additional components to support e-mail, Usenet news, and Internet Relay Chat (IRC), are sometimes referred to as "Internet suites" rather than merely "web browsers".
- Most browsers can be extended via plug-ins, downloadable components that provide additional features.

- 28 -



1.9: Web Browser
Web Browser – Examples



- **Google Chrome**
- **Mozilla Firefox**
- **Microsoft Internet Explorer**
- **Opera by Opera Software**
- **Apple Safari**

- 29 -



1.10: Working of www
Working of WWW



- WWW (World Wide Web) refers to all of the publicly accessible web sites in the world, in addition to other information sources that web browsers can access.
 - These other sources include FTP sites, USENET newsgroups, and a few surviving Gopher sites.
- Typically Internet follows client/server model where:
 - Web-browsers acts as client software on the remote machine.
 - The server software is hosted on the webserver which acts as host.
- Whenever you view a web page on the internet, you are requesting that page from a web server. When you type a URL into your browser (for example, "http://www.igate.com/igate-profile.aspx"), your browser requests the page from the web server and the web server sends the page back:



- 30 -

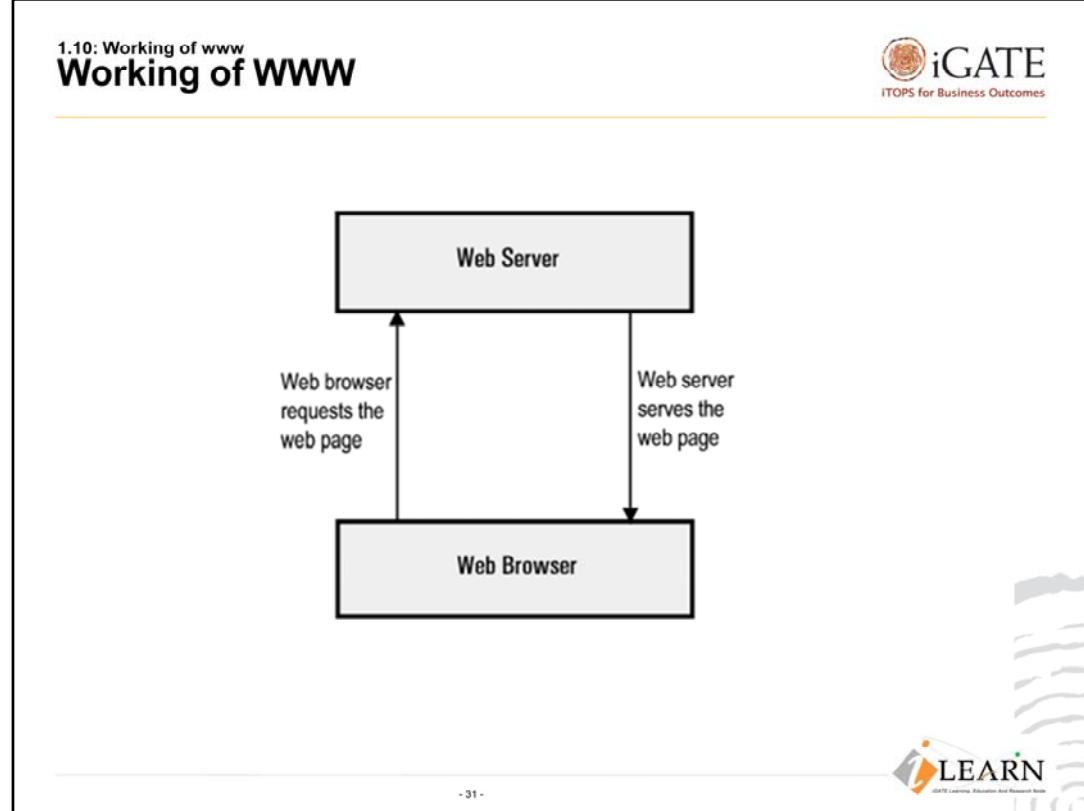


Diagram shown on above slide is simplistic version, Here is the detailed description:

- Your web browser first needs to know which IP address the website "www.igate.com" resolves to. If it doesn't already have this information stored in its cache, it requests the information from one or more DNS servers (via the internet). The DNS server tells the browser which IP address the website is located at. Note that the IP address was assigned when the website was first created on the web server.
- Now that the web browser knows which IP address the website is located at, it can request the full URL from the web server.
- The web server responds by sending back the requested page. If the page doesn't exist (or another error occurs), it will send back the appropriate error message.
- Your web browser receives the page and renders it as required.

Multiple Websites

A web server can contain more than one website. In fact, many hosting companies host hundreds, or even thousands of websites on a single web server. Each website is usually assigned a unique IP address which distinguishes it from other websites on the same machine. This IP address is also what the DNS server uses to resolve the domain name.

It is also possible to configure multiple websites without using different IP addresses using host headers and/or different ports.

Page Not Found

If the requested page isn't found, the web server sends the appropriate error code/message back to the client.

You can create user friendly error messages, then configure your web server to display that page instead of the usual error page. This can add a nice touch to your website.

Default Documents

If you've ever created a website, you may have found that if you have an "index" file (index.html for example), you don't need to specify the name of the file. For example, the following URLs both load the same page:

<http://www.example.com/html/tutorial>

<http://www.example.com/html/tutorial/index.html>

In this example, "index.html" is the *default document*. You can configure your web server so that any file name can be the default document.

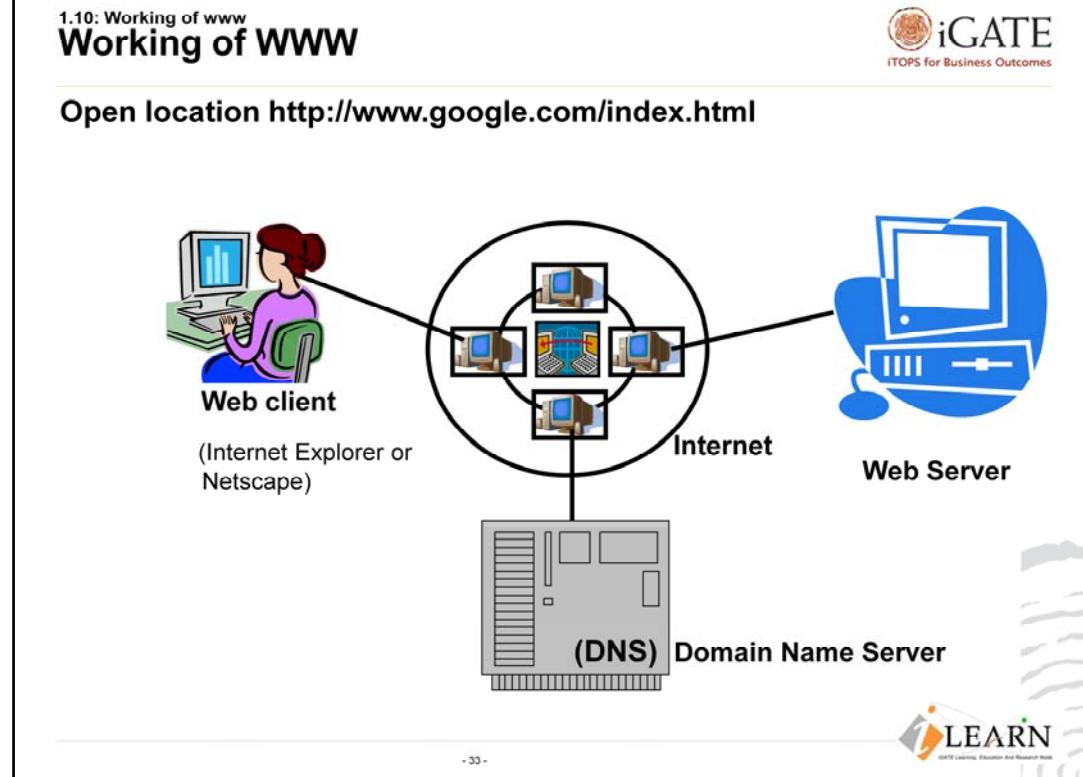
For example, you could configure your web server to use "index.html" in the event no filename has been specified. You could even specify different default documents for different directories if you like.

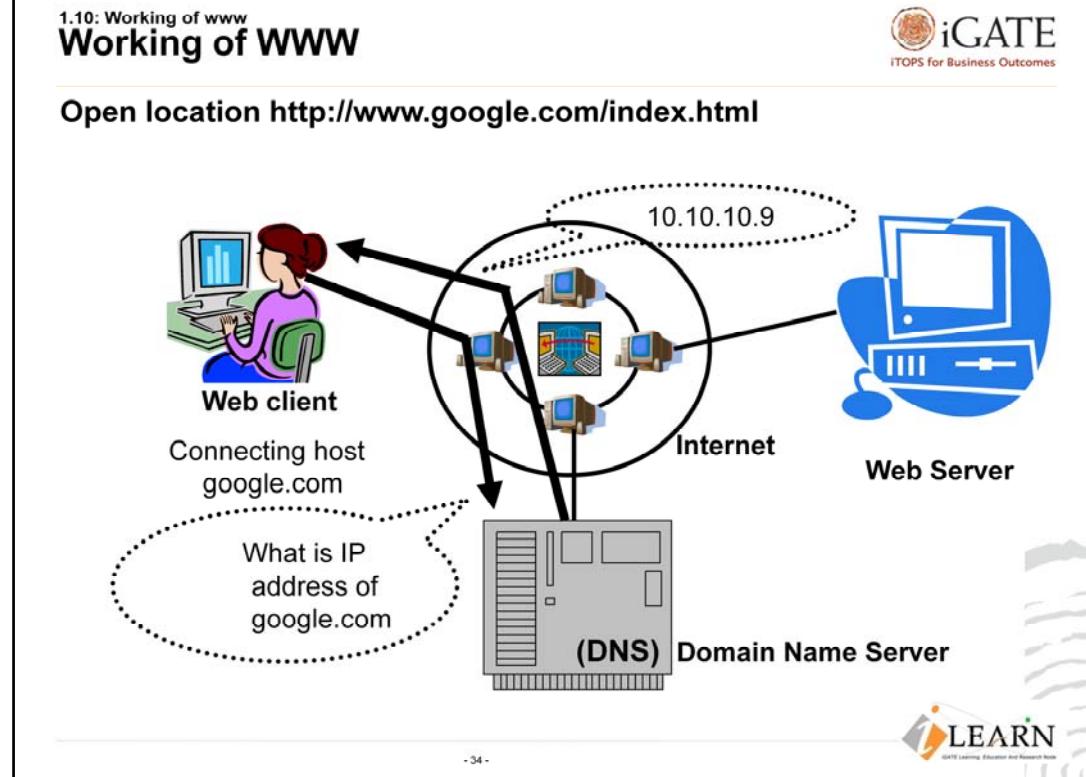
SSL Certificates

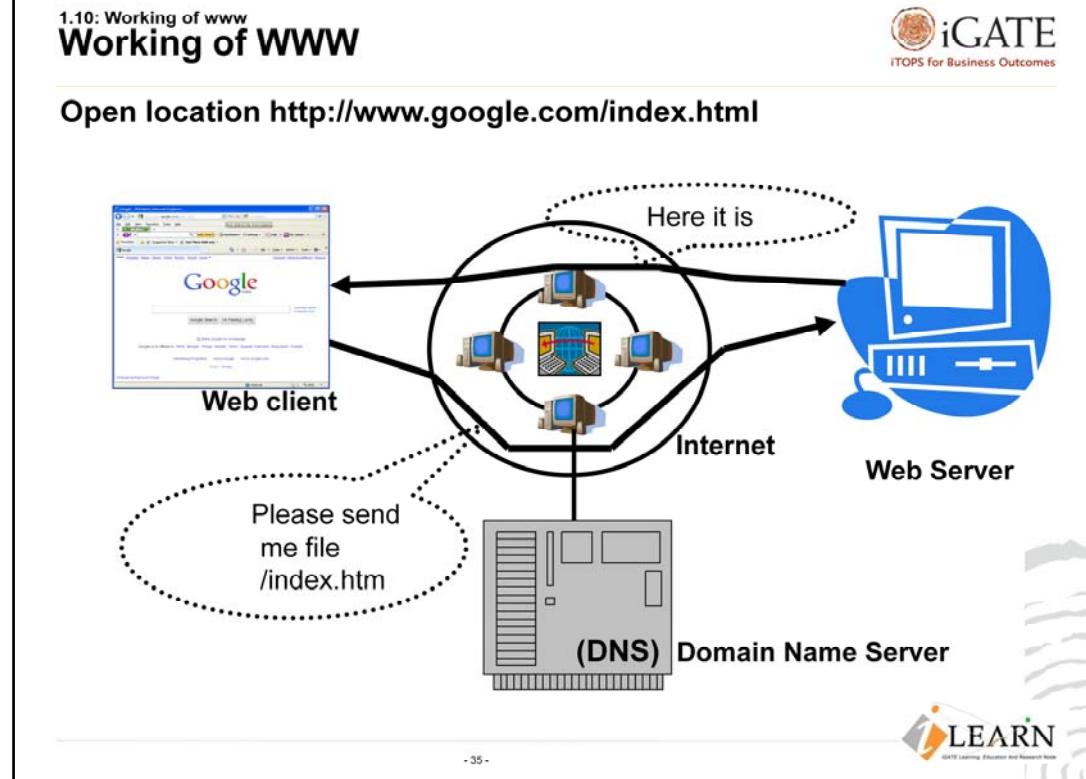
You can apply SSL certificates against a website via the web server. First you need to generate the certificate either by yourself (i.e. using a certificate generator), or by a Certificate Authority (CA). Then, once it has been generated, you apply it to your website via your web server. Applying an SSL certificate to a website is a straight forward task.

Once you've applied an SSL certificate against a website, you can navigate it using HTTPS (as opposed to HTTP). HTTPS encrypts any data that is transferred over the internet. This reduces the possibility of some malicious person being able to read your users' sensitive information.

To navigate a website using HTTPS, you simply replace the HTTP with HTTPS at the start of the URL in your browsers' location bar ("<https://www.example.com>")







1.10: HTML
HTML and WWW



➤ **What is HTML?**

HTML is a language used to describe Web Pages

- HTML stands for Hyper Text Markup Language
- HTML is a markup language
- A markup language is a set of markup tags
- The tags describe document content
- HTML documents contain HTML tags and plain text
- HTML documents are also called web pages

➤ **Web Pages are of two types:**

- Static Web Page
- Dynamic Web Page



- 36 -

HTML tags are keywords (tag names) surrounded by **angle brackets** like <html>

1.10: HTML
Static Web Page



- A static web page is a web page that is delivered to the user exactly as stored, in contrast to dynamic web pages which are generated by a web application.
- A static web page displays the same information for all users, from all contexts, subject to modern capabilities of a web server to negotiate content-type or language of the document where such versions are available and the server is configured to do so.
- Static web pages are often HTML documents stored as files in the file system and made available by the web server over HTTP.
- **Disadvantages:**
 - Any personalization or interactivity has to run client-side, which is restricting.
 - Maintaining large numbers of static pages as files can be impractical without automated tools.

- 37 -



**1.10: HTML
Dynamic Web Page**

- A dynamic web page is a web page with web content that varies based on parameters provided by a user or a computer program.
- For dynamic behavior, client side scripting and server side scripting are used.
- Client-side scripting is changing interface behaviors within a specific web page in response to mouse or keyboard actions, or at specified timing events.
- Server side scripting involves program running on a web server and is used to change the web content on various web pages, or to adjust the sequence of or reload of the web pages. Server responses may be determined by such conditions as data in a posted HTML form, parameters in the URL, the type of browser being used, the passage of time, or a database or server state.

- 38 -



In Client side scripting, the dynamic behavior occurs within the presentation. The Client-side content is generated on the user's local computer system.

Such web pages use presentation technology called rich interfaced pages. Client-side scripting languages like JavaScript or ActionScript, used for Dynamic HTML (DHTML) and Flash technologies respectively, are frequently used to orchestrate media types (sound, animations, changing text, etc.) of the presentation.

Web pages that need server side scripting are often created with the help of server-side languages such as ASP, ColdFusion, Perl, PHP, Ruby, WebDNA and other languages. These server-side languages often use the Common Gateway Interface (CGI) to produce *dynamic web pages*. Three notable exceptions are ASP.NET, JSP, and LSP, which reuse CGI concepts in their APIs but actually dispatch all web requests into a shared virtual machine.

Dynamic web pages are often cached when there are few or no changes expected and the page is anticipated to receive considerable amount of web traffic that would create slow load times for the server if it had to generate the pages on the fly for each request.

Lesson Summary



➤ From this chapter we learnt:

- Internet: Connection of interrelated networks.
- Protocols: Developed to maintain communication standard across:
 - Different computers and operating systems (Platforms).
 - TCIP/IP concepts
 - HTTP concepts
- WWW:
 - Webservers
 - Web Browsers
 - Working of WWW
- HTML
 - Static Web Page
 - Dynamic Web Page

- 39 -



Review Question



➤ **Question 1: URL is a networked extension of the standard filename concept.**

- True/False

➤ **Question 2: HTTP is a Stateless Protocol**

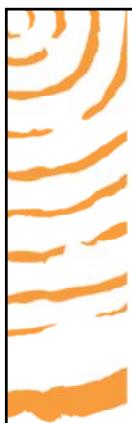
- True/False

➤ **Question 3: Which of the following Webservers host maximum number of websites?**

- Apache
- IIS
- Nginx
- GWS



- 40 -



Web Basics - HTML

Lesson 2. HTML Basics



Lesson Objectives



➤ After completing this module you will be able to:

- Understand the structure of an HTML page.
- Learn to apply physical/logical character effects.
- Learn to manage document spacing.



- 2 -

2.1: HTML Basics

What is HTML?

➤ **HTML is a language for describing web pages.**

- It stands for Hyper Text Markup Language
- HTML is a markup language and not a programming language
- HTML uses markup tags to describe web pages.

- 3 -



HTML is a markup language and these documents describe web pages. Any HTML document contains HTML tags and plain text. HTML documents are also called web pages.

A web browser (like Internet Explorer or Firefox) reads HTML documents and display them as web pages. The browser does not display the HTML tags, but uses the tags to interpret the content of the page:

2.2: HTML Elements

HTML Elements



➤ Most Web documents are created using HTML.

➤ Documents are saved with extension .html or .htm.

➤ Tags are strings in the language surrounded by a less-than (<) and a greater-than (>) sign.

- Opening tag: <html> Ending tag: </html>



- 4 -

HTML documents are text files made up of HTML elements. These files are saved with an extension of .htm or .html. A file with this extension indicates to the browser that the file is an HTML document.

You can check for the document structure and look for the presence of HTML element or tag. HTML elements are defined using HTML tags. Following are features of HTML tags:

- Used to mark-up HTML elements.
- Surrounded by the two characters "<" and ">".
- Normally come in pairs.

HTML element starts with a start tag . Then, the content of the HTML element is placed. This ends with an end tag

- Not case sensitive.
- For HTML, the usual filename extension is .html (.htm for PC-based servers).

Q: Do we need to write the end tag for all the elements?

A: In HTML, there are some tags which do not have a closing tag. For example,
 and <hr> tags. So, it is not necessary to write the closing tags for such tags. But it is a good idea to write an empty tag by providing a "/" in the end tag.

2.2: HTML Elements

Basic HTML Tags



➤ **Not case sensitive.**

➤ **Can have Attributes**

- Attributes are Name-Value pairs added to HTML start tags.



- 5 -

We stated that HTML tags are not case sensitive, but let's always use lowercase tags. To prepare yourself for the next generations of HTML, start using lowercase tags. World Wide Web Consortium (W3C) recommends lowercase tags in their HTML 4 recommendation, and XHTML (the next generation HTML) demands lowercase tags.

Tag Attributes

Tags can have attributes. Attributes provide additional information about HTML elements on your page. This tag defines the `<body>` element of your HTML page. With an added `bgcolor` attribute, you can tell the browser that the background color of your page should be red, like this:

```
<body bgcolor="red">
```

The `<table>` tag defines an HTML table. With an added `border` attribute, you can tell the browser that the table should have no borders:

```
<table border="0">
```

Attributes always come in name/value pairs like this: `name="value"`. Attributes are always added to the start tag of an HTML element.

Quote Styles

Always enclose values in quotes. Double quotes are more common, but single quotes are also allowed. In some rare situations, like when the attribute value itself contains quotes, it is necessary to use single quotes:

```
name='John "ShotGun" Nelson'
```

2.2: HTML Elements

HTML Elements (Code)

 iGATE
ITOPS for Business Outcomes

➤ Code Snippet

An HTML document appears as follows:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<html>
  <head> <title>Title of page</title> </head>
  <body>
    This is my first homepage. <b>This text is
    bold</b>
  </body> </html>
```



1. Save the file as "mypage.html".
2. Click "Browse" (or "Choose File") and locate the HTML file you just created - "mypage.html". Select it and click "Open". Now you should see an address in the dialog box, for example "C:\MyDocuments\mypage.htm". Click OK, and the browser will display the page.

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">

It tells type, version, language of particular document. The first tag in your HTML document is <html>. This tag tells your browser that this is the start of an HTML document. The last tag in your document is </html>. This tag tells your browser that this is the end of the HTML document.

Text between the <head> tag and the </head> tag is the header information. This information is not displayed in the browser window. Text between the <title> tags is the title of your document. Title is displayed in your browser's title bar. Text between the <body> tags is the text that is displayed in your browser. Text between the and tags is displayed in a bold font.

2.3. Document Sections

HTML Document Sections

 iGATE
ITOPS for Business Outcomes

➤ **HTML Head Section:**

- <head>...</head>
 - Page Title, Base URL, Meta Information

➤ **HTML Body Section:**

- <body>...</body>
 - Text, Images, Tables Colors, etc.



- 7 -

Head element can contain information about the document. The browser does not display this information to the user. Following tags can be in the head section: <base>, <link>, <meta>, <script>, <style>, and <title>.

Body element defines the document's body. It contains all the contents of the document (like text, images, colors, graphics, etc.).

2.3: Document Sections

Head Section: Page Title (Code)

 iGATE
ITOPS for Business Outcomes

➤ Code Snippet

Document Title is displayed using <title>.....</title> tag.

```
<html>
  <head>
    < title>
      My First Page
    </ title>
  </head>
</html>
```

- 8 -

 iLEARN
iGATE Learning, Education and Research Network

<TITLE> element does not have any visible effect within a browser's client area; however, the enclosed title appears in the title bar of the browser window.

2.1: Html Basics]
Demo

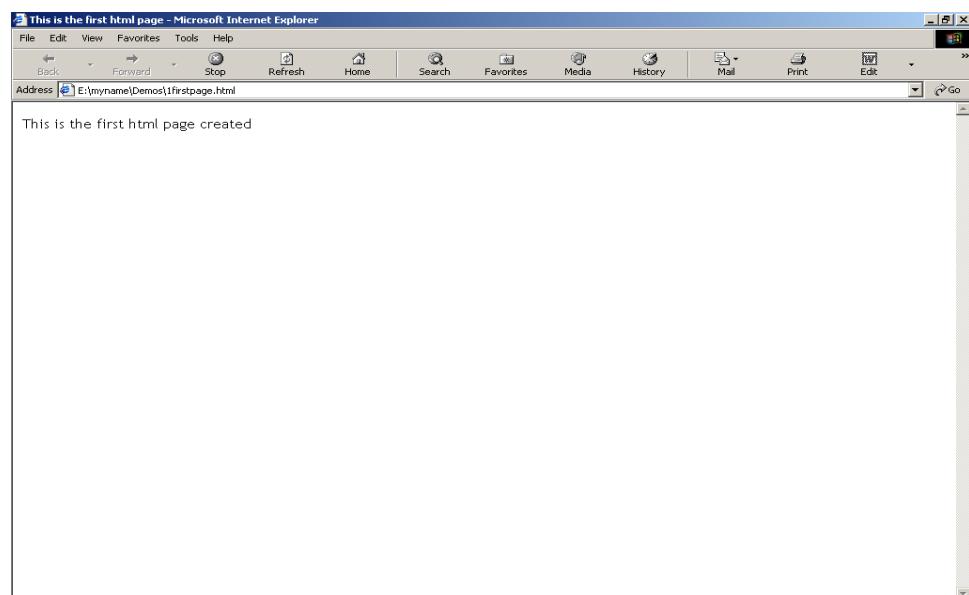
➤ **Firstpage.html**



- 9 -

iLEARN
iGATE Learning, Education And Research Node

```
<html>
<head><title>This is the first html page</title>
<body>This is the first html page created</body>
</head>
</html>
```



2.3: Document Sections

Head Section: Document Base URL (Code)

iGATE
ITOPS for Business Outcomes

➤ **Code Snippet**

```
<html>
  <head>
    <title> Document Base URL Manipulation
  </title>
  <base href="URL/">
  </head>
</html>
Examples:
<base href="http://www.state.edu/images/">
<base href="ftp://ftp.state.edu/images/">
```

- 10 -

 iLEARN
iGATE Learning, Education And Research Network

Base element specifies a base URL for all the links in a page.

Note: The `<base>` tag must go inside the `head` element.

You can set the `<BASE>` element only once in a document, in the header.
`<BASE>` element does not work outside the header.

Assume that the absolute address for an image is:

```

```

Now, we insert the `<base>` tag, which specifies a base URL for all of the links on a page, in its head section:

```
<head>
  <base href="http://www.state.edu/images/" />
</head>
```

When we insert images on the page, in the example above, we just specify the relative address, `` and the browser looks for that file using the full URL, "http://www.state.edu/images/smile.gif".

2.3: Document Sections

Head Section: Meta Information (Code)



➤ **Code Snippet**

```
<meta .....>
<meta name="keywords" content="HTML, CSS" />
<meta http-equiv=refresh content=60 />
<meta http-equiv=refresh
      content="20;url=c:/html/htm3.htm" />
```



- 11 -

META: This element is used to specify additional meta information and appears in the <HEAD> tag. You can specify multiple <META> tags for the document. It provides meta-information about your page, such as descriptions and keywords for search engines and refresh rates. Following are some uses of META:

Define keywords for search engines:

```
<meta name="keywords" content="HTML, DHTML, CSS, XML, XHTML,
JavaScript, VBScript" />
```

Provide a description of your web page:

```
<meta name="description" content="Free Web tutorials on HTML, CSS,
XML, and XHTML" />
```

Define the last revision of your page:

```
<meta name="revised" content="Hege Refsnes, 6/10/99" />
```

Refresh page every 5 seconds:

```
<meta http-equiv="refresh" content="5" />
```

2.3: Document Sections
Demo

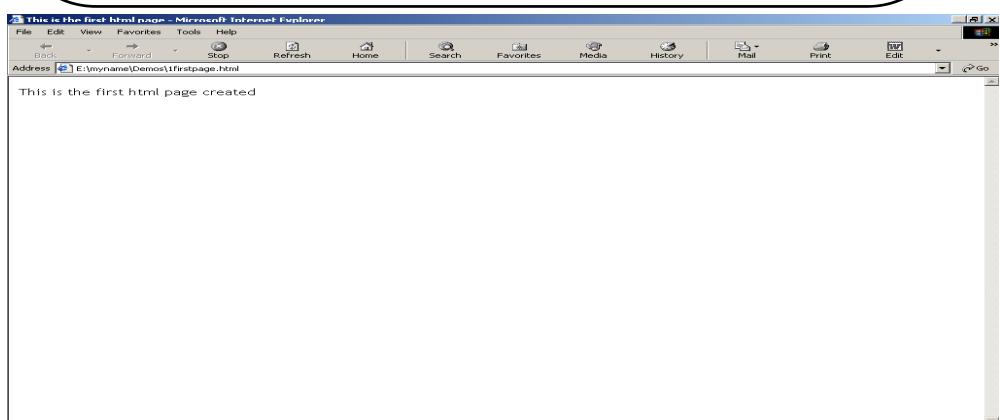
➤ **Meta.html**



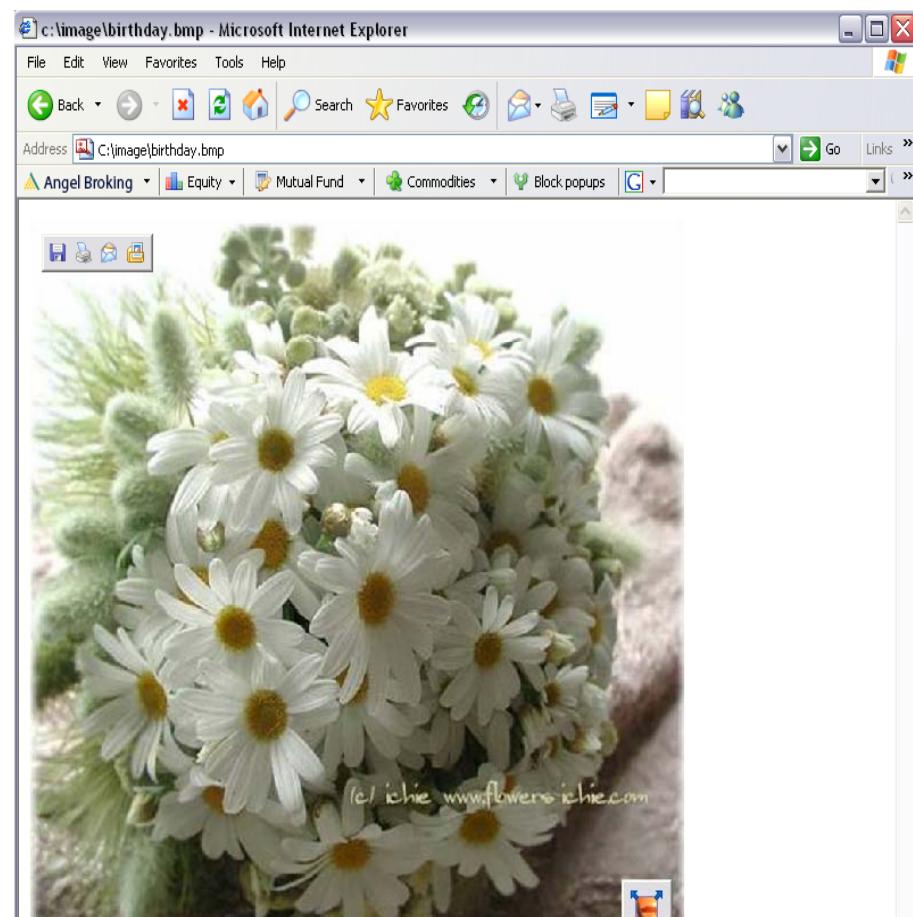
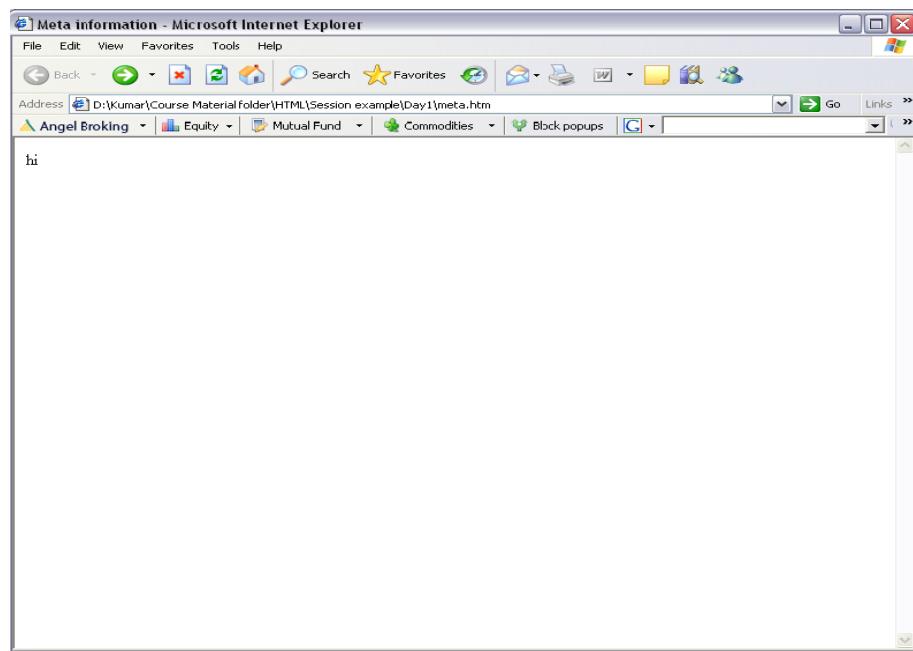
- 12 -

iLEARN
iTOPS Learning, Education And Research Network

```
<html>
<head>
<!-- in meta tag in name attribute user can specify the value like keywords, description --&gt;
&lt;meta name="generator" content="microsoft visual studio 6.0"&gt;
&lt;title&gt;meta information&lt;/title&gt;
<!-- this meta information help you to refresh the page after 2 sec and load image --&gt;
&lt;meta http-equiv=refresh content="2;url=c:\image\birthday.bmp"/&gt;
&lt;/head&gt;
&lt;body&gt;
Hi &lt;p&gt;&amp;nbsp;&lt;/p&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```



Before Automatic Refresh



2.3: Document Sections

HTML Body Section



> <body> Element:

- Represents information content.
- Each document can have at most one <body> element.
- Attributes used in <body> element are:
 - BGCOLOR: Gives a background color to HTML page.
 - BACKGROUND: Use to specify images to the BACKGROUND.
 - TEXT: Specifies text color throughout the browser window
 - LINK: Used to specify link color.
 - ALINK: Specifies the active link color.
 - VLINK: Specifies visited link color.

- 14 -



<BODY> tag defines the HTML element containing the body of the HTML document. <BODY> opening tag can contain a list of attribute-value pairs.

Attributes used in <body> element are: BGCOLOR, TEXT, LINK, ALINK, VLINK, BACKGROUND.

Each HTML document can have at most one <BODY> element. It is not required if all information is contained in the <HEAD> element. Body tag will close like "</BODY>" .

Text attribute of body tag:

```
<body text="red"> OR <body text="#FF0000">
```

Set Colors for Hypertext Links:

```
<body link="red" alink="blue" vlink="purple">
```

Attributes for manipulating hyperlink colors are:

- link : Color of the link.
- alink: Color of text for activated link.
- vlink: Color of text for previously visited link.

Set Web Page Background Color:

```
<body bgcolor="black"> OR <body bgcolor="#000000">
```

Use Image as Web Page Background:

```
<body background="http://www.mysite.edu/img1.gif">
<body background="symbol.gif" text="red" link="blue"
bgproperties="fixed">
```

2.3: Document Sections

Document (Body) Contents



Body Text

- HTML truncates spaces in your text.
- Use
 to insert new lines.
- Use <p> tag to create paragraphs.

Other Elements of Body Section:

- <table> tags are used to create tables.
- tags are used to insert images.

- 15 -



When you write HTML text, you are never sure how text is displayed in another browser. Some computers have large displays whereas some have small ones. Text is reformatted every time the user resizes the window.

Avoid reformatting text in your editor by adding empty lines and spaces. HTML truncates spaces. Any number of spaces as well as a new line counts as one space. Use
 to insert blank lines. You might have noticed that
 tags can be written without the closing tag </br>.

HTML automatically adds an extra blank line before and after some elements, like before and after paragraphs and headings.

Use a <P> tag to start a new paragraph, to which you can assign new attributes. Most browsers also place an extra space after a <P>. A
 causes the browser to maintain the current paragraph attributes but to start placing text on a new line.

</TABLE> is used to close the table tag.

To insert an image, the tag is used. Image should be saved with extension .gif. The tag is empty, which means that it contains attributes only and it has no closing tag.

2.3: Document Sections

Comments in HTML Document



➤ Increase code readability.
➤ Ignored by the browser.
➤ Example of HTML comment:
➤ <!-- This is a Sample HTML Comment -->



- 16 -

Comments are used for better code readability. Browser will ignore a comment. You can use comments to explain your code, or to store program-specific information, which can help you when you edit the source code at a later date.

Comments are not visible to the user, but they are still available to the program. A good practice is to comment text inside the script and style elements to prevent older browsers, that do not support scripting or styles, from showing it as plain text.

To include comments in the document, use the comment tag. Comment is a special tag starting with a "<" sign, followed by "!" and two hyphens. Then, type the commented text. It ends with two hyphens and a ">" sign.

2.3: Document Sections

Demo

➤ **Body.html**



- 17 -

iGATE
ITOPS for Business Outcomes

LEARN
iGATE Learning, Education And Research Network

```
<html>
<head>
<title>body tag</title>
</head>
<body bgcolor="pink" text="red" alink="green" link="yellow">
<a href="body.html">background</a> color of the page is pink.
and text color is red
</body>
</html>
```

2.3: Home Page
Home Page Concepts



- Most popular home pages reflects the personality of the sponsoring organization or corporation.
- Keep the initial home page short and to the point.
- First element visitor sees is a collection of navigation buttons to navigate to other pages.
- When you publish a URL in print or any other marketing material, it points to the location of your home page.



- 16 -

2.4: Physical Character Effects
Physical Character Effects



- **Bold Font:** `...`
- **Italic:** `<i>...</i>`
- **Underline:** `<u>...</u>`
- **Strikethrough** `<strike> or <s>`
- **Fixed width font** `<tt>`
- **Subscript** `<sub>`
- **Superscript** `<sup>`

**BOLD**

`... `: Tags make text between them bold e.g. `patni ` is displayed as "patni".

ITALIC

`<i>...</i>`: Tags make text between them italicized e.g. `patni ` is displayed as "patni".

UNDERLINE

`<u>...</u>`: Tags make text between them underlined e.g. `patni ` is displayed as "patni".

Fixed Width Font : `<TT>text</TT>`

```
<html>
<head> <title> My first page </title> </head>
<body bgcolor="black" text="yellow" link="red" alink="blue"
      vlink="green">
This is <TT>Fixed width</TT> font
</body>
</html>
```

Subscripts and Superscripts

`_{text}` subscripts text e.g. Chemical formula of water is $H₂O$ is displayed as " H_2O :

Formula for a parabola is $y=x²$ is displayed as $Y=X^2$.

STRIKETHROUGH

`<strike> or <s> <strike> or <s>`: Tags make text between them appear striked out e.g. `<s>patni </s>` is displayed as "patni" (with a strike across it).

TYPEWRITER TEXT

`<tt>.....</tt>`: This is normal text in fixed-width font e.g. `<tt>fixed width of rectangle </tt>` is displayed as "fixed width of rectangle".

2.4: Physical Character Effects

Font Manipulation



➤ **Font Face**
—

➤ **Font Color**
—

➤ **Font Size**
—
— <small>
— <big>
— <basefont>



- 20 -

 tag specifies the font face, font size, and font color of text.

Function of the attributes is as given below:

Face: Change the font face.

Size: Provide finer control over the size of the font used in the document. Value of size takes an integer between 1 and 7 (1 being the smallest and 7 being the largest value in the range). Default size is 3.

Color: Change the font color. Value can be specified as a color string constant or hexadecimal value.

<small> Make the font size smaller by an arbitrary amount, place the <small> tag at the beginning of the text and </small> at the end.

<big> Increase the font size by an arbitrary amount. Place the <big> tag at the beginning of the text and </big> at the end.

<basefont> Change the font size of the entire document using the BASEFONT element at the beginning of the document. Usage is as follows:

<BASEFONT SIZE="2">

Also, you can specify relative font size by:

2.4: Physical Character Effects
Demo



➤ **Physicalcharactereffect.html**



- 21 -

2.5: Logical Character Effects

Logical Character Effects



➤ Heading Styles:

- **<hn>.....</hn>**
- **Value of n can range from 1 to 6**

➤ Syntax

```
<h1 align="center">This is level 1 heading</h1>
```



- 22 -

Heading Style

<hn>.....</hn>: Tag identifies *headings* and *subheadings* in a document. Here, *n* is the size of the heading ranging from 1 to 6 where 1 is largest and 6 is smallest. To align heading to the left, right or center of the window use the *align* attribute at the beginning of the heading. For example:

```
<H1>This is an example of level 1 heading<H1>
<H6>This is an example of level 6 heading<H6>
```

Following tags are used to align headings:

<h1 align="left">This is level 1 heading</h1>: Tag moves the heading to the left.

<h2 align="right">This is level 2 heading</h2>: Tag moves the heading to the right.

<h3 align="center">This is level 3 heading</h3>: Tag moves the heading to the center.

```
<html>
<head><title>This is the first html page</title>
<body>This is the first html page created
<h1 align="left">This is level 1 heading</h1>
<h2 align="right">This is level 2 heading</h2>
<h3 align="center">This is level 3 heading</h3>
</body>
</head>
</html>
```

2.5: Logical Character Effects

Logical Character Effects (Contd...)



<code>	Displays any code part in the web page.
<var>	Displays any variable on the web page.
<kbd>	Displays computer commands and arguments.
	Highlights specific areas of text enclosed within it.
	Used for strong emphasis
<dfn>	Designed specifically for words and phrases defined in text.
<address>	Specifies information about the creator e.g. Address, mailing address.

- 23 -



Code

`<code>..</code>:` Tag displays any code part on the web page

e.g. `<code>perform 2000-modify-para</code>` displays
“perform 2000-modify-para”

Var

`<var>....</var>:` Tag displays any variable on the web page e.g.
`<var>count</var>` displays *count*.

Keyboard

`<KBD>....</KBD>:` Tag to display computer commands and arguments, especially those to be entered by the user. The text tagged by `<KBD>` is usually shown in a fixed-width font. Indicating text as keyboard input. E.g. `<KBD>copy *.exe c:\dir1</KBD>` displays `copy *.exe c:\dir1`.

Emphasis

`.....`: Tag provides generic emphasis or highlights specific areas of text without requiring the browser to set a specific physical style for them e.g. `This is patni` displays "*This is patni*".

Strong

`...`: Tag for strong emphasis e.g. `This is an L2 question` displays "This is an L2 questions".

Emphasis and strong emphasis tags highlight specific areas of text without the browser having to use a specific physical style for them.

Definition

`<dfn>.....</dfn>`: Tag is designed specifically for words and phrases that are defined in the text. Physical method of emphasizing the words is left to the browser. This tag changes the style of the text contained between the `<DFN>` and `</DFN>` tags. Style used is dependent on the browser, but is usually either bold or italic e.g. `<dfn>Definition</dfn>`.

Address

`<address>...</address>`: Tag is used to specify information about the creator or maintainer of a Web page. This information can include the name, e-mail address, phone number, mailing address, or other relevant information.

2.5: Logical Character Effects
Special Characters in HTML



➤ **Character Entities**

—Comprise following three parts:

- Ampersand (&),
- Entity name or a #
- Character code
- Semicolon (;)

— Included in HTML page using:

- Character code/Entity number: Include any character using its ISO Latin 1 character code.
- To display ">" symbol, character code is 62 i.e. >



- 25 -

Some characters like the "<" character, have a special meaning in HTML, and therefore cannot be used in text. To display a less than sign (<) in HTML, we need use a character entity.

Character Entities

Some characters have a special meaning in HTML, like the less than sign (<) that defines the start of an HTML tag. If we want the browser to actually display these characters we must insert character entities in the HTML source.

A character entity has three parts: an ampersand (&), an entity name or a # and an entity number, and finally a semicolon (;). To display a greater than sign in an HTML document we write: > or >

The advantage of using a name instead of a number is that a name is easier to remember. The disadvantage is that not all browsers support the newest entity names, while the support for entity numbers is very good.

Note: Entities are case sensitive.

Non-Breaking Space

The most common character entity in HTML is the non-breaking space. Normally HTML will truncate spaces in your text. If you write 10 spaces, HTML will remove 9 of them. To add spaces, use the character entity.

2.5: Logical Character Effects
Alignment of the Text (Code)



➤ **Code Snippet**

```
<center>This line is centered</center>
<div align="center">This line is centered</div>
<h2 align="right">This heading is centered</h2>
<p align="justify">This paragraph is justified</p>
```

- 26 -



`<center>This line is centered</center>`: Tag centers whole blocks of text and other Web page element .

`<div align="center">This line is centered</div>`: Tag centers the text.

`<h2 align="right">This heading is centered</h2>`: Tag moves the heading to the right.

`<p align="justify">This paragraph is justified</p>`: Tag to justify the whole text.

2.6: Managing Document Spacing

Horizontal Lines in a Web Page

 iGATE
ITOPS for Business Outcomes

- **Horizontal Rule in an HTML document: <hr>**
- **Attributes:**
 - Size: Line thickness <hr size="5">
 - Width: Line width either in pixels or % of browser window
 - <hr width="100"> or <hr width="60%">
 - Align: Alignment values can be left, center or right
 - <hr align="center">
 - Color: to display colored horizontal lines
 - <hr color="red">

- 27 -



Size: Thickness in pixels (default is 2).

<hr size=5>

Width: Width of the line either in pixels or the percentage of the browser window's width.

<hr width=100> or <hr width="60%">

Align: Alignment of the horizontal line(center, left or right). <hr align="center">

Color: For the colored horizontal line. <hr color="red">

<hr>...</hr>: Tag adds the horizontal line to the document.

<hr size="5">..</hr>: Tag to change the size of the horizontal line.

<hr width="100"> or <hr width="60%">: Tag to change the width of the line either in pixel or % of browser.

<hr align="center">: Tag makes alignment as centre of the horizontal line.

<hr color="red">: Tag to change the color of line.

2.6: Managing Document Spacing
Example of Horizontal Line in web Page



<hr>...</hr>
<hr size="5">..</hr>
<hr width="100"> or <hr width="60%">
<hr align="center">
<hr color="red">

+ 28 +



Add the notes here.

2.6: Managing Document Spacing

Managing Vertical Spacing



➤ **Paragraphs / Line Breaks**

- <p>: Paragraph Break
-
: Line Break



- 29 -

<p> ...</p>: Tag to start and end the new paragraph.

...</br> Tag maintains current paragraph but enters text in the new line.

2.6: Managing Document Spacing
Demo

➤ **Spacing.html**



- 30 -

LEARN
iGATE Learning, Education And Research Network

```
<html>
<head><title>Horizontal and Vertical spacing</title></head>
<body>
<p>
This paragraph contains a lot of lines
in the source code,
but the browser ignores it.
</p>
<hr size="2" width="50%" color="blue">
<p>
Notice the horizontal rule occupying 50 % of the window width.
</p>
This paragraph contains <br> line breaks in the
source code <br> so this
is the third line displayed within the paragraph.
</body>
</html>
```

2.6: Managing Document Spacing

Preformatted Text (Code)



Code Snippet

Usage of <pre>.....</pre>

```
<html>
  <head><title> Preformatted Text</title>
  <body>
    <pre>Browser would display this paragraph
      as you are viewing here.
      No need to provide line or paragraph breaks.
    </pre>
  </body>
</html>
```

- 31 -



<PRE>...</PRE> Tags display preformatted blocks of text with a fixed-space font. When they appear inside <PRE> tag, white space, line breaks, and tabs are also displayed. It is mainly useful for simulating program listings.

Note: The <pre> tag takes care of carriage returns, tabs, spaces and even the bold tag.

E.g.:

```
<html>
  <head><title>PREFORMATTED TEXT EXAMPLE</title></head>
  <body>
    <h3>EMPLOYEE DETAILS PREFORMATTING</h3>
    <hr>
    <pre>
      <b>EMPLOYEE DETAILS</b>
      EMPID          EMPNAME        DEPT
      1001           JOHN           ACCOUNTS<BR>
      1002           TOM            PURCHASE<BR>
      1003           TAMMY          SALES<BR>
    </pre>
  </body></html>
```

2.6: Managing Document Spacing
Divisions in an HTML Document (Code)



➤ **Code Snippet**

```
<div>.....</div>
<div align="center">
    This text is at the center of the browser window </div>
<div align="left">
    This text is at the left side of the browser window </div>
<div align="right">
    This text is at the right side of the browser window </div>
```



- 32 -

<div>...</div>: Tag creates divisions in Web pages. These can be used to set the alignment for an entire section of the page.

<div align=center>This text is at the center of the browser window.</div>: Tag to align the text to the center of web page.

<div align=left>This text is at the left side of the browser window.</div>: Tag to align the text to the left of the web page.

<div align=right>This text is at the right side of the browser window.</div>: Tag to align the text to the right of the web page.

Difference Between PRE and DIV Tags:

<pre> is used for formatting the entire text.

<div> is used to divide the webpage and use center, left or right attribute to align the text of the divided web page.

Difference Between DIV and SPAN Tags:

DIV is used as a paragraph break as it creates a logical division of the document. In contrast, SPAN simply applies style and alignment as specified.

DIV has *ALIGN* attribute in it which is not present in case of SPAN.

DIV is generally used for a block of text. SPAN is used for words or sentences.

Primary difference between them is that ** does not format by itself. The *<div>* tag includes a paragraph break, as it defines a logical division in the document.

Example of Div Tag (Code)



```
<html>
<head><title>
Divisions Example
</title></head>
<body>
<div align=center>This text is at the center of the browser
window.</div>
<div align=left>This text is at the left side of the browser
window.</div>
<div align=right>This text is at the right side of the browser
window.</div>
</body>
</html>
```

- 33 -



Lab Session



➤ Lab 1



- 34 -

Summary



- Understand the structure of an HTML page.
- Learn to apply physical/logical character effects.
- Learn to manage document spacing.



- 35 -

Review Question



- **Question 1:What does ALINK specifies:**
 - Link color.
 - Active link color.
 - Visited link color.

- **Question 2:HTML document is saved with an extension .xml.**
 - True/False

- **Question 3:A Var tag is used to display the_____in the web page.**



- 30 -

Review Question: Match the Following



1. Code

2. Var

3. Kbd

4. Emphasis

5. Strong

6. Definition

a) Tag displays user-entered computer commands and arguments

b) Tag is used for strong emphasis.

c) Tag displays any code part on the web page.

d) Tag displays any variable on the web page.

e) This tag designed specifically for words and phrases that are defined in the text.

f) Tag for generic emphasis means this tag highlights specific areas of text.



- 37 -



Web Basics - HTML

Lesson 3. Tables



Lesson Objectives



➤ **After completing this module you will be able to:**

- Understand the structure of an HTML table.
- Learn to control table format like cell spanning, cell spacing, border Text to come here (Arial 16, Normal)

- 2 -



Tables

Tables are defined with the `<table>` tag. A table is divided into rows (using `<tr>`), and each row is divided into data cells (using `<td>`). Letters `td` stand for "table data," which is the content of a data cell. A data cell can contain text, images, lists, paragraphs, forms, horizontal rules, tables, etc.

3.1:Tables
Creating Tables



➤ **Topics covered in this module are:**

- Attributes of a Table
- Table Headers
- Table Data
- Table Formatting
- Control Table Borders
- Grouping of Columns

- 3 -



3.2: Attributes of a Table

Attributes of a Table (Code)

iGATE
ITOPS for Business Outcomes

- Use `<table>` tag to create a table.
- Table Attributes:
 - Border:
 - `<table border="2">.....</table>`
 - Alignment:
 - `<table align="center">.....</table>`
 - Width:
 - `<table width="75%">.....</table>`
 - `<table width="400">.....</table>`

- 4 -

 iLEARN
iGATE Learning, Education and Assessment Suite

Border:

To display a table with borders, you will have to use the border attribute:

Example of Border attribute:

```
<table border="1">
    <tr border="1">
        <td>Row 1, cell 1</td>
        <td>Row 1, cell 2</td>
    </tr>
</table>
```

Alignment:

ALIGN attribute of the table element defines the horizontal alignment of the table element.

Values of the align attribute are right, left and center.

Width:

* Width attribute of the table element defines the width of the table element.

3.3: Table Headers

Table Headers (Code)

 iGATE
ITOPS for Business Outcomes

➤ Code Snippet

Table column headings:

```
<table>
    <caption>This is table caption</caption>
    <tr>
        <th>COLUMN 1</th>
        <th>COLUMN 2</th>
        <th>COLUMN 3</th>
    </tr>
</table>
```

- 5 -



Table Caption:

The `<caption>` tag defines a table caption.

The `<caption>` tag must be inserted immediately after the `<table>` tag.

You can specify only one caption per table. Usually the caption will be centered above the table.

Table Headers:

The `<th>` tag defines a header cell in an HTML table.

3.4: Table Data

Table Data (Code)

Code Snippet

```
<table>
  <tr> <th>Column1 Header</th> <th>Column2 Header</th></tr>
  <tr> <td>Cell 1,1</td> <td>Cell 1,2</td> </tr>
  <tr> <td>Cell 2,1</td> <td>Cell 2,2</td> </tr>
</table>
```

- 6 -

 iGATE LEARN
iTOPS for Business Outcomes

An HTML table has two kinds of cells:

An HTML table has two kinds of cells:

- * Header Cells: Contain header information (created with the th element).
- * Standard Cells: Contain data (created with the td element).

The text in a th element is bold and centered.

The text in a td element is regular and left-aligned.

3.4: Table Data

Table Data contd..(Code)

➤ **Code Snippet**

```
<table>
  <thead>
    <tr><td>.....</td></tr>
  </thead>
  <tfoot>
    <tr><td>.....</td></tr>
  </tfoot>
  <tbody>
    <tr><td>....</td></tr>
  <tbody>
</table>
```

- 7 -

 iGATE
ITOPS for Business Outcomes

 LEARN
iGATE Learning, Education and Research Network

The table also has a few more tags to layout your data.

The `<thead>` tag is used to group the header content in an HTML table. The `<thead>` element should be used in conjunction with the `<tbody>` and `<tfoot>` elements.

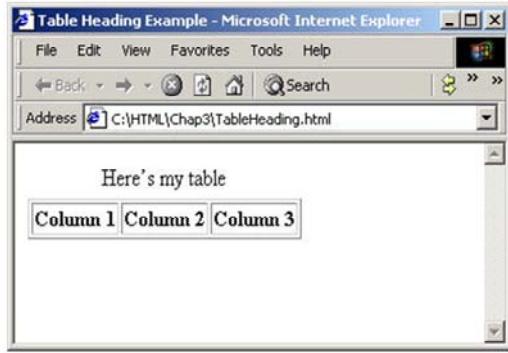
The `<tbody>` element is used to group the body content in an HTML table and the `<tfoot>` element is used to group the footer content in an HTML table.

`<tfoot>` must appear before `<tbody>` within a table, so that a browser can render the foot before receiving all the rows of data.

Note that the `<thead>`,`<tbody>` and `<tfoot>` elements are seldom used, because of bad browser support. Expect this to change in future versions.

3.4: Creating Table
Demo

➤ **Tableheading.html**



Here's my table

Column 1	Column 2	Column 3
----------	----------	----------

- 8 -



```
</head>
<body border=1>
<table>
<caption>Here's my table</caption>
<th>Column 1<th>Column 2<th>Column 3
<!- - Here's the data for the rest of the table..- ->
</table>
</body>
</html>
```

Blank Data Cell (Code)



➤ **Inserting Blank Data Cell :**

- <td></td>
- <td>
</td>

- 9 -



You can avoid missing a border around empty cells. Add a non-breaking space () to empty data cells, to make the borders visible.

3.5: Table Formatting

Table Formatting

- Cell Spanning
- Cell Spacing
- Alignment of Cell Contents
- Table Colors
- Table Background Image

- 10 -



Cell Spanning:

There are two types of cell spanning. Row and column spanning.

Cell Spacing:

Space between cell borders.

Cell Padding:

Space between cell content and cell border.

Alignment of Cell Contents:

Align: Defines horizontal alignment of contents in the column group.

Valign: Defines vertical alignment of contents in the column group.

Table Colors:

You can insert a `bgcolor` attribute in a `<table>`, `<td>`, `<th>` or `<tr>` tag to set the color of the particular element.

Table Background Image:

BACKGROUND sets a background image for the table.

3.5: Table Formatting

Cell Spanning (Code)

 iGATE
ITOPS for Business Outcomes

➤ **Row spanning/Column spanning:**

```
<table>
  <th rowspan=m>Multiple Column Header</th>
  <th colspan=n>Multiple Row Header</th>
</table>
```

➤ **m & n are integers specifying number of rows and columns respectively.**

- 11 -



colspan="number of columns" ~

By default, the number of columns in a table is defined by the number of table data cells appearing in the table row that contains the most data. You would typically place the same number of data cells in each table row. If a table row does not contain the requisite number of table cells, then it will essentially be in 'error' and will be displayed with a missing cell.

rowspan="number of rows" ~

Rowspan attribute works just like the colspan attribute except that you may find the situation a little more difficult to visualize when working with the source code. But once again the principle is the same. By using the rowspan attribute, you can force a table cell to span the number of rows specified by the respective value.

3.5: Table Formatting

Cell Spacing

iGATE
ITOPS for Business Outcomes

➤ **Distance between cells:**

- <table cellspacing=n> </table>
- n: specifies the space between cells.

➤ **Distance between cell edge to the cell data:**

- <table cellpadding=m> </table>
- m: specifies the space between cell wall and the content.

- 12 -

 iLEARN
iGATE Learning, Education and Assessment Suite

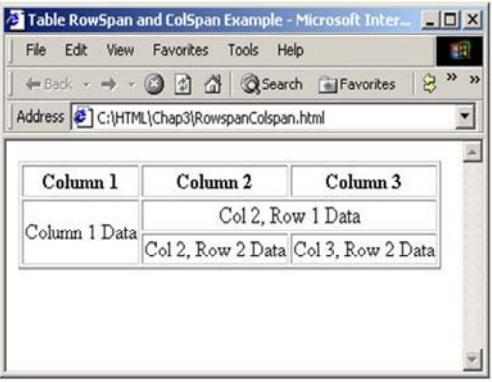
Cell Spacing:

Space between cell borders

Extra space between table cells is visible as space between cell borders if they are turned on.

3.5: Table Formatting
Cell Spanning Demo

➤ **span.html**



The screenshot shows a Microsoft Internet Explorer window displaying a table with three rows and three columns. The first row has three cells labeled 'Column 1', 'Column 2', and 'Column 3'. The second row has three cells labeled 'Column 1 Data', 'Col 2, Row 1 Data', and 'Col 3, Row 1 Data'. The third row has three cells labeled 'Col 2, Row 2 Data', 'Col 2, Row 2 Data', and 'Col 3, Row 2 Data'. The cell in the first row and first column spans two rows, and the cell in the second row and first column spans two columns.



A cartoon illustration of a person with glasses and a blue shirt sitting at a desk, looking at a computer monitor. The monitor displays a blue screen with some graphical elements. The person is holding a mouse and keyboard.



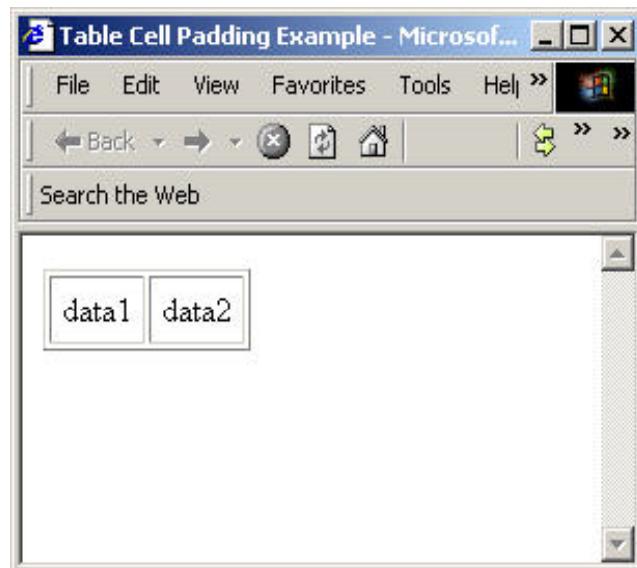
iLEARN
GATE Learning, Education And Research Network

```
<html>
<head>
<title>Table RowSpan and ColSpan Example</title>
</head>
<table border=1>
<tr>
<th>Column 1</th><th>Column 2</th><th>Column 3</th>
</tr>
<tr>
<td rowspan=2>Column 1 Data</td>
<td align=center colspan=2>Col 2, Row 1 Data</td>
</tr>
<tr>
<td>Col 2, Row 2 Data</td><td>Col 3, Row 2 Data</td>
</tr>
</table>
</body>
</html>
```

Cell Padding:

Space between cell content and cell border.

```
<html>
<head>
<title>Table Cell Padding Example</title>
</head>
<body>
<table border=1 cellpadding=5>
<tr>
<td>data1</td><td>data2</td>
</tr>
</table>
</body>
</html>
```



3.5: Table Formatting

Horizontal Data Alignment

iGATE
ITOPS for Business Outcomes

➤ Horizontal alignment of column contents:

```
<table>
<tr><td align="right">Right aligned text</td></tr>
</table>
```

➤ Values of align attribute are:

- left
- center
- right

- 15 -

 iLEARN
iGATE Learning, Education and Assessment Suite

Alignment:

ALIGN attribute of the table element defines the horizontal alignment of the table element

Values of align attribute are right, left and center.

3.5: Table Formatting

Vertical Data Alignment



➤ Use VALIGN attribute to vertically align text in a cell

```
<table>
    <tr><th valign="top">Top aligned Header </th></tr>
</table>
```

➤ Values for valign attribute are:

- TOP - Aligns text at the top of the cell.
- MIDDLE - Aligns text in the middle of the cell.
- BOTTOM - Aligns text at the bottom of the cell.
- BASELINE - Aligns all the text in a row with this alignment attribute set on common baseline.



- 16 -

<colgroup> and <col> are another attempt to help define parts of a table more easily. Before these tags were defined, you had to keep track of how many <td> cells were in each row to know how many columns were in your table, which cells were in the same column, and how to format the cells in a given column.

Align: Defines the horizontal alignment of the contents in the column group.

Width: Defines the width of the column group.

Span: Defines the number of columns the <colgroup> should span.

Valign: Defines vertical alignment of contents in the column group.

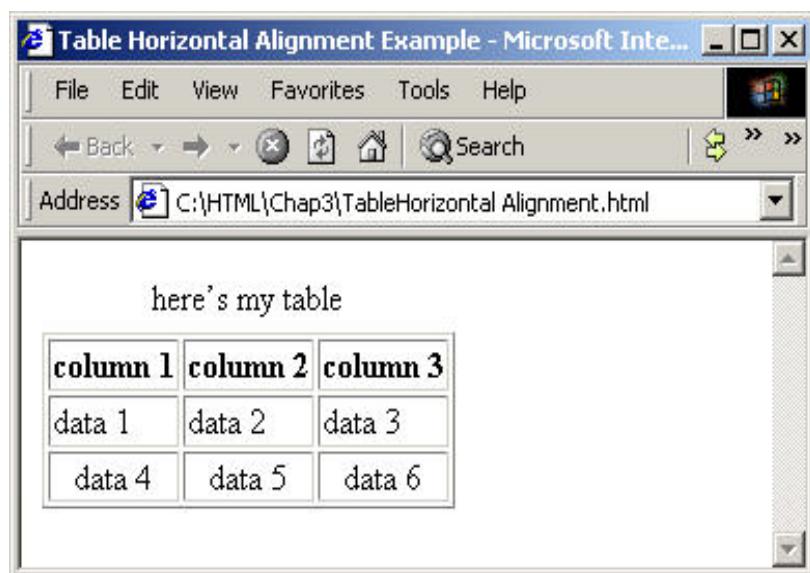
If you wish to put the first three columns in a group and format them the same way then write the HTML code as follows:

```
<table>
    <colgroup width="10%" span="3"> </colgroup>
    <col>
    <col>
    <col>
    <tr>
        table contents.....
    </tr>
</table>
```

Creating new row of Data

```
<table border=1>
<caption>here's my table</caption>
<th>column 1<th>column 2<th>column 3
<tr><td>data 1<td>data 2<td>data 3
<tr><td>data 4<td>data 5<td>data 6
<!-- here's the data for the rest of the table..-->
</table>
```

```
<html>
<head>
<title>Table Horizontal Alignment Example</title>
</head>
<table border=1>
<caption>here's my table</caption>
<th>column 1<th>column 2<th>column 3
<tr>
<td>data 1</td><td>data 2</td><td>data 3</td>
</tr>
<tr align=center>
<td>data 4</td><td>data 5</td><td>data 6</td>
</tr>
<!-- here's the data for the rest of the table..-->
</table></body></html>
```



3.5: Table Formatting

Table Colors (Code)

 iGATE
ITOPS for Business Outcomes

➤ You can insert a **bgcolor** attribute in a **<table>**, **<td>**, **<th>** or **<tr>** tag to set the color of the particular element.

```
<table bgcolor="cyan">
<tr bgcolor="blue">
<th bgcolor="red">Header 1</th> <th>Header 2</th>
</tr>
<tr> <td bgcolor="green">data 1</td> <td>data 2</td>
</tr></table>
```



Table Colors:

You can insert a **bgcolor** attribute in a **<table>**, **<td>**, **<th>** or **<tr>** tag to set the color of the particular element.

3.5: Table Formatting

Table Background Image (Code)

iGATE
ITOPS for Business Outcomes

➤ Add background image to a table:

```
<table border=2 background="picture.gif">  
.....  
</table>
```

- 19 -

 iLEARN
iGATE Learning, Education and Assessment Suite

Table Background Image:

- BACKGROUND sets a background image for the table.
- <TABLE CELLPADDING=8 CELLSPACING=0 BACKGROUND="image.gif">

3.6: Control Table Borders

Control Table Borders

 iGATE
ITOPS for Business Outcomes

- Control Table border with rules or frame attributes inside a <table> tag.
- Rules specify vertical or horizontal divider lines.
`<table rules="rows" border="2">`
- Frame specifies how outer borders are displayed.
`<table frame="vsides" border="1">`
- Use both attributes in conjunction with the "border" attribute.

- 20 -



Rules:

Rules specify vertical and horizontal divider lines.

This attribute specifies which rules will appear between cells within a table. Rendering of rules is user agent dependent.

Possible values:

none: No rules. This is the default value.

cols: Rules will appear between columns only.

all: Rules will appear between all rows and columns.

FRAME: This attribute determines where a frame, surrounding the table, should display. Its possible values are :

void: none.

above: top side.

below: bottom side.

hsides: horizontal sides.

lhs: left-hand side.

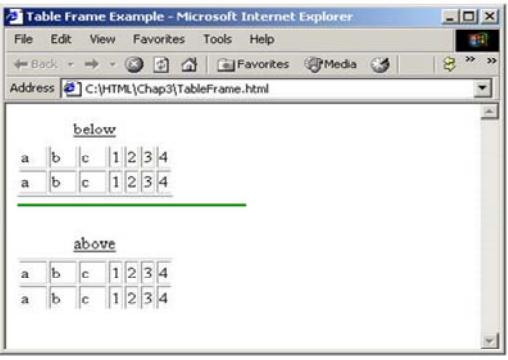
rhs: right-hand side.

vsides: vertical sides.

box/border: all sides.

3.6: Control Table Borders
Demo

➤ **Tableframe.html**



- 21 -

```
<html>
<head>
<title>Table Frame Example</title>
</head>
<body>
<table border frame="below">
<caption><u>below</u></caption>
<colgroup span="3" width="20%" >
<colgroup span="4" width="10%" >
<tr><td>a<td>b<td>c
<td>1<td>2<td>3<td>4
<tr><td>a<td>b<td>c
<td>1<td>2<td>3<td>4
</table>
<hr color="green" size="3" width=200 align=left><p>
<table border frame="above">
<caption><u>above</u></caption>
<colgroup span="3" width="20%" >
<colgroup span="4" width="10%" >
<tr><td>a<td>b<td>c
<td>1<td>2<td>3<td>4
<tr><td>a<td>b<td>c
<td>1<td>2<td>3<td>4
</table>
</body></html>
```

3.7: Grouping of Columns

Grouping of Columns

iGATE
ITOPS for Business Outcomes

- Use **<colgroup>** and **<col>** tags to group columns with common properties
- Attributes of **<colgroup>**:
 - align, width, span and valign.
- Span attribute identifies number of columns in the current group. Default value is 1
- Provide span attribute and omit **<col>** tag

- 22 -

 iLEARN
iGATE Learning, Education and Assessment Suite

<COLGROUP> and <COL> are another attempt to help define parts of a table more easily. Before these tags were defined, you had to keep track of how many <td> cells were in each row to know how many columns were in your table, which cells were in the same column, and how to format the cells in a given column.

Align: Defines the horizontal alignment of the contents in the column group

Width: Defines the width of the column group

Span: Defines the number of columns the <colgroup> should span

Valign: Defines the vertical alignment of the contents in the column group

If you wish to put the first three columns in a group and format them the same way you need to write the following HTML code:

```
<table>
  <colgroup width="10%" span="3"> </colgroup>
  <col>
  <col>
  <col>
<tr>
  table contents.....
</tr>
</table>
```

3.7: Grouping of Columns

Grouping of Columns (Code) (Contd...)



- **<col> tag defines attribute values for one or more columns in a table**

```
<table>
  <colgroup width="10%" span="3"> </colgroup>
  <col width="50">
  <tr><td>10</td> <td>20</td>
    <td>30</td> <td>40</td></tr>
</table>
```

- 23 -



<COL> tag defines attribute values for one or more columns in a table. You can only use this element inside a *colgroup*. Use this element when you specify different attribute values to a column inside a *colgroup*. Without a *col* element a column will inherit all its attribute values from the column group.

3.7: Grouping of Columns

Demo

➤ tabcol3.htm



- 24 -



```
<html><head><title>Table with Colgroup</title></head>
<body>
<table width="100%" frame="void">
<tr><th colspan="3">Ecommerce Stream</th>
<th colspan="4">Mainframe Stream</th>
<colgroup span="3" width="20%" bgcolor="orange">
<colgroup span="4" width="10%" bgcolor="lightgreen">
<tr><td>Internet/HTML<td>Javascript<td>Java
<td>CICS<td>COBOL<td>JCL<td>IMS
<tr><td>JSP<td>Servlet<td>Struts
<td>ADS/O<td>VSAM<td>IDMS<td>MVS
</table>
</html>
```

3.8: Nested Tables

Nested Tables (Code)

iGATE
ITOPS for Business Outcomes

➤ A cell can contain another table within it.

```
<table bordercolor="red" border="1">
<tr> <th>Zone</th> <th> State </th> </tr>
<tr> <td>South</td>
<td><table bordercolor="green" border="1">
<tr> <th>Name</th> <th>Capital City</th> </tr>
<tr> <td>Karnataka</td> <td>Bangalore</td> </tr>
<tr> <td>Tamilnadu</td> <td>Chennai</td> </tr>
<tr> <td>Andhra Pradesh</td> <td>Hyderabad</td>
</tr>
</table></td></tr>
</table>
```

Zone	State								
South	<table border="1"><thead><tr><th>Name</th><th>Capital City</th></tr></thead><tbody><tr><td>Karnataka</td><td>Bangalore</td></tr><tr><td>Tamilnadu</td><td>Chennai</td></tr><tr><td>Andhra Pradesh</td><td>Hyderabad</td></tr></tbody></table>	Name	Capital City	Karnataka	Bangalore	Tamilnadu	Chennai	Andhra Pradesh	Hyderabad
Name	Capital City								
Karnataka	Bangalore								
Tamilnadu	Chennai								
Andhra Pradesh	Hyderabad								

LEARN

Nested Tables:

Nesting tables within tables lets you create some complex effects from just basic HTML table tags. Nesting tables places one table inside of another. To nest a table, start the table as you normally would. When you come to the cell into which you want to place the nested table, enter the table data tag. Then, instead of typing the contents of the cell as you normally would, start the nested table. The table becomes that cell's content. After you end the nested table, be sure to end the table cell it is in with the `</td>` tag.

3.8: Nested Tables
Demo

➤ **tabnest4.htm**



- 26 -

iGATE
ITOPS for Business Outcomes

LEARN
iGATE Learning, Education And Research Network

```
<html>
<head><title>Nested Tables</title></head>
<table width=100% height=100%>
<tr><th bgcolor="red">col1<th bgcolor="green">col2
<tr align=center><td bgcolor="aqua">data1
<td bgcolor="black"><font color="yellow">data2</font>
<table width=100% height=100%>
<tr><th bgcolor="pink">col1<th bgcolor="blue">col2
<tr align=center><td bgcolor="orange">data1<td bgcolor="purple">
<font color="yellow">data2</font>
</table>
</table>
</html>
```

Lab



➤ Lab 2



- 27 -

Summary



➤ **After completing this module you know:**

- Structure of an HTML table
- Control table format such as:
 - cell spanning
 - cell spacing
 - border
- Use tables to format contents of an HTML Page.



- 28 -

Review - Questions



- **Question 1: Rules specifies:**
 - Vertical/horizontal divider lines.
 - Display of Outer borders.
 - Both of the above.

- **Question 2: The <col> tag defines the attribute values for one or more columns in a table.**
 - True/ False?

- **Question 3: Table border can be controlled using _____ or _____ attributes inside a <table> tag.**



- 29 -

Add the notes here.

Review – Match the Following

1.Cell Spacing	1.Defines the horizontal alignment of the contents in the column group.
2.Cell Padding	2.Space between cell borders.
3.Align	3.Joining cells together to make a larger cell.
4.Valign	4.Defines the width of the table element.
5.Cell Spanning	5.Space between cell content and cell border.
6.Width	6.Defines the vertical alignment of the contents in the column group.



- 30 -

Add the notes here.



Web Basics - HTML

Lesson 4. List

Lesson Objectives



- Numbered List
- Bulleted List
- Directory List
- Glossary List



HTML provides support for ordered, unordered and other types of lists

4.1: Numbered List

Numbered List (Code)



```
<html>
<body>
<h2>My favorite cricket teams</h2>
<ol>
<li>INDIA</li>
<li>SRILANKA</li>
</ol>
</body>
</html>
```



+ 3 +

Numbered List:

HTML enables you to create a numbered, or ordered, list that will automatically generate numbers in front of each item in the list.

The number placed in front of an item depends on the location of the item in the list.

At the beginning of the list, place the `` (for ordered list) tag. At the end of the list, place the `` tag and in front of each item in the list, place the `` (for list item) tag.

4.1: Numbered List

START attribute in Numbered List (Code)

Example:

```
<html>
<body>
<ol start=11>
<li>INDIA <li>SRILANKA
<li>PAKISTAN <li>AUSTRALIA <li>SOUTH AFRICA
</ol>
</body>
</html>
```

LEARN
iGATE Learning, Education and Research Solutions

One enables you to alter the style of numbers used in the lists, whereas another sets the starting number of the list.

To start an ordered list at a number other than 1, place the START attribute on the tag at the beginning of the list.

You can also use the VALUE attribute within tag to change the numbering sequence within a list. Refer the above example.

In the above example we consider that our starts from 11.



4.1: Numbered List

Attribute Type



➤ Valid values for the Start Attribute

- A-Uppercase letters
- a-Lowercase letters
- I-Uppercase Roman letters
- i-Lowercase Roman letters
- 1-Standard numbers, default



+ 5 +

You can even select the type of numbering system to be used with the type attribute. For example:

A-Uppercase letters: <OL TYPE=A>

a-Lowercase letters: <OL TYPE=a>

I-Uppercase Roman letters: <OL TYPE=I>

i-Lowercase Roman letters: <OL TYPE=i>

1-Standard numbers: <OL TYPE=1>

Creating a Numbered List

```
<html>
<head>
<title>Numbered List Example</title>
</head>
<body>
<h2>My favorite cricket teams</h2>
<ol>
<li>INDIA
<li>SRILANKA
<li>PAKISTAN
<li>AUSTRALIA
<li>SOUTH AFRICA
</ol>
</body>
</html>
```



4.2: Bulleted List

Bulleted List

```
<html>
<body>
<h2>My favorite cricket teams</h2>
<ul>
<li>INDIA</li>
<li>SRILANKA</li>
</ul>
</body>
</html>
```



To create a bulleted list use the ``(unordered list) tag at the opening of the list and `` at the end of the list.

4.2: Bulleted List

Modify the appearance of bullets



➤ **TYPE** can have one of three values:

- DISC
- SQUARE
- CIRCLE

+ 8 +



To change the type of bullet used throughout the list, place the TYPE attribute in the tag at the beginning of the list.

- **DISC** will give the bullet in disc form.
- **SQUARE** will give the bullet in square form.
- **CIRCLE** will give the bullet in circle form.

4.2: Bulleted List

Example (Code):

➤

```
<li> <b>Unordered Lists</b></li>
<ul>
<li type="disc"> Disc bullet</li>
<li type="circle"> Circle bullet </li>
<li type="square"> Square bullet </li>
</ul>
```



+ 9 +

Creating a Bulleted List

```
<html>
<head>
<title>Bulleted List Example</title>
</head>
<body>
<h2>My favorite cricket teams</h2>
<ul>
<li>INDIA </li>
<li>SRILANKA <li>
<li type=i>PAKISTAN </li>
<li>AUSTRALIA </li>
<li>SOUTH AFRICA </li>
</ul>
</body>
</html>
```



4.3: Directory List

Directory List (Code)

Preceding Text:

```
<dir>
<li>html</li>
<li>xhtml</li>
<li>css</li>
</dir>
```

iGATE
ITOPS for Business Outcomes

LEARN

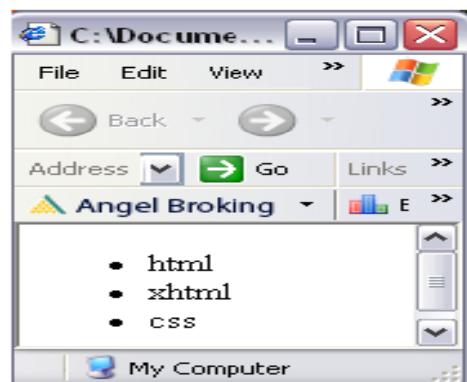
DIR element is similar to the **UL** element. It represents a list of short items, typically up to 20 characters each. Items in a directory list may be arranged in columns, typically 24 characters wide.

Rendering of the DIR tag is similar to that of the UL tag. However, It is recommended to use UL tag instead of DIR tag.

Notes:

It is not permitted to use a block element, list element or TABLE in an LI within a MENU or DIR.

DIR element was deprecated in HTML 4.01, and is not supported in XHTML 1.0 Strict DTD.



4.4: Glossary List

Glossary List (Code)

iGATE
ITOPS for Business Outcomes

```
<dl>
    <dt>HTML Tag</dt>
    <dd>HTML tags are used to markup HTML elements.
        They normally come in pairs. </dd>
        <dt>Title Tag</dt>
        <dd>Title tag is used to display the page title in the
            browser window.
    </dd></dl>
```

C:\Documents and Settings\lab1\Desktop\lassignment\list\demo\glossary.html - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address C:\Documents and Settings\lab1\Desktop\lassignment\list\demo\glossary.html

HTML Tag
HTML tags are used to markup HTML elements. They normally come in pairs.

Title Tag
Title tag is used to display the page title in the browser window.

- 12 -

LEARN

Lab



➤ Lab 3



+ 13 +

Summary



- Types of available list items
- Create bulleted and numbered list items



+ 14 +

Review Question: Questions



- **Question 1: Which tag is used at the beginning of an Ordered List**
 -
 -
 -

- **Question 2: To create a bulleted list tag is used**
 - True/ False

- **Question 3: _____ attribute is used to modify the appearance of bullets**



- 15 -

Review Question: Match the Following

- | | |
|------------------------------------|-----------------------------|
| • <dd> | • 1-Standard numbers |
| • <dt> | • Square Form |
| • <OL TYPE=I> | • A-Uppercase letters |
| • <OL TYPE=A> | • I-Uppercase Roman letters |
| • <OL TYPE=1> | • Data Definition |
| • <li type="square"> Square bullet | • Data Term |



+ 16 +



Web Basics - HTML

Lesson 5. Working with Links



Lesson Objectives



➤ **After completing this module you will be able to:**

- Understand the working of hyperlinks in web pages.
- Learn to create hyperlinks in web pages.
- Add hyperlinks to list items and table contents.



- 2 -



5.1: Creating Links

Creating Links



➤ **Topics covered in this module are:**

- How does a hyperlink work?
- Create Links to Web documents
- Links to E-Mail
- How to add CC n BCC link
- Hyperlinks for Lists and Table data
- Providing target for a hyperlink
- Relative Path

- 3 -



5.2: Hyperlink Work

How Does a Hyperlink Work?

➤ **Hyperlinks access resources on the internet.**

➤ **Tag to create a link is <a> which stands for anchor.**

```
<a href="http://www.mysite.com/login.htm">  
Login Here</a>Hello, Welcome to  
<a href="patni_welcome.htm">Patni</a>
```




Hyperlinks make navigating and finding information on the Internet as easy as possible. When information is published on the Internet, hyperlinks allow readers to access related information on other Web pages or Web sites.

What makes HTML Hyper?

Hyper means more than or outside of. A hyperlink takes you to another page that gives you more than what was on the original page. That page is outside the main page, although it is connected to it.

Create Link to a Local Page

Anchor Tag

Tag to create a link is <A>, which stands for *anchor*. Anchor tag begins with <A> and is followed by a reference comment (HREF) and the name of the page or the URL where the page to link resides. After closing the initial anchor command, you type in the text you wish to highlight as a link, and then close the tag with .

E.g.: The news on Indian Festival.

The link above displays the word *news* in blue with an underline. When you click it, you see the web page named *news.htm*.

Relative Linking

When you create a link from one page to another page on the same computer, it is not necessary to specify a complete Internet address. You can simply provide the local path. Also, if the two pages are in the same directory folder, you can simply use the name of the HTML file. This is called relative linking.

Absolute Linking

In absolute linking, you provide the full URL of the page you wish to link.

5.2: Hyperlink Work

Create Links to Web Documents (Code)

< a href="http://...>Hyperlink Text

I have some < a href="http://www.state.edu/info/info.htm">
older information about this subject.



- 5 -

Create Link to Other Pages

Use Hypertext Transfer Protocol (HTTP), to connect to one computer and transfer a copy of documents from that computer to a browser that runs on another computer. Place the document URL in the anchor of a link using the HREF attribute.

For example:

I have some <A HREF="<http://www.stateu.edu/info/info.txt>"> older information
about this subject.

5.3: Links to E-Mail

Links to E-Mail

➤ **Link to electronic mail:**

```
<a href="mailto:eDude@patni.com?subject =More  
Info">Contact Me</a>Please <a  
href="mailto:author@patni.com>mail</a>your comments to  
me.
```

- 6 -



Create Link to Electronic Mail

HTML includes a URL tag called *mailto*, which identifies the address to which an e-mail is to be sent to. If the link is selected, the browser starts an email program to send a message to the recipient listed in the URL.

For example:

Send me <A HREF="<mailto:abc@pcsbom.patni.com>?subject=My
comments">e-mail with your comments about my page.

CC and BCC in Email (Code)



```
<html>
<body>
<!----add the CC and BCC's--->
<a href="mailto:name@domain.com?Subject=Hello&Cc=jane@domain.com&bcc=joe@domain.com"> Email patni </a>
</body>
</html>
```



- 7 -

Try the following code and see if it works for BCC:

```
<html>
<body>
<!----add the CC and BCC's--->
<a href="mailto:name@domain.com?Subject=Hello
&&Cc=jane@domain.com&&bcc=joe@domain.com"> Email patni </a>
</body>
</html>
```

5.4: Links to Specific Part of a Page

Links to Specific Part of a Page



➤ **Name attribute, used in the anchor tag, identifies a section of a page.**

➤ **Set HREF to the section name. Precede the name with a # symbol**

```
<a name="address1"></a>
Patni Computer Systems Ltd., Unit No. 134/135, SDF V,
Seepz, Andheri-East, Mumbai 92
...Contents of the page...
I am working in <a href="#address1">Patni</a>
```

- 8 -



For documents located on other computers, attach the section name, preceded by the # symbol, at the end of the document URL.

E.g. I am working in

Patni

Create Link to A Specific Part of A Page

NAME attribute, used in the anchor tag, identifies a section of a page. Subsequently, users can access this section via a link within the document or other documents.

To include a link to the named section elsewhere in your page, create a link and set HREF to the name of the section as defined by the NAME attribute. Precede the name with a # symbol to differentiate it from the name of another document.

For example:

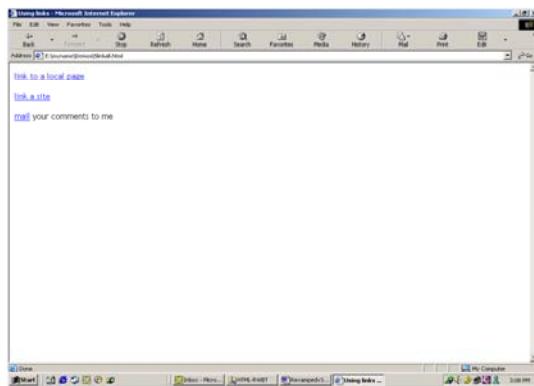
< -- Text of section 1-- >

You find the relevant information in section 1.

To include a link to the named section from another local document, create a local link and include the name of the section, preceded by the # symbol, in the location of the link defined by the HREF attribute.

Demo

➤ Linkall.html



```
<html>
<head>
<title>Using links</title>
</head>
<body >
<a href="6image1.html" target="_blank" > link to a local page</a>
<p>
<a href="http://intranet/patni-intranet" > link a site</a>
<p>
<p>
<a href="mailto:eDude@patni.com?subject=More Info"> mail</a>
your comments to me
</body>
</html>
```

5.4: Hyperlinks in Lists Items

Hyperlinks in Lists Items (Code)



➤ Add links to list items and table contents:

```
<ul>
<li><a href="home.html">mumbai</a></li>
<li><a href="home.html">pune</a></li>
<li><a href="home.html">nasik</a></li>
</ul>
```

- 10 -



5.5: Hyperlinks in Table Elements

Hyperlinks in Table Elements (Code)



➤ Add links to table contents:

```
<table border=1>
<tr><th>team</th><th>points</th><th>grade</th>
<tr><td>
<a href=home.html>mumbai</a></td><td>90</td><td>a</td>
</tr>
<tr><td>
<a href=home.html>pune</a></td><td>86</td><td>b</td>
</tr>
<tr><td>
<a href=home.html>nasik</a></td><td>80</td><td>c</td>
</tr>
</table>
```

- 11 -



5.5: Hyperlinks in Table Elements

Demo➤ **Linkintable.html**

- 12 -



```
<html>
<head>
<title>Links in lists and table Example</title>
</head>
<body>
<ul>
<li><a href=home.html>mumbai</a>
<li><a href=home.html>pune</a>
<li><a href=home.html>nasik</a>
</ul>
<table border=1>
<tr><th>team<th>points<th>grade
<tr><td><a href=home.html>mumbai</a><td>90<td>a
<tr><td><a href=home.html>pune</a><td>86<td>b
<tr><td><a href=home.html>nasik</a><td>80<td>c
</table>
</body>
</html>
```

5.6: Providing Target for a Hyperlink

Providing Target for a Hyperlink



- **Target attribute of the <a> tag specifies where to load the linked document.**

```
<a href="link2.htm" target="frame2">Document  
2</a>
```

- **It takes following values:**

- name of the existing frame/window
- _blank
- _self
- _parent
- _top

- **By default it loads the linked document in the current window.**

- 13 -



Example of target: blank, self, parent and top:

```
<Html>  
<Head><title>my First Frame Page</TITLE>  
</Head>  
<Frameset Cols="25%,75%">  
  <FRAME Name="navigation" Src="navigation_page.Htm">  
  <FRAME Name="content" Src="content_page.Htm">  
  <A Href="new_content.Htm" Target="_top">new Content Page Link Text  
  </A>  
  <A Href="new_content.Htm" Target="_parent">new Content Page Link Text  
  </A>  
  <A Href="new_content.Htm" Target="_self">new Content Page Link Text  
  </A>  
  <A Href="new_content.Htm" Target="_blank">new Content Page Link Text  
  </A>  
</Frameset>  
</Html>
```

5.6: Providing Target for a Hyperlink

blank (Code)

➤ “_blank”:

- opens a new document in a new window.

➤ This code produces this:

```
<A HREF="newwindow.html" TARGET="_blank">a new window</A>
```

- 14 -



Using target="blank"

This causes the link to open in a totally new browser window, with the page with the referring link still open behind it. Unlike [JavaScript pop ups](#), the developer has no control over the size of the resulting window. It depends on what the browser did the last time before it was closed.

5.6: Providing Target for a Hyperlink**_self (Code)****➤ “_self”:**

- Puts new document in the same window and frame as the current document.
- Works the same as if you had not used TARGET at all.

➤ This code produces this go to:

```
<A HREF="selftarget.html" TARGET="_self">next</A> page
```

- 15 -



Using target="_self"

This loads the page within the same frame as the link tag.

5.6: Providing Target for a Hyperlink

_top (Code)

➤ “_top”:

- loads linked document in the topmost frame. This means that the new page fills the entire window.

```
<A HREF="selftarget.html" TARGET="_top">top</A>
```

- 16 -



Using target=_top“

This, within a link tag causes the new page to load in the full body of the window. It is useful if you want to break out of the frameset you have created in order to have a frameless page.

5.6: Providing Target for a Hyperlink**_parent (Code)****➤ “_parent”:**

- Used when a frameset file is nested inside another one
- A link in one of the inner frameset documents which uses "_parent" loads the new document where the inner frameset file had been

```
<A HREF="bigframe.html" TARGET="_parent">bigframe</A>
```



- 17 -

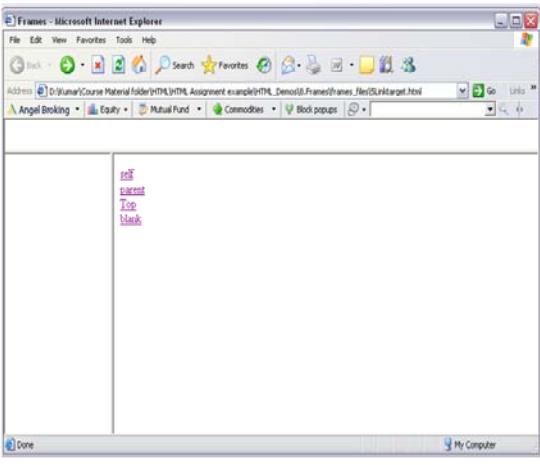
Using target=_parent

This is similar to target=_top but refers to the immediate parent of a frame. In more advanced frame usage there may be several nested frames. This allows more control over which frames are specified. (It is actually something developers rarely need to use).

5.7: Hyperlinks in Table Elements

Demo

➤ **Linktarget.html**



- 18 -



```
<HTML>
<HEAD><TITLE>Frames</TITLE></HEAD>
<FRAMESET rows=10%,*>
    <FRAME border=0 name=top src="blank" noResize scrolling=no>
    <FRAMESET cols=20%,*>
        <FRAME name=left src="blank">
        <FRAME name=parent src="parent.html">
    </FRAMESET>
</FRAMESET>
</HTML>
```

Relative Path



- Relative paths change depending upon what page the links are located on.
- Links in the same directory as the page have no path information listed
`<filename>`
- sub-directories are listed without any preceding slashes
`<sub directory/filename>`
- links up one directory are listed as
`<../filename>`

+ 19 +



LAB

➤ Lab 4



- 20 -

Summary



➤ **After completing this module you now:**

- Understand the work of hyperlinks in web pages
- Learnt to create hyperlinks in web pages
- Learnt to add hyperlinks to list items and table contents



- 21 -



Review Question



- **Question 1:** The name attribute along with # used in the anchor tag identifies:
 - Page section.
 - Path to load the linked document.
 - Address to send the e-mail to.

- **Question 2:** URL tag mailto, identifies the address from where e-mail is to be received.
 - True/ False

- **Question 3:** _____ attribute of the <a> tag specifies where to load the linked document.



- 22 -



Review Question: Match the Following

"_blank"	Identifies address to send the e-mail to.
"_self"	Only filename is needs to be specified.
"_top"	Gives entire address of the page.
mailto	Puts the new document in the same window and frame as the current document
Relative Linking	Opens the new document in a new window.
Absolute Linking	Loads linked document in the topmost frame.

- 23 -





Web Basics - HTML

Lesson 6. Image Handling



Lesson Objectives



➤ **After completing this module you will be able to:**

- Understand the role of images in web pages
- Learn to add images to web pages
- Learn to use images as hyperlinks



- 2 -

Image Formats



➤ Some of the widely used image formats

- GIF : Graphical Interchangeable Format
- JPEG : Joint Photographic Expert Group
- PNG : Portable Network Graphics

- 3 -



•**GIF** images are quickly downloadable, because its size is less. It is used for animated images, thumbnails, B/W image . It supports only 256 colors

•**JPEG** images uses [lossy](#) compression method, meaning that some original image information is lost and cannot be restored, possibly affecting image quality.

•**PNG** images are lossless, portable, well-compressed storage of raster images. It supports image transparency

6.1: Inline Image

Attributes of an Inline Image



- **Images are added into a document using tag.**
- **Attributes of tag are:**
 - Alt: Alternative text to display for non-graphical browsers.
 - Align: Image horizontal alignment. Valid values: left, right, center.
 - Vwidth: Space above and below an image and surrounding text.
 - Hwidth: Space between image and surrounding text.
 - Border: Graphic border. Takes a numerical value to determine border thickness. A higher number indicates a thicker border.
 - Width/Height: Sets the width and height of the image.
 - Src: Specifies the image path or source.



- 4 -

After you have the image, save it in GIF format. Then, use the tag to include the image in your document. The tag is empty, which means that it contains attributes only and it has no closing tag.

Following list shows attributes and options you can include inside the tag:

Alt: Alternative text to display for non-graphical browsers

Align: Aligns graphic with surrounding text. Values include left, right, and center

Vwidth: Spacing above and below an image and surrounding text.

Hwidth: Spacing between an image and surrounding text.

Border: Sets a border around a graphic. It takes a numerical value to determine border thickness. A higher number indicates a thicker border.

Width: Sets the width of the image.

Height: Sets the height of the image.

6.1: Inline Image**Text & Image Alignment (Code)**

```
  
<p><br clear=left>
```

Here is the same picture aligned right:

```
  
<br clear=right>
```

This text should no longer flow around the image

**Align Images and Text with Advanced Tags**

HTML 2 allows simple alignment of text and images by adding ALIGN attributes to the tag. HTML 3 expands this attribute by adding left and right options. When you use these options, text flows down the right or left side of an image aligned to them. Text flow continues until either the text passes the image, ends, or browser encounters a <BR CLEAR="Left/Right/all"> tag.

```
<body>  

```

Add the ALIGN option to the tag. Basic format of this tag with the ALIGN option is:

```

```

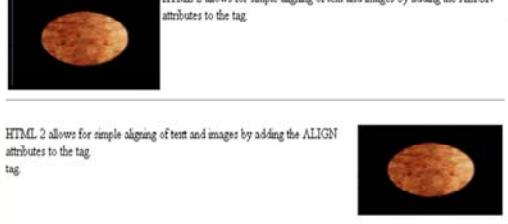
HTML 3 added left and right to these. Left puts the image on the left side of the browser and then aligns any adjacent text to it. Place any descriptive text around the first image.

Insert the <BR CLEAR=LEFT/RIGHT> command at the spot in the descriptive text where text is to stop flowing around the image.

6.1: Inline Image**Demo****➤ Image.html**

The Example of image alignment

HTML 2 allows for simple aligning of text and images by adding the ALIGN attributes to the tag.



- 6 -



6.2: Image as Link**Use of Image as a Hyperlink (Code)****➤ Images used as hyperlinks:**

```
<A href="6image1.html"></a>
```

➤ Images contained within a table:

```
<table align="center" border="2">
<tr><th>Product</th><th>Cost</th><th>Image</th></tr>
<tr> <td>Pencil</td> <td>$8</td>
    <td></td> </tr>
<tr> <td>Brush</td> <td>$15</td>
    <td></td></tr></table>
```



- 7 -

Use Image as Link

To create an image link, place the **** tag with the image URL between **.....** tags.

```
<H2>MY FIRST IMAGE LINK</H2>
<A HREF="BALLOON.HTM"><IMG
SRC="BALLOON.GIF"></A>Click on it to have
additional information on it.<P>
```

6.2: Image as Link

Demo➤ **imagelink2.html**

- 8 -

```
<html>
    <head> <title> Table with Text & Images </title> </head>
<h1>Images in tables and Image as a link</h1>
    <table align="center" border="2">
        <tr><th>Product</th><th>Cost</th><th>Image</th></tr>
        <tr> <td>Pencil</td> <td>$8</td>
            <td></td> </tr>
        <tr> <td>Brush</td> <td>$15</td>
            <td></td></tr>
        <tr> <td>Pin</td> <td>$3</td>
            <td></td></tr>
    </table>
    <br>
    <A href="6image1.html"></a>
    </body>
</html>
```

6.3: Image Maps

Image Maps

➤ **Image Maps come in two flavours:**

- Server-side maps: Contain linking information on the server
- Client-side maps: Linking information is bundled along with the HTML document



Difference Between Client Server-side Image Maps:

In Server-side image maps, when you click an image leading to multiple links, you actually invoke a cgi-bin program. The program reads a separate map file that tells it where to go based on the section of the image you click. The program returns the destination to the web [browser](#), which opens the page. Image map is server-side because the web browser needs to contact the remote host to know which site it should contact.

Client-side image maps, in contrast, do not require a cgi-bin program to function. The image map is actually an [HTML](#) construct that can be contained on the same page as the clickable image. Rather than the program on the remote host figuring out what site to go to, the web browser itself knows what regions of the image are associated with which sites. The browser, rather than the remote host, interprets the image map.

Client-side image maps are usually faster than server-side image maps and, in general, easier to create.

Their main disadvantage is that older browsers, and less fully featured ones, may not support them. Fortunately, it is usually possible to create a clickable image that can have both a server-side and a client-side image map.

6.3: Image Maps

Clickable Image Maps – Client Side

iGATE
ITOPS for Business Outcomes

➤ **Steps to include client-side image maps:**

- Identify areas on the image that you need hyperlink to a document.
- Get their pixel locations
- In HTML documents, do the following:
 - Insert the `<map>` tag and specify the name.
 - Insert `<area>` tag between `<map>` tag. Specify the `shape` attribute value.
 - Specify `coords` attribute, to identify corners of the image map area.
 - Attribute `href` identifies the document to load if this area is selected.

- 10 -

LEARN

Include Client-side Image Maps

Netscape and Microsoft's Internet Explorer support client-side image maps in which the different regions of the image are specified within the document or in another HTML document. No CGI applications are required to read client-side image maps. Therefore, they work even while not connected to the Internet. Perform the following steps:

1. Open your document, and then load an image with a viewer such as Photoshop. This allows you to get the pixel locations from the image so you can specify them in links.
2. Identify areas on the image that you wish should serve as links.
3. Note pixel locations of the upper-left and lower-right corners of the rectangle.
4. In the HTML document, insert the `<MAP>` and `</MAP>` tags.
5. Use the `NAME` attribute for `<MAP>` to give a unique name to the image map.
6. Between `<MAP>` and `</MAP>` tags, insert an `<AREA>` tag. There should be one `<AREA>` tag for each link on the map.
7. `SHAPE` attribute of `<AREA>` identifies the shape of the area in the image map. As only rectangles are currently supported, `SHAPE` takes only one value, "RECT".
8. `COORDS` attribute identifies boundaries of the area on the image map. For `RECT`, `COORD` has four values, separated by commas: x and y coordinates of the upper-left and lower-right corners.

6.3: Image Maps

Clickable Image Maps – Client Side (Contd...) (Code)



- **Usemap attribute of refers to the image map**

```
<map name="mymap">
<area shape="RECT" coords="10, 10, 30, 50" href="info1.htm"/>
<area shape="RECT" coords="50, 50, 70, 70" href="info2.htm"/>
<area shape="RECT" coords="90, 90, 120, 95" href="info3.htm"/>
</map>

```

- 11 -



For each area, the HREF attribute identifies the URL of the document that should be loaded if the user selects that area. Document can be a local file or a document on another server

You can reference the image map information to an image on the page by adding the USEMAP attribute to the element. USEMAP is set equal to the name of map information from the NAME attribute of <MAP>.

```
<IMG SRC="IMAGEMAP.GIF" ALT="An Imagemap"
USEMAP="#mymap">
```

6.3: Image Maps

Client-Side Maps (Code)

➤ Example:

```
<A Href="http://www.webdevelopersnotes.com/">
<Img Src="home.gif" Width="152" Height="25" Alt="Back to
homepage" Border="0"/>
</A>
```

- 12 -



Create Server-side Clickable Image Map

Server side maps also have a main viewable image (IMG) like client-side maps. Instead of the *usemap* attribute, they set only the Boolean attribute *ismap*.

When a user activates a region of a server-side image map with a mouse, pixel coordinates of the click are sent to the server-side agent specified by the href attribute of the 'A' element. Server-side agent interprets the coordinates and performs some action.

For server-side image maps, the client browser just sends the coordinates of a mouse click to the server where calculations are made and the appropriate page is sent back to the client. Because of this essentially visual (and mouse) method of selecting a new page, there is no way to make server-side image maps accessible.

With server-side image maps, "active regions" in the picture only display a set of coordinates. Browsers cannot indicate to the user the URL that will be followed when a region of the map is activated. Therefore, a redundant text link is necessary to provide access to the page.

Demo



➤ clientsideimagemap.html



- 13 -

```
<html>
<head><title>Client-Side Image Map</title></head>
<body>
<map name="menu">
<area shape=rect coords="4,4,96,36" href="home.htm" >
<area shape=poly coords="5,39,95,104,95,39,5,39" href="fun.htm">
<area shape=poly coords="4,44,95,113,6,114,4,44" href="games.htm">
<area shape=circle coords="51,164,36" href="work.htm">
<area shape=rect coords="16,215,89,284" href="6clientsideimagemap.htm">
<area shape=default href="6clientsideimagemap.htm">
</map>

</body></html>
```

Lab



➤ Lab 5



- 14 -

Lesson Summary



➤ After completing this module you now:

- Understand the role of images in web pages
- Learn to add images to web pages
- Learn to use images as hyperlinks



- 15 -

Review - Questions



➤ **Question 1:** Which attribute of inline image sets the amount of space above and below an image and surrounding text?

- Vwidth
- Hwidth
- Width/Height



➤ **Question 2:** Client-side image maps are usually faster than server-side image maps.

- True/ False

➤ **Question 3:** _____ contain the linking information on the server.



- 16 -

Review Question: Match the Following

Alt	Space above and below an image and surrounding text.
Align	Space between image and surrounding text.
Vwidth	Set a border around a graphic.
Hwidth	Sets the width and height of the image.
Border	Alternative text to display for non-graphical browsers.
Width/Height	Image horizontal alignment.





Web Basics - HTML

Lesson 7. HTML Forms for User Input

Lesson Objectives



➤ **After completing this module you will be able to:**

- Understand the role of forms in web pages
- Understand various HTML elements used in forms.



- 2 -

7.1. Forms

HTML Forms for User Input

 iGATE
ITOPS for Business Outcomes

➤ **Topics covered in this module are:**

- Data Submission using a Form
- Types of Form Fields:
 - Single line text field
 - Text area
 - Check box
 - Radio buttons
 - Password fields
 - Pull-down menus
 - File selector dialog box

- 3 -

 iLEARN
iGATE Learning, Education And Research Network

Forms:

HTML forms are used to accept of user input.

A *form* is an area that contains form elements.

Form elements are elements that allow users to enter information (text fields, text area fields, drop-down menus, radio buttons, checkboxes, etc.) in a form.

Define a form with the <form> tag.</FORM>

7.1. Forms

Data Submission using a Form (Code)



- User input forms are created using <form> tag

```
<form method="get/post" action="URL"
      enctype="Encryption Type">
  Field definitions
</form>
```

- method, action and enctype attributes are all optional.



Basic syntax for the <FORM> tag is:

```
<FORM METHOD="Get or Post" ACTION="URL" ENCTYPE="type">
  Field definitions
</FORM>
```

<FORM> tag tells a browser that there is a *fill-in-the-blank* form in this HTML document.

Method:

METHOD attribute states the method to use when you send the form to the server. Two acceptable methods are GET and POST.

GET sends information entered in the form to the server at the end of the URL.

POST sends information entered in the form to the server as a data body/document.

ACTION attribute:

Gives the address of the script that processes the form.

Defines the name of the file to send the content to. File defined here typically does something with the received input.

Enctype:

Specifies how the data is to be encoded.

Applies only if you use the POST method. There is only one possible value, the default value "application/x-www-form-urlencoded".

7.1. Forms

Types of Form Fields



➤ **<input> tag is used to create form input fields.**

- Type attribute of <input> tag specifies the field type
 - Single line text box <input type="text">
 - Password field <input type="password">
 - Hidden field <input type="hidden">
 - Radio button <input type="radio">
 - Checkbox <input type="checkbox">
 - File selector dialog box <input type="file">
 - Button <input type="button">
 - Submit/Reset <input type="submit/reset">

- 5 -

**Input:**

The most used form tag is the <input> tag. Input type is specified with the type attribute. Type attributes are:

Text
Password
Hidden
Radio
Checkbox
File
Button
Submit/reset

Single Line Text Field

<INPUT> tag: Add an <INPUT> tag to accept some input from the reader. This is a standalone tag. This is the default input type .Syntax is as follows:

```
<input type="text" name="UserID" size=20  
value="Guest" maxlength=unlimited [disabled] [readonly]>  
<input type="text" name="name" size=30  
value="Shilpa"
```

Give the user the option to submit this form or reset it and enter fresh values. To do this, use the <INPUT> tag and set TYPE to either “SUBMIT” or “RESET”.

```
<input type="submit" name="submit_button" value="Submit">  
<input type="reset" name="reset_button" value="Reset">
```

TYPE attribute tells the browser the type of button used. NAME is a variable that you can access later when referring to this information. VALUE is button caption.

Text Fields: These are used to type letters, numbers, etc. in a form.

Example:

```
<form>  
    First name: <input type="text" name="firstname">  
    <br>  
    Last name:<input type="text" name="lastname">  
</form>
```

Text Area:

The <textarea> tag defines a multi-line text input control.

A text area can hold an unlimited number of characters. Text renders in a fixed-width font (usually Courier).

You can specify text area size with *cols* and *rows* attributes. An even better way to do it is through CSS *height* and *width* properties.

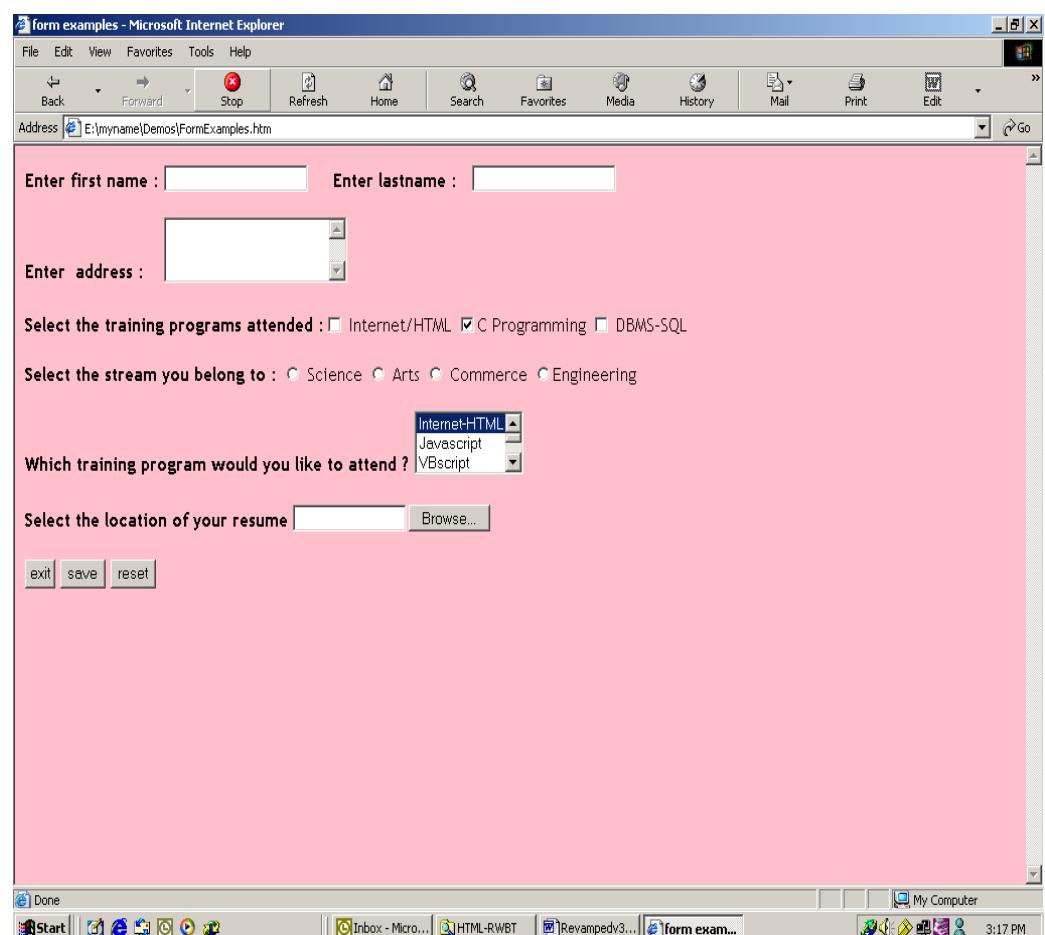
```
<textarea rows="4" cols="20">  
<textarea name="name" rows="10" cols="50" [disabled][readonly]>  
    Default Text  
</textarea>
```

```
<textarea name="address" rows=5 cols=10>  
    Please write your address</textarea>
```

Demo**➤ FormExamples.html**

```
<html><head><title>form examples</title></head>
<body bgcolor="pink">
<form name="form1" action="store.html" method="post">
<p>
<strong>Enter first name</strong>: <input name="username">
  &nbsp;&nbsp;&nbsp;
<strong>Enter lastname</strong>: &nbsp;
<input maxlength="30" name="surname"></p>
<p><strong>Enter address:</strong>&nbsp;&nbsp;&nbsp;&nbsp;
<textarea name="addr" rows="3"></textarea>
<br><br>
<strong>Select the training programs attended:</strong>
<input type="checkbox" value="internet/html" name="internet-html">
Internet/HTML
<input type="checkbox" checked value="c programming"
name="c-programming"> C Programming
<input type="checkbox" value="dbms-sql" name="dbms-sql"> DBMS-SQL
</p> <p><strong>Select the stream you belong to:</strong>
<input type="radio" value="science" name="s-grp"> Science
<input type="radio" value="arts" name="s-grp"> Arts
<input type="radio" value="commerce" name="s-grp"> Commerce
<input type="radio" value="oth2" name="s-grp"> Engineering </p>
<strong>Which training program would you like to attend ?</strong>
<select multiple size="3" name="pref">
```

```
<option value="ih" selected>Internet-HTML
    <option value="js">Javascript
    <option value="vbs">VBscript
    <option value="as">ASP
    <option value="xm">XML
    <option value="jv">JAVA
    <option value="jsp">jsp</option>
</select>
<br><br>
<strong>Select the location of your resume</strong>
<input type="file" size="15" name="fnm"> <br>
<input type="hidden" value="patni" name="coname"> <br>
<input type="button" value="exit" name="but">
<input type="submit" value="save">
<input type="reset" value="reset">
</p>
</form>
</body>
</html>
```



Demo**➤ FormSubmission.html**

- 9 -

```
<HTML>
<HEAD>
<TITLE>Form Submission</TITLE>
</HEAD>
<BODY>
<form method="post" action="store.html">
<P>Enter UserID:<INPUT id=text1 name=text1></P>
<P>Enter Pasword:
<INPUT id=password1 type=password name=password1></P>
<P>
<INPUT id=submit1 type=submit value=Submit name=submit1> &nbsp;
<INPUT id=reset1 type=reset value=Reset name=reset1></P>
</form>
</BODY>
</HTML>
```

Lab

➤ Lab 6



- 10 -

Summary-



➤ **After completing this module you now:**

- Understand the role of forms in web pages
- Understand various HTML elements used in forms like
 - Single line text field
 - Text area
 - Check box
 - Radio buttons
 - Password fields
 - Pull-down menus
 - File selector dialog box



+ 11 +



Review Question



- **Question 1: Radio Buttons are used when you want the user to select:**
 - one of a limited number of choices.
 - one or more options of a limited number of choices.
 - many of unlimited number of choices.

- **Question 2: METHOD attribute states the method to use when you send the form to the server.**
 - True/ False

- **Question 3: The _____ attribute of form specifies how the data is to be encoded.**



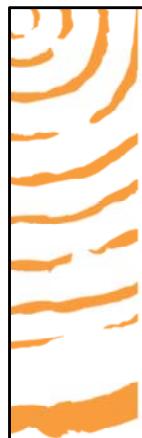
- 12 -

Review Question: Match the Following

• Check Boxes	• Populate data in the list.
• Radio Buttons	• Create a select list.
• Password fields	• Include form data without rendering it to the user.
• Hidden Input Type	• Entered text is shown as asterisks.
• <select> tag in Drop Down List	• Only one option of a limited number of choices.
• <option> tag in Drop Down List	• One or more options of a limited number of choices



Add the notes here.



Web Basics - HTML

Lesson 8. Frames

Lesson Objectives



➤ **After completing this module you will be able to:**

- Understand the need for frames in web pages.
- Learn to create and work with frames.



- 2 -

Frames: With frames, you can display more than one HTML documents in the same browser window. Each HTML document is called a frame, and each frame is independent of the others.

Disadvantage of using frames is that the web developer must keep track of more HTML documents. Also, it is difficult to print the entire page.

8.1: Frames

Working with Frames



➤ **Topics covered in this module are:**

- Frame Basics
- Frameset Tag Attributes
- Frame Tag Attributes
- Nested Frames
- Noframes Tag

- 3 -



8.2: Frames Basics

Frame Basics (Code)



- <frameset> tag replaces the body tag.

```
<html>
  <head> <title>Multi Frame Page</title> </head>
  <frameset>
    Other FRAMESETS or FRAME tags or NOFRAME
    tag
  </frameset>
</html>
```



- 4 -

Include Frames in a Web Page

Frames are created through the use of frame documents, which in turn are

created with <FRAMESET> and <FRAME> tags.

The <frameset> tag defines how to divide the window into frames.

Each frameset defines a set of rows **or** columns.

Values of rows or columns indicate the amount of screen area each row or column occupies.

The frameset element states only how many columns or rows there will be in the frameset.

Important: You cannot use <body></body> tags with <frameset></frameset> tags. However, if you add a <noframes> tag containing some text for browsers that do not support frames, you will need to enclose the text in <body></body> tags.

8.2: Frames Basics

Frameset Tag Attributes (Code)



➤ **<frameset> tag has two attributes:**

- Rows: The number of rows of the screen allocated to each frame
- Cols: The number of columns of the screen in which each frame in the document will be allocated

```
<frameset rows="50%, 25%, 25%">  
<frameset rows="150, *">  
<frameset cols="40%, *">  
<frameset cols="450, *, 100">
```

- 5 -



<FRAMESET>....</FRAMESET> tags are used as the container of the frame document and to define the frame size. The tag has two attributes, ROWS and COLS. ROWS describes the numbers of screen rows allocated to each frame. COLS describe the number of screen columns in which each frame in the document is allocated. These numbers can be absolute numbers or percentage value, or you can use a* to indicate that the corresponding frame should receive all remaining space.

<frameset rows="50%, 25%, 25%">

The above code creates three frames. The first frame is twice as long as the other two frames.

<frameset rows=150, *>

The above code creates two frames. The first has the fixed column width of 150 pixels. The second receives the remaining part of the screen.

It is possible to nest <FRAMESET>....</FRAMESET> tags inside other <FRAMESET>....</FRAMESET> tags to create all sorts of effects.

Example:

```
<html>  
<frameset cols="25%,50%,25%">  
  <frame src="frame_a.htm" />  
  <frame src="frame_b.htm" />  
  <frame src="frame_c.htm" />  
</frameset></html>
```

8.2: Frames Basics

Frame Tag Attributes



➤ <Frame>

- Second tag used to create frames.
- Defines a frame within a frameset.
- Attributes of <frame> tag are:
 - src="url"
 - name="window_name"
 - marginwidth="value"
 - marginheight="value"
 - scrolling="yes/no/auto"
 - [noresize]

- 6 -



Frame :

<frame> tag defines which HTML document to put into each frame.

If a frame has visible borders, the user can resize it by dragging the border. To prevent a user from doing this, you can add noresize="noresize" to the <frame> tag.

Second tag used to create frames is the <FRAME> tag. It defines a frame within a frameset and has six possible attributes. <FRAME> tag syntax is:

**<frame src="url" name="window_name" marginwidth="value"
marginheight="value"
scrolling="yes/no/auto" {noresize}>**

SRC is the URL of the source document to be displayed in this frame.

NAME is an optional attribute that assigns a name to a frame so that it can be used by links in other documents or by JavaScripts.

MARGINWIDTH is an optional attributes used to set the width of the margins of the frame. The value is in pixels.

MARGINHEIGHT is the same as MARGINWIDTH, except that it controls the upper and lower margins of the frame.

SCROLLING is used to describe whether the frame should have a scrollbar. If you set this attribute to “yes”, it means the frame always has a scrollbar. If you set this attribute to “no”, it means that the frame never has a scrollbar. Use of the auto setting displays a scrollbar when needed. This is an optional attribute with “auto” being the default.

NORESIZE is an optional attribute. If this attribute appears in the tag, the user cannot change the size of the frame. By default, all frames can be resized.

Finally, <NOFRAME>.....</NOFRAME> tags can be used in conjunction with the other frame tags. Use this tag to provide alternative information to browsers that are non-frame-capable clients. A frame-capable client ignores all items found inside the <NOFRAME> tags.

8.2: Frames Basics

Frame Tag Attributes (Contd...) (Code)

```
<frame src="c:/html/index.htm" name="index"  
scrolling="no" noresize>
```

- Target attribute value of the <a> tag becomes the value of name.



- 8 -

The `<frame>` tag defines one particular window (frame) within a frameset. Each frame in a frameset can have different attributes, such as border, scrolling, the ability to resize, etc. Example:

```
<html>
<head>
<title>frames</title>
</head>
<frameset rows=10%,*>
<frame border=0 name=top src="top.html" noresize scrolling=no>
<frameset cols=20%,*>
<frame name=left src="Left.html">
<frameset rows=70%,*>
<frame name=main src="Main.html">
<frame name=details src="about:blank">
</frameset>
</frameset>
</frameset></html>
```

There is a very simple header document, `header.htm`. Document `picture.htm` contains graphics you wish to show in the second (left side) frame. `Facts.html` contains the information you wish to show in the third (right side) frame.

```
<frameset rows="70,*">
```

This tag splits the screen into two horizontal sections. The first section is 70 rows long. The remaining section occupies the rest of the screen.

Add the first frame with the `<frame>` tag:

```
<frame src="header.htm" noresize>
```

This tag causes the document `header.htm` to display in the first frame, the one designated to be 70 rows long. The user cannot resize this frame.

Add a second frameset that is nested into the first frameset. This is done with `<frameset cols="50%,*"/>`. This tag creates another frameset in the area of the screen not occupied by the `header.htm` frame. This new frameset consists of two frames of equal size.

Add the second and third documents to their respective frames. Refer the following html code:

```
<frame src="picture.htm" marginheight=0 marginwidth=4
      scrolling=no name="toon">
<frame src="facts.htm" name="story" noresize>
Close the two framesets.
</frameset>
</frameset>
```

8.3: Nested Frames

Nested Frames (Code)



- Nesting of <frameset> is possible:

```
<frameset rows="20%,*>
  <frame src="header.htm"></frame>
  <frameset cols="40%,*>
    <frame src="index.htm"></frame>
    <frame src="details.htm"></frame>
  </frameset>
</frameset>
```

- 10 -



Nested Frames:

Nesting of <frameset> is possible. Framesets may be nested to any level. A frameset is "nested" if it is used in place of a frame declaration in an outer frameset. The frame, divided by the nested frameset, is called the **parent** frame.

8.4: Noframes Tag

Noframes Tag



➤ <noframes>:

- Your browser does not support frames </noframes>
- Can be used in conjunction with the other frame tags
- Provide alternative information to browsers that are non-frame-capable clients
 - A frame-capable client ignores all items found inside the <noframes> tags.

- 11 -



Noframes:

The <noframes> tag is used for browsers that do not handle frames. It's element can contain all the elements that you find inside the body element of a normal HTML page. Noframes is used widely to link to a non-frameset version of the web site or to display a message to users that frames are required. Noframes element goes inside the frameset element.

A simple three-framed page, with a <noframes> tag:

```
<html>
<frameset cols="25%,50%,25%">
  <frame src="frame_a.htm" />
  <frame src="frame_b.htm" />
  <frame src="frame_c.htm" />
<noframes>
  Sorry, your browser does not handle frames!
</noframes>
</frameset>
</html>
```

USE ONE FRAME TO INDEX ANOTHER THAT WILL DISPLAY THE CONTENTS

Use Frameset to:

- Create an index or a table of contents that gives users a choice of options and then updates or displays whatever the user choose in another corresponding window
- Split the screen into two separate frames.
 - Create one frame that contains the specified information and label it with the NAME attribute in the <FRAME> tag.
 - Create another frame that contains links to the first frame. Make sure this information loads in the first frame, specify the frame's name in the <A HREF> tag under TARGET attribute.

k1

Demo



➤ frames.html



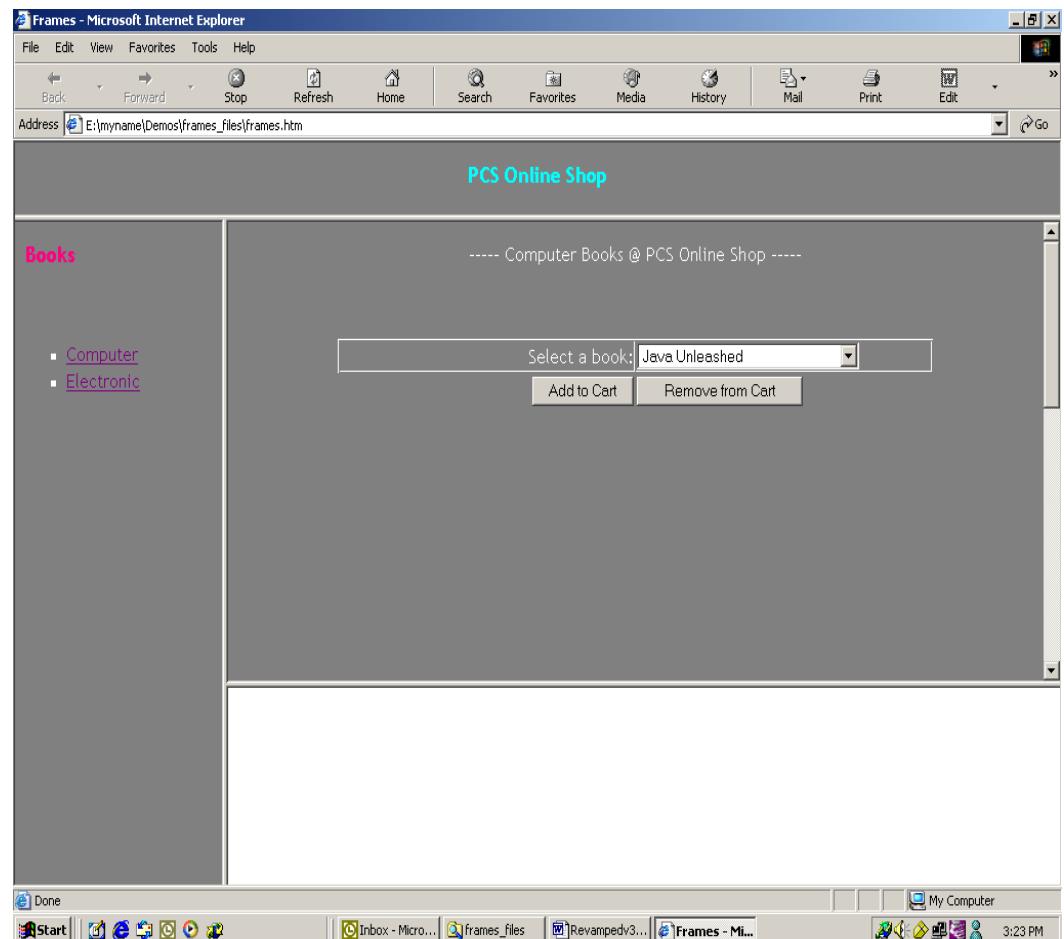
- 13 -



```
<HTML>
<HEAD><TITLE>Frames</TITLE></HEAD>
<!—Dividing the Web document into three parts using frameset
First frameset divides the browser horizontally into two parts
First half contain the Top. Html page
Second half again gets divided vertically into two parts 'A' Part
showing the 'left.htm' page and 'B' part further get divided horizontally
into two parts x part showing the 'main.htm' and y part showing 'blank' -->
<FRAMESET rows=10%,*>
    <FRAME border=0 name=top src="top.htm" noResize
scrolling=no>
    <FRAMESET cols=20%,*>
        <FRAME name=left src="left.htm">
    <FRAMESET rows=70%,*>
        <FRAME name=main src="main.htm">
        <FRAME name=details src="about:blank">
    </FRAMESET>
</FRAMESET>
</FRAMESET>
</HTML>
```

Slide 13

k1 Did not see comments enough to explain the code. Only target comment is repeated in all the three pages
kunchuva, 21-07-2009



This is the PCS online website. It is divided into *four* frame with help of *three* framesets. Topmost frame displays the Heading of web site "PCS Online Shop". Left-most frame has two radio buttons, *Computer* and *Electronic*.

When you click the *Computer* radio button it opens the 'computer books @PCS online shop page' in right-most window. Similarly, when you click the *Electronic* radio button, it opens "Electronic books @PCS online shop page" in right-most window.

In the right-most window, after you select books from the drop-down list and click 'Add to Cart' a message displays in the last frame "Data get update on server".

```
<HTML><HEAD><TITLE></TITLE>
</HEAD>
<BODY text=#ffffff bgColor=gray>
<!--This page act as Header for frames.html page displaying
heading 'PCS Online Shop' -->
<P align=center><FONT color=#00ffff size=4><STRONG>
PCS Online Shop</STRONG></FONT></P></BODY></HTML>
```

Top.html

```
<HTML><HEAD><TITLE></TITLE>
</HEAD>
<BODY text=#ffffff bgColor=gray>
<P align=left><FONT color=#ff0080 size=4><B>Books</B></FONT> </P>
<P align=left>&nbsp;</P>
<UL type=square>
<!-- This page appears on left side in the frames.html ,which
contains two link 1. Computer and 2.Electronic. Both have target
attribute which refers to two different parts of the same page, tells
the browser to open the specified html file in frame whose name is
Main in multi frame web page-->
<LI><A href="main.htm#cbooks" target=main>Computer</A> </LI>
<LI><A href="main.htm#ebooks" target=main>Electronic</A> </LI>
</UL></BODY></HTML>
```

left.html

```
<HTML><HEAD><TITLE></TITLE>
</HEAD>
<BODY text=#ffffff bgColor=#808080>
<!-- This page having two different forms which is separated
with help of <div> and <a> tag. So when ever user click on 'computer'
link on Left.htm page it will open the form under the link name cbook.
Similarly when ever user clicks on 'Electronic' link on Left.htm page
it will open the form under the link name ebook-->
<!-- Link with name =cBook-->

<P align=center><A name=cBooks>
----- Computer Books @ PCS Online Shop
-----</A></P>
<P>&nbsp;</P>
<!-- Form design-->
<FORM name=frmbooks action="store.html" method=get target=details>
<DIV align=center>
<CENTER>
```

```
<TABLE cellSpacing=1 cellPadding=0 width="75%" border=1>
<TBODY>
<TR>
<TD width="50%">
<P align=right>Select a book:</P></TD>
<TD width="50%">
<!-- Drop down list for selecting any computer books-->
<SELECT size=1 name=cBook>
<OPTION selected>Java Unleashed</OPTION>
<OPTION>Java by Example</OPTION>
<OPTION>Java by Reference</OPTION>
<OPTION>Java - The Complete Reference</OPTION>
</SELECT>
</TD></TR></TBODY></TABLE></CENTER></DIV>
<DIV align=center>
<CENTER>
<TABLE cellSpacing=1 width="75%" border=0>
<TBODY>
<TR>
<TD align=middle width="50%"> <P align=right>
<!-- After clicking on submit button, the data selected in drop down
list is transfer to store.html page with help of get method-->
<INPUT type=submit align=right value="Add to Cart" name=bcBook>
</P></TD>
<TD align=middle width="50%"> <P align=left>
<INPUT type=submit align=left value="Remove from Cart" name=bcBook>
</P></TD></TR>
</TBODY></TABLE></CENTER></DIV></FORM>

<P align=center>&nbsp;</P>
<!-- Link with name =eBook-->
<P align=center><A name=eBooks>----- Electronic Books
@ PCS Online Shop
-----</A></P>
<P align=center>&nbsp;</P>
<FORM name=frmbooks action="store.html" method=get target=details>
<DIV align=center>
<CENTER>
```

```
<TABLE cellSpacing=1 cellPadding=0 width="75%" border=1>
<TBODY>
<TR>
<TD width="50%">
<P align=right>Select a book:</P></TD>
<TD width="50%">
<!-- Drop down list for selecting any electronic books-->
<SELECT size=1 name=eBook>
<OPTION selected>Processor Operations</OPTION>
<OPTION>Digital Electronics</OPTION>
<OPTION>Microprocessor made simple</OPTION>
</SELECT></TD></TR></TBODY></TABLE></CENTER></DIV>
<DIV align=center>
<CENTER>
<TABLE cellSpacing=1 width="75%" border=0>
<TBODY>
<TR>
<TD align=middle width="50%"> <P align=right>
<!-- After clicking on submit button, the data selected in drop down listed
is transfer to store.html page with help of get method-->
<INPUT type=submit value="Add to Cart" name=beBook></P></TD>
<TD align=middle width="50%"> <P align=left>
<INPUT type=submit value="Remove from Cart" name=beBook>
</P></TD></TR></TBODY></TABLE></CENTER></DIV></FORM>
<P align=center>&nbsp;</P>
<P align=center>&nbsp;</P>
<P align=center>&nbsp;</P>
<P align=center>&nbsp;</P>
<P align=center>&nbsp;</P>
<P align=center>&nbsp;</P></BODY></HTML>
```

main.html

Lab

➤ Lab 7



LEARN
iGATE Learning, Education And Research Note

- 18 -

Lesson Summary

➤ **After completing this module you now:**

- Frameset helps to open multiple html pages in single web page with the help of frames
- Frames' target attribute redirect (open) the page at specific



- 19 -

Review Question



- **Question 1: Which tags create frames?**
 - <FRAME>
 - Both of the above
 - <FRAMESET>

- **Question 2: Values of the rows or columns indicate the amount of screen area each row or column will occupy.**
 - True/ False?

- **Question 3: _____ attribute appears in the tag, the user will not be able to change the size of the frame.**



- 20 -

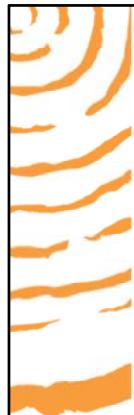
Review – Match the Following



• SRC	• Controls the upper and lower margins of the frame.
• NAME	• Sets the width of the margins of the frame.
• MARGINWIDTH	• Decides that the frame should have a scrollbar.
• MARGINHEIGHT	• Size of the frame cannot be changed.
• SCROLLING	• Assigns a name to a frame.
• NORESIZE	• Source document's URL is displayed in this frame.

- 21 -





Web Basics - HTML

Lesson 9: XHTML



Lesson Objectives



- After completing this module you will be able to:
- Work with XHTML 1.0
 - Understand DTD's for XHTML
 - Understand XHTML Namespaces
 - Validating XHTML with W3C Standards



- 2 -



What is XHTML?



- **XHTML is a stricter and cleaner version of HTML**
- **XHTML consists of all the elements in HTML 4.01 combined with the syntax of XML**



- 3 -

When people are new to XML, they compare XML and HTML. HTML is a SGML application. XML is a subset of SGML. It is more compact and language that has benefits over SGML. So it makes sense that HTML should be based on XML instead of SGML.

It is not possible to immediately convert the web to an XML-based version of HTML. We can take a step to move towards a more structured future in terms of HTML. The W3C is leading the way with XHTML, a version of HTML that has been reformulated as an application of XML. XHTML is similar in many ways to HTML 4.0 and it represents the future of HTML.

In a nutshell:

- XHTML is a combination of HTML and XML
- XHTML consists of all the elements in HTML 4.01 combined with the syntax of XML

Why XHTML?



- Traditional HTML does not insist for well formed HTML documents
- Applying styles to poorly structured HTML is a cumbersome process
- Latest browsers and internet devices will process the poorly structured HTML documents haphazardly
- XHTML documents are based on the XML syntax, which can be served as text/html
- Easy to process XHTML documents



Many pages on the WWW contain “bad” HTML

Following HTML code will work fine if you view it in a browser
However, it does not follow the HTML rules:

XML has to be “well-formed” document XML was designed to
describe data and HTML was designed to display data

Currently, the web is based on HTML technology that has very little structure
Poor coding, browser leniency, and proprietary browser extensions have
combined to create a web that is extremely unstructured This is not a good
thing

HTML will always have a place on the web because it is too deeply ingrained to
replace Besides even though XML paired with CSS/XSL has huge structural
benefits over a purely presentational HTML web page, it does involve more
work There are certainly situations in which it does not matter whether or not
content is separated from how it is displayed, in which case HTML represents a
more efficient solution

One reason XML is viewed as a beneficial direction for HTML is that new types
of browsers are being developed for Internet devices that will not have the
processing power to deal with poorly structured HTML documents These non-
desktop internet devices will rely on highly structured HTML documents so that
their processing overhead will be minimal. Browsers will still have to support
the old, unstructured version of HTML for the foreseeable future Consider the
following benefits associated with XHTML, which are mentioned in the HTML
W3C working draft

XHTML documents are based on the XML syntax, which
means

that they can be viewed, edited, and validated with standard XML
tools

XHTML documents can be served as media type text/html.

XHTML DTD



- DTD specifies the syntax of a web page in SGML
- DTD is used to specify rules that apply to the markup
- XHTML is specified in an SGML document type definition or “DTD”
- XHTML DTD describes in precise, computer-readable language the allowed syntax and grammar of XHTML markup

- 5 -



XHTML DTD



- **W3C has developed three DTDs, which are all included in the XHTML specification**
- **The three XHTML DTDs are as follows:**
 - Transitional: HTML presentation elements are available for formatting documents
 - Frameset: Frames as well as HTML presentation elements are available
 - Strict: No HTML presentation elements are available (font, table, etc) CSS/XSL must be used to format documents for display



- 6 -

XHTML Namespaces



- The `xmlns` namespace declaration attribute is used to declare an XHTML namespace in the HTML element
- The namespace must match the DTD specified in the document type declaration
- Following are the three XHTML namespaces that are associated with the three DTDs mentioned earlier:
 - Strict <http://www.w3.org/TR/xhtml1/strict>
 - Transitional <http://www.w3.org/TR/xhtml1/transitional>
 - Frameset <http://www.w3.org/TR/xhtml1/frameset>



- 7 -

Creating XHTML Document



- Change your DOCTYPE to XHTML 10 Strict
`<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 10 Strict//EN"
"http://www3.org/TR/xhtml/DTD/xhtml1-strict.dtd">`
- Add the xmlns, lang and xml:lang attributes to your <html> element
`<html xmlns="http://www.w3.org/1999/xhtml" lang="en"
xml:lang="en">`



- 8 -

Sample XHTML document



➤ Code Snippet

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 10
Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="en"
xml:lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
<title>Skeletal XHTML Documents</title>
```



- 9 -

Sample XHTML document



➤ Code Snippet

```
</head>
<body>
<p>
This is a skeletal XHTML document
</p>
</body>
</html>
```

- 10 -



Block level element



- Block elements separate content into blocks
- Each block element is displayed on its own, as if it has a line break before and after it
- <p>,<div>, <blockquote>, , , and are all block elements

- 11 -



Inline element



- **Inline elements wont separate content into blocks**
- **Inline element is just displayed in the flow of the paragraph**
- **,<i>,<u>,, and <a> are all inline elements**

- 12 -



Rules



- All inline elements and text need to be inside another block element before they can go in the <body> element
- Block elements are not allowed inside an inline element
- Keep block elements out of your <p> element
- Put text and inline elements inside block elements before adding them to a <blockquote>
- and can have only block level elements
- We can put text, inline elements, or block elements inside



- 13 -

XHTML Checklist



- **Documents must be well-formed**
- **Element and attribute names must be in lowercase**
- **Every Starting tag should contain an ending tag**
- **Attribute values must be always quoted**
- **Attribute names cannot be minimized, that is used without a value**
- **The head and body elements cannot be omitted**
- **The title elements must be the first element in the head element**



- 14 -

XHTML Checklist



- All script and style elements must be enclosed within CDATA section
- Should not use “&” in the content of html “&” is used for starting entities so use & instead
- Block level and inline elements should be handled properly

- 15 -



W3C validator



- The W3C validator is a free online service that checks pages for compliance with standards
- Use the validator to ensure that your HTML is well formed and that your elements and attributes meet the standards
- It ensures cross browser compatibility
- W3C validator URL is <http://validator.w3.org>

- 16 -



Summary



- Work with XHTML documents
- Rules need to follow to create XHTML document
- Create XHTML with W3C Standards



- 17 -



Review Question



- **Question 1:** In XHTML all the elements should be in _____
- **Question 2:** _____ is used to check to check XHTML Syntax
- **Question 3:** _____ website is used to Check the markup (HTML, XHTML, ...) of Web documents



- 18 -



Web Basics - HTML

Lesson 10. Cascading Style Sheets



Lesson Objectives



➤ After completing this module you will be able to:

- Understand the use of CSS in web pages
- Learn to use CSS in different modes



- 2 -

10.1: Cascading Style Sheet Basics

What is CSS?



- CSS stands for Cascading Style Sheets
- Styles define how to display HTML elements
- Styles are normally stored in style sheets
- Styles were added to HTML 4.0 to solve a problem
- External Style Sheets save you a lot of work
- External Style Sheets are stored in CSS files
- Multiple style definitions cascade into one

- 3 -



10.1: Cascading Style Sheet Basics

Why CSS?



➤ **Solves common problem:**

- Separate document presentation from the web page content.

➤ **Save lots of work:**

- Allows developers to control the style and layout of multiple Web pages all at once.



Why use CSS?

Styles solve a common problem. HTML tags were originally designed to define the document content. They were supposed to say "This is a header", "This is a paragraph", "This is a table", by using tags like `<h1>`, `<p>`, `<table>`, and so on. Browser was to take care of the layout of the document without using any formatting tags.

Two major browsers - Netscape and Internet Explorer - continued to add new HTML tags and attributes (like the `` tag and the `color` attribute) to the original HTML specification. Subsequently, it became more difficult to create HTML documents with content clearly separate from the presentation layout.

To solve this problem, W3C, the non-profit, standard setting consortium responsible for standardizing HTML, created STYLES in addition to HTML 4.0. Both Netscape 4.0 and Internet Explorer 4.0 support Cascading Style Sheets.

Style Sheets Save a Lot of Work

Styles in HTML 4.0 define how HTML elements are displayed, just like the *font* tag and the *color* attribute in HTML 3.2. Styles are saved in files external to your HTML documents. External style sheets allow you to change the appearance and layout of all pages in your website. Simply, edit a single CSS document. If you have ever had to change the font or color of all the headings in all your Web pages, you will understand how CSS can save you a lot of work.

CSS is a breakthrough in Web design because it allows developers to control the style and layout of multiple Web pages all at once. As a Web developer you can define a style for each HTML element and apply it to as many Web pages as you want. To make a global change, simply change the style, and all elements in the Web are updated automatically.

10.1: Cascading Style Sheet Basics

Cascading Style Sheet

- Unlike DHTML and DOM, cascading style sheets exists as a real and accepted standard on the W3C
- Standards are referred to as Cascading Style Sheets 1 (CSS1) and Cascading Style Sheets 2 (CSS2)
- We use style sheets to control HTML tag formatting

- 6 -



Cascading Style Sheet:

It is a standard to specify the appearance of text and other elements. CSS was developed for use with HTML in Web pages. It is also used in other situations, notably in applications built using XPFE. CSS is typically used to provide a single "library" of styles that are reused throughout a large number of related documents, as in a web site. A CSS file might specify that all numbered lists are to appear in italics. By changing that single specification, the look of a large number of documents can be easily changed.

What is a Style Sheet?

Style sheets define how to display HTML elements.

Style sheets (SS) provide a means for web authors to separate the appearance of web pages from the content.

Style sheets are an accepted standard on the W3C. The standards are referred to as Cascading Style Sheets 1 (CSS1) and Cascading Style Sheets 2 (CSS2).

10.1: Cascading Style Sheet Basics

Cascading Style Sheet (Contd...)



➤ Cascading:

- Use multiple styles in an individual HTML page.
- Browser follows orders (In which it is specified) to interpret the information.
- Designer uses all three: Link Style sheet, Embedded style sheet, External style sheet.



The word cascading means that multiple styles can be used in an individual HTML page. Browser follows an order called a cascade to interpret the information.

10.2: Types of Cascading Style Sheets

Types of CSS

- **Embedded Style Sheets**
- **Linked Style Sheets**
- **Inline Style Sheets**
- **Using Classes**

- 8 -



Inline: Style sheet definition only applies to the tag contents that contain it. It is used to control a single tag element. Each tag does not need to have its style defined as it inherits from its parent.

Embedded: Embedded style sheets are placed within HTML code of the page they are to be applied to. Style sheet syntax comes between opening and closing <STYLE> tags. These tags are placed either in the <HEAD> section or between the </HEAD> and <BODY> tags.

Linked: Linked style sheets exist as separate files that are linked to a page with the <LINK> tag. They have the css extension and are referenced with a URL. Inside the css file, style attributes are contained within opening and closing <STYLE> tags. Placing a single <LINK> tag within the <HEAD> tags links the page that needs these styles.

Classes: Another method of defining which tag gets a style sheet definition is by using classes. Style sheet classes are defined by placing a period before a unique name. This can come after a tag or by itself as in the following example:

```
H1.myClass {color: blue}  
.myOtherClass {color: red}
```

10.2: Types of Cascading Style Sheets

CSS Precedence

➤ Rules apply to style sheets' order of precedence:

- Browser determines default format.
- How close the style definition is to the tag.
- More specific tags.

+ 9 +



Style Sheet Precedence

There are several rules that apply to the order of precedence of style sheets. All tags have a default format determined by the browser. This is what you see if no style sheet attributes are set. This also represents the lowest priority.

Another level of priority is established by how close the style definition is to the tag. For this order, linked style sheets are lower than embedded style sheets, which are lower than inline style sheets. If you accidentally include the same property in a linked style sheet as in inline style sheet, then the priority goes to the definition closest to the tag, which would be inline style.

Style sheets for more specific tags have priority over general tags. For example, if a Web page marks the <BODY> tag with a certain style sheet definition and an <H3> tag with same property and a different value, then the <H3> tag has the priority, even though it is also part of the body.

Several rules apply to the order of precedence of style sheets:

All tags have a *default* format determined by the browser.

This is what you see if *no* style sheet attributes are set.

This also represents the *lowest* priority.

Another level of priority is established by how *close* the style definition is to the tag. For this order, linked style sheets are:

Lower than **embedded** style sheets, which are **lower than** inline style sheets.

Style sheets for more specific tags have priority over general tags.

10.2: Types of Cascading Style Sheets

Embedded Style Sheet Syntax

➤ Embedded Style Sheet:

- Set of style definitions placed within <STYLE> tags.
- Located between the head and body of the document.

- 10 -



Embedded Style Sheet:

An embedded style sheet is a set of style definitions placed within <STYLE> tags and located between the **head** and **body** of the document. It sets the style attributes for the entire page where it is located.

Following style sheet description applies to the <H5> tag. It sets the font face to be either Arial, Impact, or Sans Serif, depending on which one it finds first on the user's system. Text color is also defined as *blue*.

H5 {font-family: Arial, Impact, Sans Serif; color: blue}

You can also group tags together by separating them with commas:

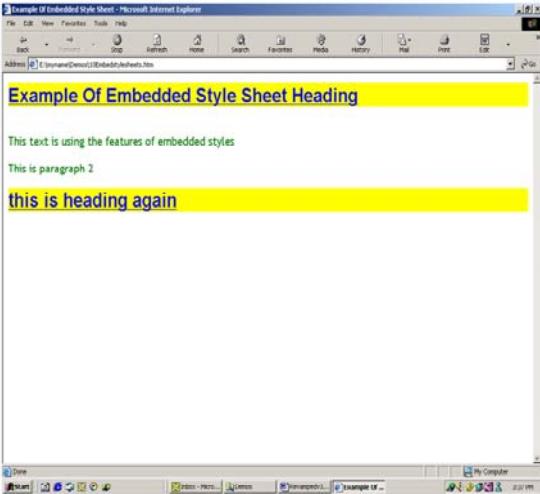
H1, H2, H3 {font-family: Arial, Impact, Sans Serif; color: blue}<html>

```
<head>
<title>Embedded Style Sheet</title>
</head>
<style>  body {background: black; color:green}
h1 {background: orange; font-family: Arial, Impact, Sans serif;
color: blue; font-size:30pt; text-align: center}
h2, h3 {background: gold; font-family: Arial, Impact, Sans Serif; color:red}
</style> <body>
<h2>This is Level 2 Heading, with style</h2>
<h1>This is Level 1 Heading, with style</h1>
<h3>This is Level 3 Heading, with style</h3>
<h4>This is Level 4 Heading, without style</h4>
</body>
</html>
```

10.2: Types of Cascading Style Sheets

Demo

➤ **Embedstylesheets.htm**



The screenshot shows the Microsoft Internet Explorer browser displaying a web page titled "Example Of Embedded Style Sheet". The page contains several examples of embedded CSS. The first heading is styled with a yellow background and blue text. The following two paragraphs are plain black text. The second heading is also styled with a yellow background and blue text. The browser's toolbar and address bar are visible at the top, and the taskbar at the bottom shows other open windows.

- 11 -



iLEARN
iGATE Learning, Education And Research Network

```
<html>
<head><title>
Example Of Embedded Style Sheet
</title>
</head>
<style>
h1 {text-decoration: underline;
    font-family: arial;
    background: yellow;
    color: blue
    }
p { font-size: 14pt; color: green}
</style>
<body>
<h1>Example Of Embedded Style Sheet Heading</h1>
<br>
<p>This text is using the features of embedded styles</p>
<p>This is paragraph 2</p>
<h1>this is heading again</h1>
</body></html>
```

10.2: Types of Cascading Style Sheets

Link Style Sheet



➤ Link Style Sheet:

- All style definition are store in one file (.css file)
- This file gets called by the HTML file during page loading
- Link .css and HTMI file using <LINK> tag. Specify in the <style> tag in header

```
<link rel="stylesheet" href="test.css" type="text/css">
```



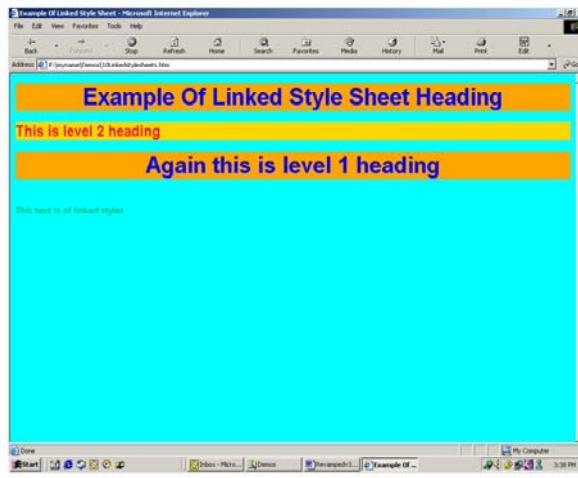
- 12 -

Same as embedded style sheet. The only difference is that the separate css file contains all styles, and gets called by the HTML file.

```
<html>
<head>
<title>Linked Style Sheet</title>
<link rel="stylesheet" href="linked_ex2.css" type="text/css">
</head>
<body>
<h2>This is Level 2 Heading, with style</h2>
<h1>This is Level 1 Heading, with style</h1>
<h3>This is Level 3 Heading, with style</h3>
<h4>This is Level 4 Heading, without style</h4>
</body></html>
```

Run code **linked_ex2.css**:

```
body {background: black; color:green}
h1 {background: orange; font-family: Arial, Impact, Sans
    serif; color: blue; font-size:30pt; text-align: center}
h2, h3 {background: gold; font-family: Arial, Impact, Sans
    serif; color:red}
```

10.2: Types of Cascading Style Sheets**Demo****➤ Linkedstylesheets.htm**

```
<html>
<head>
<title>Example Of Linked Style Sheet</title>
<link rel="stylesheet" type="text/css" href="linked_ex2.css">
</head>
<body>
<h1>Example Of Linked Style Sheet Heading</h1>
<h2>This is level 2 heading</h2>
<h1>Again this is level 1 heading</h1>
<br>
<p>This text is of linked styles
</p>
</body>
</html>
```

10.2: Types of Cascading Style Sheets

Inline Style Sheet (Code)

Inline Style Sheets:

- All style attribute are specified in the tag it self.
- It gives desired effect on that tag only. It does not affect any other HTML tag.

```
<p style="font-size: 14pt" id="p2">  
<body bgcolor="black" style="font-family: Arial; color: blue">
```

- 14 -

 LEARN
iGATE Learning, Education And Research Network

Inline Style Sheet:

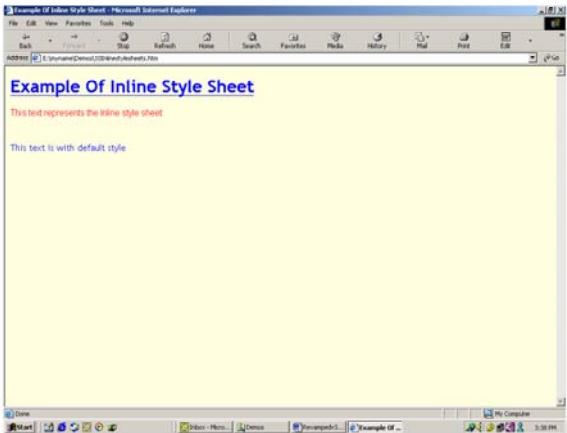
Definitions appear next to other tag attributes. You need to remember to place the style sheet description within quotes, like the following:

```
<html>  
<head><title>Inline Style Sheet</title></head>  
<body style="background: white; color:green">  
<h2 style="background: gold; font-family: Arial, Impact, Sans Serif;  
color:red">  
This is Level 2 Heading, with style</h2>  
<h1 style="background: orange; font-family: Arial, Impact, Sans serif;  
color: blue;font-size:30pt; text-align: center">  
This is Level 1 Heading, with style</h1>  
<h3 style="background: gold; font-family: Arial, Impact, Sans Serif;  
color:red">  
This is Level 3 Heading, with style</h3>  
<h4>This is Level 4 Heading, without style</h4>  
<h1>This is again Level 1 heading with default styles</h1>  
</body>  
</html>
```

10.2: Types of Cascading Style Sheets

Demo

➤ **Inlinestylesheets.htm**



iGATE
ITOPS for Business Outcomes

- 15 -



```
<html>
<head>
<title>
Example Of Inline Style Sheet
</title>
<body bgcolor="lightyellow" text="blue">
<h1 style="text-decoration: underline">
Example Of Inline Style Sheet</h1>
<p style="font-family: arial, impact, sans serif; color:red">
This text represents the Inline style sheet
</p>
<br>
<p>This text is with default style</p>
</body>
</html>
```

10.3: Advanced Style Sheet Features

Style Sheet Classes

- Definition is by using classes
- Defined by placing a period before a unique name
- Comes after a tag or by itself:
`H1.myClass {color: blue}.myOtherClass { color: red}`
- Style defined by the class name is used for the tag whose CLASS attribute it matches

- 16 -



USING CLASSES

Styles defined by the class name are used for the tag whose CLASS attribute it matches. An `<H1>` tag with the class attribute set to `myClass` would be blue, and any tag with the CLASS attribute set to `myOtherClass` would be red as follows:

```
<H1 CLASS="myClass">This text would be blue.</H1>
<P CLASS="myOtherClass">The text would be red.</P>
```

10.3: Advanced Style Sheet Features

Style Sheet Classes (Contd...) (Code)**➤ Code Snippet**

```
<style>
H1.myClass {color: blue}
.myOtherClass { color: red; text-align:center}
</style>
```

- 17 -



10.3: Advanced Style Sheet Features

Style Sheet Classes (Contd...)**➤ Code Snippet**

```
<body style="background: white; color:green">
<H1 class="myClass">This text would be blue</H1>
<p class="myOtherClass">The text would be in red</P>
<H3 class="myOtherClass">This is level 2 heading</H3>
<table class=myotherClass border width=100%>
<caption>Sample Table</caption>
<td>Data 1</td><td>Data 2</td>
</table>
```

- 18 -



10.3: Advanced Style Sheet Features

Style Sheet Classes (Contd...)**➤ Syntax**

```
<h3>This is Level 4 Heading, without style</h3>
<h1>This is again Level 1 heading with default styles</h1>
```

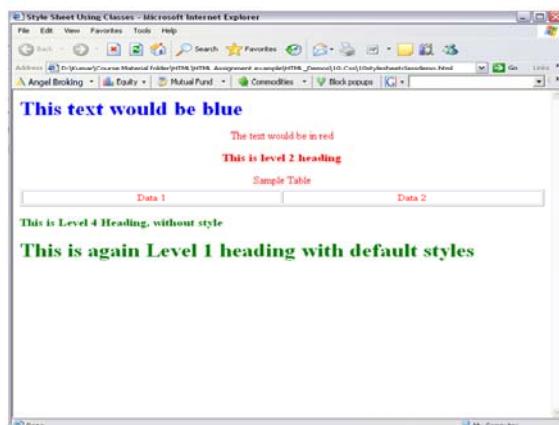
- 19 -



Demo



➤ stylesheetclassdemo.html



- 20 -

```
<html>
<head><title>Style Sheet Using Classes</title></head>
<style>
H1.myClass {color: blue}
.myOtherClass { color: red; text-align:center}
</style>
<body style="background: white; color:green">
<H1 class="myClass">This text would be blue</H1>
<p class="myOtherClass">The text would be in red</P>
<H3 class="myOtherClass">This is level 2 heading</H3>
<table class=myotherClass border width=100%>
<caption>Sample Table</caption>
<td>Data 1</td><td>Data 2</td>
</table>
<h3>This is Level 4 Heading, without style</h3>
<h1>This is again Level 1 heading with default styles</h1>
</body>
</html>
```

Controlling Visibility

Visibility Property: Use this property to turn the display of element off. These invisible elements still affect the layout, but their content is hidden.

Accepted values are:

Visible

Hidden

Example:

```
h3 { background: gold; position: absolute; left: 40px; top: 40px; width:100px  
height: 200px; z-index: 2; visibility: hidden}
```

Here 'Visibility: hidden' style attribute is specified in the H3 header tag. It causes to hide content specified in <H3></H3>tag. Hence, content enclosed in <H3></H3> does not display in the browser window .

```
h2 { background: yellow; position: absolute; left: 80px; top: 80px; width:  
200px; height: 200px; z-index: 1; visibility: visible}
```

Here 'Visibility: visible' style attribute is specified in to H2 header tag. Hence, content enclosed in <H2></H2>tag displays in the browser window.

```
<html>  
<head>  
<title>Style Sheet visibility Example</title>  
<style>  
h3 { background: gold; position: absolute; left: 40px;  
top: 40px; width: 100px; height: 200px; z-index: 2; visibility: hidden}  
h2 { background: yellow; position: absolute; left: 80px;  
top: 80px; width: 200px; height: 200px; z-index: 1; visibility: visible}  
</style>  
</head><body>  
    <h3>This is an example of z-index</h3>  
    <h2>This is one of the paragraph from the document</h2>  
</body></html>
```

LAYER:

This tag allows you to position blocks of contents. These blocks of positioned content are also called layers in **Navigator 4.0**. Positioned blocks of content can overlap each other, be transparent or opaque, and be visible or invisible. They can also be nested. Use the LAYER tag to specify an *absolute* position for a block of content, and use the ILAYER tag to specify a *relative* position.

ILAYER:

This tag allows you to offset content from its natural position in the page, which is the position the content would normally have if it was not positioned. That is, you can position the content relative to its natural position. This relatively positioned content is both inflow, in that it occupies space in the document flow, and inline, in the sense that it shares line space with other HTML elements. If you want relatively positioned content to appear on a separate line, you can insert a break before the ILAYER tag or embed the ILAYER tag in the <DIV> tag in **Navigator 4.0**.

NOLAYER:

Content that is surrounded by <NOLAYER> and </NOLAYER> tags is displayed by browsers that *do not* support LAYER and ILAYER tags. This content is ignored by browsers that *do* support these tags.

Absolute coordinates are measured to the right and down from the upper-left corner of the browser window.

Relative coordinates are measured to the right and down from the position where the layer would normally appear.

<LAYER> and <ILAYER> tags are exclusive to Netscape Navigator 4.0 and are not recognized in any other browser. You can position layers using absolute or relative coordinates. <LAYER> tag uses *absolute* positioning and the <ILAYER> tag uses *relative* positioning.

Attributes for the <LAYER> tag include the following:

above: This attribute lets you stack a layer directly on top of a layer whose name you identify. For example, above="bird" places the current layer above the "bird" layer.

background: This attribute accepts an URL of an image to be tiled behind the layer.

bgcolor: This attribute sets the background color of the layer.

below: This attribute lets you stack a layer directly underneath a layer whose name you identify. For example, below="bird" places the current layer below the "bird" layer.

clip: This attribute lets you show only a portion of the layer. It accepts four pixel values representing the coordinates of the upper-left and lower-right corners of the visible area. These pixel values are relative to the upper-left corner of the layer. For example, if a layer has a height and width of 200 pixels, then `clip="20,20,140,140"` displays an area 20 pixels from the left and top edge and 60 pixels from the right and bottom edges.

Height

Width

Id: This attribute identifies that layer with a name. JavaScript can use this name to dynamically alter the properties of the layer.

Visibility: This attribute lets you make layers visible or invisible. It accepts the values SHOW, HIDE, or INHERIT which means that the layer has the same visibility as its parent.

Z-index: This attribute sets the stacking order for the layer. The higher the number, the higher it appears on the stack.

Layer Tag Attributes

ID="layerName"

specifies the name of the layer, enabling other layers and JavaScript scripts to refer to it.

LEFT="pixelPosition" and TOP="pixelPosition"

specify the horizontal and vertical positions of positioned layers or the relative horizontal and vertical positions for inflow layers.

PAGEX and PAGEY

specify the horizontal and vertical positions of the layer relative to the document's window.

SRC="file"

specifies the pathname of a file that contains HTML-formatted content for the layer.

Z-INDEX="n"

specifies the stacking order of a layer. Allows a layer's z-order to be specified in terms of an integer. Layers with higher numbered values are stacked above those with lower ones. Only positive Z-INDEX values are allowed. The use of this attribute cannot be combined with the use of the ABOVE or BELOW attributes.

ABOVE="layername"

specifies the layer immediately on top of a newly created layer in the stacking order; that is, the new layer is created just below the layer specified by *layername*. The use of this attribute cannot be combined with the use of the BELOW or Z-INDEX attributes.

BELLOW="layername"

identifies the layer immediately beneath the newly created layer in the stacking order; that is, the new layer is created just above the layer specified by *layername*. The named layer must already exist. The use of this attribute cannot be combined with the use of the ABOVE or Z-INDEX attributes.

WIDTH="width"

specifies the width of the layer's content. It controls the right margin for wrapping purposes. The value can be expressed as an integer pixel value or as a percentage of the width of the containing layer.

HEIGHT="height"

specifies the height of the layer's clipping region and serves as the reference for setting the relative height of children layers. The value can be expressed as an integer pixel value or as a percentage of the height of the containing layer (or of the window for a top-level layer).

CLIP="n,n,n,n"

specifies the clipping rectangle (viewable area) of the layer, which can be less than the width and height of the content of the layer. The value is a set of four numbers. Each of the four values represents numbers of pixels.

VISIBILITY

specifies whether the layer is visible or not. A value of SHOW shows the layer. A value of HIDDEN hides the layer. A value of INHERIT causes the layer to have the same visibility as its parent layer.

BGCOLOR="color"

specifies the background color of the layer. See [Color Units](#) for information about color values.

BACKGROUND="imageURL"

specifies an image to use as the background.

OnMouseOver="JScode", OnMouseOut="JScode"

are JavaScript event handlers that are invoked when the mouse cursor enters or leaves the layer. For information about JavaScript, see the [JavaScript Guide](#) or the [JavaScript Reference](#).

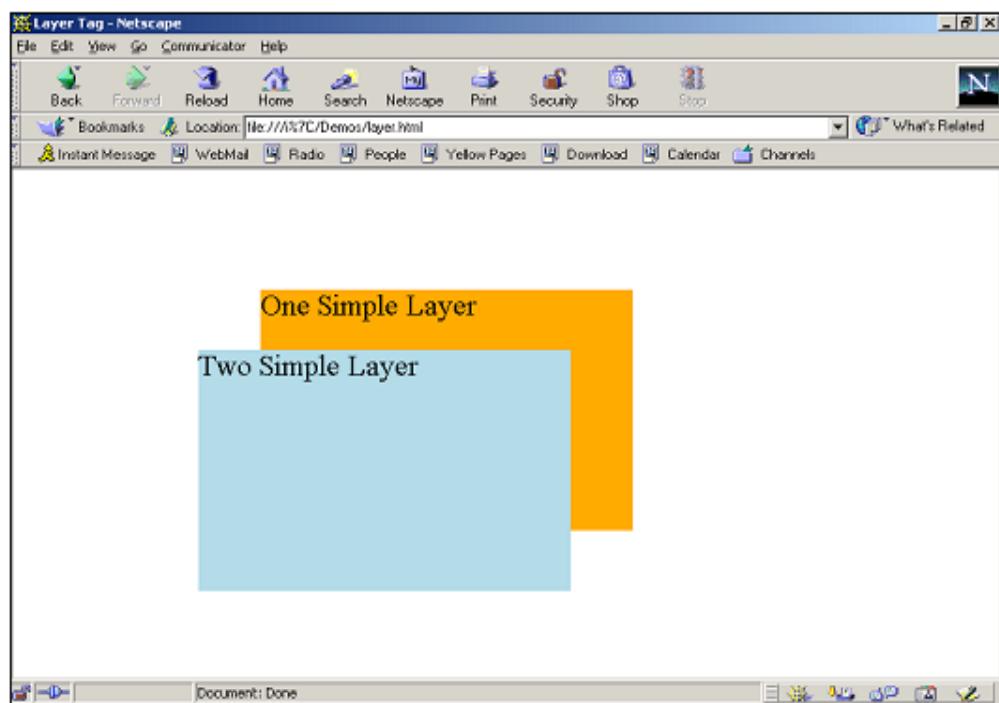
OnFocus="JScode", OnBlur="JScode"

are JavaScript event handlers that are invoked when the layer gets or loses keyboard focus.

OnLoad="JScode"

is an event handler invoked when the layer is loaded.

```
<html><head><title>Layer Tag</title></head>
<body>
<layer id="simple1" bgcolor="orange" left=200
top=100 width=300 height=200>
<font size=5>One Simple Layer</font>
</layer><layer id="simple2" bgcolor="lightblue"
below="simple1" left=150 top=150 width=300 height=200>
<font size=5>Two Simple Layer</font>
</layer></body></html>
```



Scripting languages are the drivers of Dynamic HTML. The combination of HTML and cascading style sheets (CSS) provides you with control over the look and feel of your Web page; however by themselves, the technologies are static and unmoving, much like an automobile without a driver. Scripting languages allow you to bring interactivity and motion to your HTML and CSS.

Internet Explorer 4.0, includes two scripting languages that you can use with Dynamic HTML: JavaScript and VBScript.

Netscape only supports JavaScript.

Lab



➤ Lab 8



- 26 -

Summary



- Use of CSS in web pages
- Learnt to use CSS in different modes



- 27 -

Review Question



- **Question 1: Absolute coordinates are measured as:**
 - right and down from the upper-left corner of the browser window
 - right and down from the position where the layer would normally appear
 - Both of the above
- **Question 2: The z-index property sets the stack length of an element**
 - True/False
- **Question 3: CSS stands for _____**



- 28 -

Review Question: Match the Following

- | | |
|-------------------------|--|
| • Inline Style Sheets | • Set the stacking order for the layer. |
| • Embedded Style Sheets | • Placed either in the <HEAD> section or between the </HEAD> and <BODY> tags. |
| • Linked Style Sheets | • Attribute lets you make layers visible or invisible. |
| • Visibility | • Separate files with a style sheet code you can access using this style sheet type. |
| • Z-index | • Uses style attribute to specify style in tags. |



- 29 -



Web Basics (HTML)

Lab Book

Copyright © 2011 iGATE Corporation. All rights reserved. No part of this publication shall be reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior written permission of iGATE Corporation.

iGATE Corporation considers information included in this document to be Confidential and Proprietary.

Document Revision History

Date	Revision No.	Author	Summary of Changes
2-Jun-05	1.0	Kumar B	Initial Draft
1-Oct-09	1.0	Ummeaiman Diwanji	Quality review. Transferred to the new template.
1-Oct-09	1.0	Anu Mitra	Final review
16/06/10	2	Anu Mitra	Refinements
27/05/11	3	Anu Mitra	Integration Refinements

Table of Contents

<i>Getting Started</i>	4
Overview.....	4
Setup Checklist for HTML.....	4
Instructions	4
Learning More (Bibliography)	4
<i>Lab 1: HTML Basics</i>	5
1.1: Create HTML Page.....	5
1.2 Example: MyFirstPage.html	7
1.3 Example: Headers.html	8
1.4 Example: Address.html.....	9
1.5 Example: HorizontalRuler.html.....	10
1.6 Example: PreFormattedText.html.....	11
<i>Lab 2: Creating Tables</i>	14
Problem 1: Table Layout	14
Problem 2: Fun With Food.....	15
Problem 3: Table Heading << To Do>>	17
Problem 4: Calendar <<To Do>>	17
<i>Lab 3: Working with Lists</i>	19
Problem 1: Types of Lists	19
<i>Lab 4: Working with Links</i>	23
Problem 1: Welcome to Big Company <<To Do>>.....	23
<i>Lab 5: Image Handling</i>	27
<i>Lab 6: HTML Forms for User Input</i>	30
<i>Lab 7: Working with Frames</i>	33
<i>Lab 8: Cascading Style Sheets</i>	37
Problem 1: Inline Style Sheet.....	37
Problem 2: Embedded Style Sheets	38
Problem 4: XHTML and Stylesheets	40
Appendix A: HTML Standards.....	42
Appendix B: Table of Figures	46
Appendix C: Table of Examples	47

Getting Started

Overview

This lab book is a guided tour for learning HTML. It comprises solved examples and 'To Do' assignments. Follow the steps provided in the solved examples and work out the 'To Do' assignments given.

Setup Checklist for HTML

Here is what is expected on your machine in order for the lab to work.

Minimum System Requirements

- Intel Pentium 90 or higher (P166 recommended)
- Microsoft Windows 95, 98, or NT 4.0, 2k, XP.
- Memory: 32MB of RAM (64MB or more recommended)
- Internet Explorer 6.0 or higher

Please ensure that the following is done:

- A editor like Notepad, Eclipse, Visual Studio 2008 is installed.

Instructions

- For all coding standards refer Appendix A. All lab assignments should refer coding standards.
- Create a directory by your name in drive <drive>. In this directory, create a subdirectory html_assgn. For each lab exercise create a directory as lab <lab number>.
- You may also look up the on-line help provided in the MSDN library.
- The faculty will introduce you to the editor to be used.

Learning More (Bibliography)

- HTML Source Book by Ian S. Graham
- HTML: Complete Concepts and Techniques by Gary B. Shelly
- HTML: The Definitive Guide by Chuck Musciano
- Dynamic HTML: The Definitive Reference by Danny Goodman
- HTML: The Complete Reference by Thomas A. Powell

Lab 1: HTML Basics

Goals	<ul style="list-style-type: none"> Understand the process of creating an HTML page and viewing it in a browser window. Learn to apply physical or logical character effects. Learn to manage document spacing
Time	45 minutes

1.1: Create HTML Page

Create a web page to display the text 'This is the first html page created'.

Solution:

Step 1: Click the **Start** button. On the **Programs** menu, navigate to the **Accessories** submenu. Click **Notepad**.

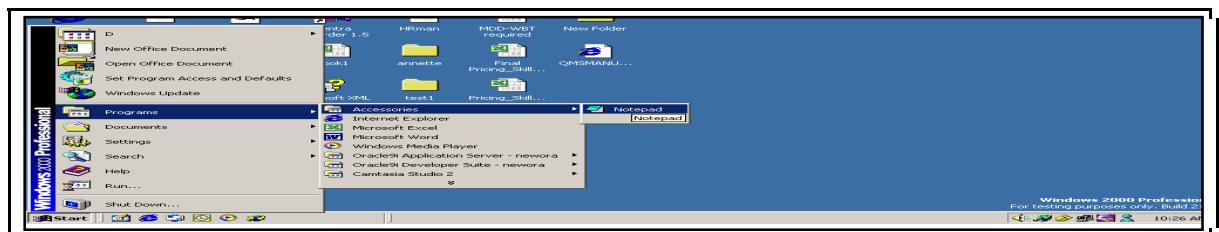


Figure 1: Open Notepad

Step 2: Write the HTML program in Notepad. Refer to the figure that follows, for the HTML code.

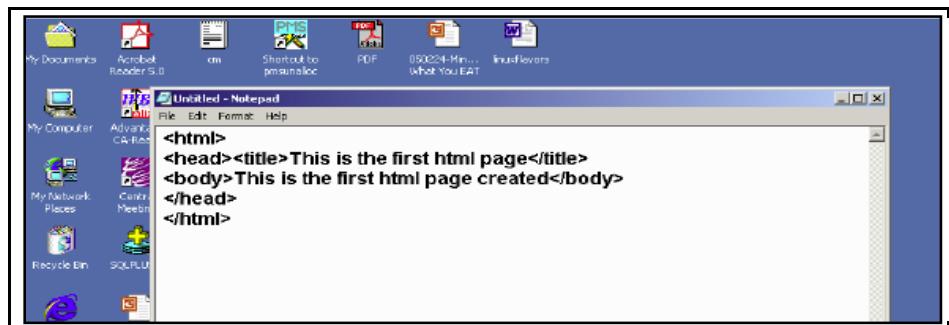


Figure 2: HTML Code

Step 3: Save the file with extension *.html*. Save it in the *lab1* directory as *firstpage.html*.

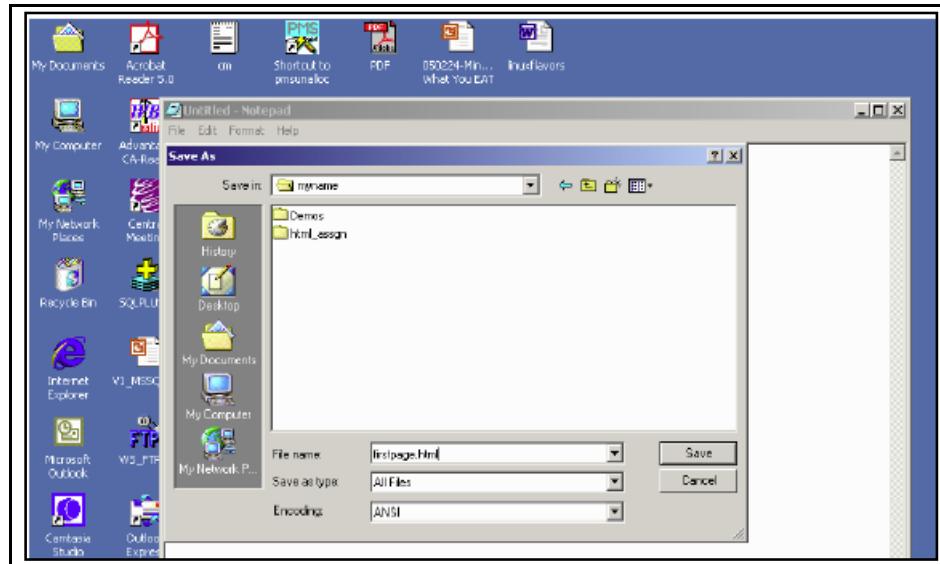


Figure 3: Save contents as an .html file

Step 4: From Internet Explorer, on the **File** menu, click **Open**. **Open** dialog box appears. Click **Browse** to select the file you have just saved. Refer to the figure that follows.



Figure 4: Open the .html file from the browser

Step 5: Once you have selected the file, click **OK** in the **Open** dialog box. Output appears as shown in the figure that follows.

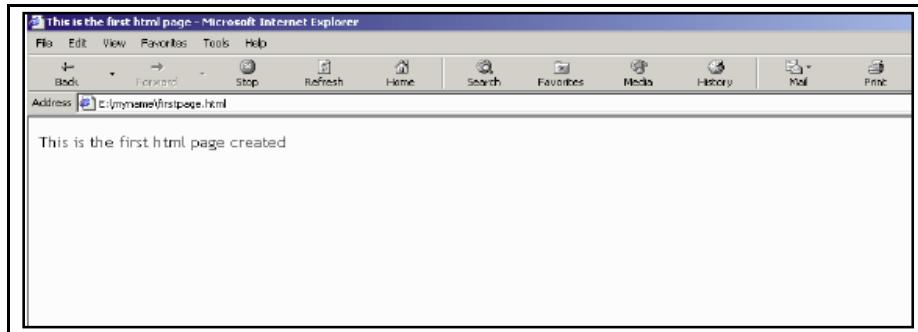


Figure 5: First.html in a browser

1.2 Example: MyFirstPage.html

```
<html>
<head>
<title>My First Page</title>
<META [http-equiv] [contents=n]>
<meta http-equiv=refresh content=60>
-will refresh the current document after every 60 seconds.

<meta http-equiv=refresh content="20;url=c:/html/html34.htm">
-will load specified file after 20 seconds.
<base href="c:/mydir/html/">
<!-- you can use shortcuts in your URLs if you must reference several files from the same
location.-->
</head>
<body> Hello World!! </body>
</html>
```

Example 1: MyFirstPage.html

Output of the above HTML code is:

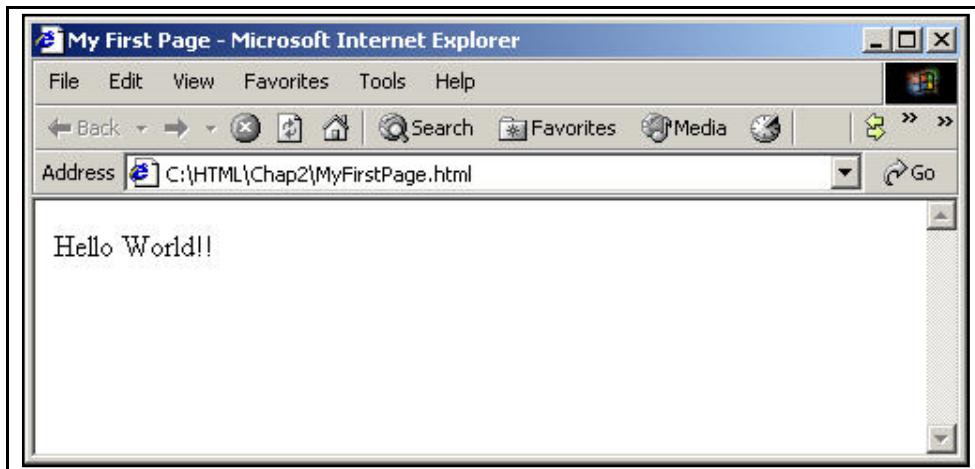


Figure 6: *MyFirstPage.html* Output

1.3 Example: Headers.html

```
<html>
<head><title>This is the first html page</title>
<body>This is the first html page created
<h1 align="left">This is level 1 heading</h1>
<h2 align="right">This is level 2 heading</h2>
<h3 align="center">This is level 3 heading</h3>
</body>
</head></html>
```

Example 2: *Headers.html*

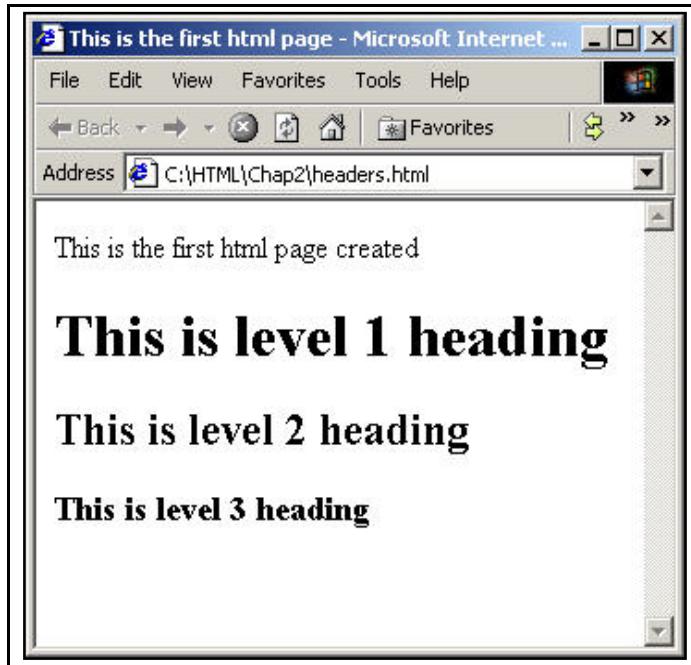


Figure 7: *Headers.html* Output

1.4 Example: Address.html

```
<html>
<head>
<title>Address Example</title>
</head>
<body>
<font size="2">Your address:</font><br>
Abc Xyz<br>
<address>b/102 royal palms,</address>
<address>off. s. v. road,</address>
<address>Andheri-West,</address><address>Mumbai.</address>
</body>
</html>
```

Example 3: Address.html

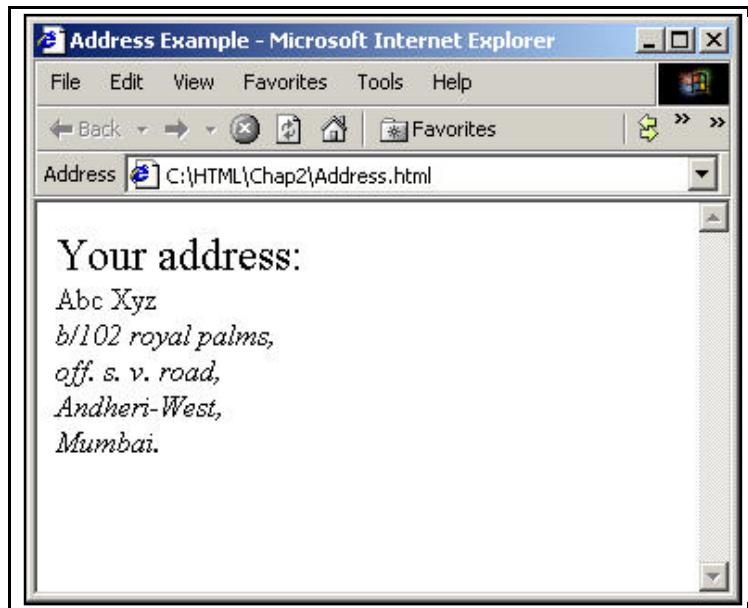


Figure 8: *Address.html* Output

1.5 Example: HorizontalRuler.html

```
<html>
<head><title>
The HR Tag
</title></head>
<body>
<hr width="80%" color="GREEN" align=right>
</body>
</html>
```

Example 4: *HorizontalRuler.html*

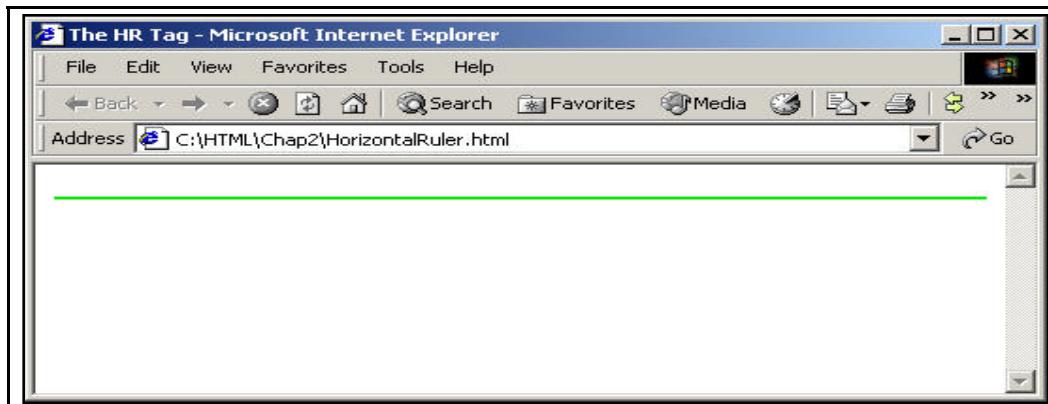
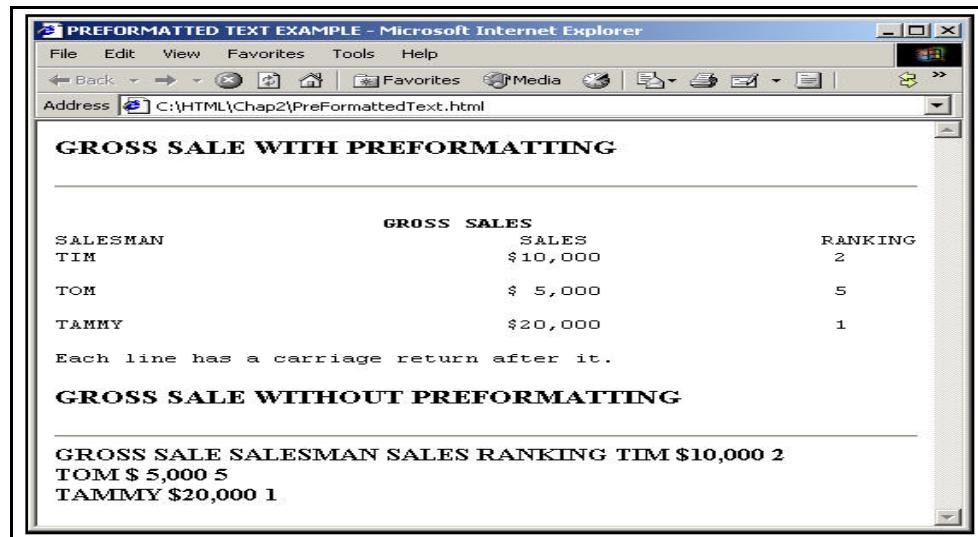


Figure 9: HorizontalRuler.html Output**1.6 Example: PreFormattedText.html**

```

<html>
<head><title>
PREFORMATTED TEXT EXAMPLE
</title></head>
<body>
<h3>GROSS SALE WITH PREFORMATTING</h3>
<hr>
<pre>
    <b>GROSS SALES</b>
SALESMAN          SALES          RANKING
TIM               $10,000        2<BR>
TOM               $ 5,000        5<BR>
TAMMY              $20,000        1<BR>
Each line has a carriage return after it.
</pre>
<h3>GROSS SALE WITHOUT PREFORMATTING</h3>
<p><hr>
    <b>GROSS SALE</b>
SALESMAN          SALES          RANKING
TIM               $10,000        2<BR>
TOM               $ 5,000        5<BR>
TAMMY              $20,000        1<BR>
</body>

```

Example 5: PreFormattedText.html**Figure 10: PreFormattedText.html Output**

Problem 1: Resume Creation <<To Do>>

Problem Statement:

Create your resume page as per the format shown in the figure that follows.



Figure 11: Resume Page

Solution:

1. Open **Editor**. Type the code and save the file.
2. Use Heading 2 for the headings “Educational Qualifications” and “Skill Set”.
3. Use font size 3 for data pertaining to educational qualifications and skill set.
4. Display details against categories under *Skill Set* in *italics*.
5. Start the Internet Explorer. On the **File** menu, click **Open**. **File** open dialog box appears. Click the **Browse** button and select *prob2.html* file.
6. Check if the output is as per the requirement.

Lab 2: Creating Tables

Goals	At the end of this lab session you will understand: <ul style="list-style-type: none"> • Attributes of a Table • Table Headers • Table Data • Table Formatting • Control Table Borders • Grouping of Columns
Time	90 minutes

Problem 1: Table Layout

Problem Statement:

Create a web page that contains a heading 'Layout is here!' followed by a horizontal rule and a table with a single row as shown in the figure that follows.



Figure 12: Table layout

Solution

Step 1: Write the following code in **Notepad** and save the file:

```
<HTML>
<HEAD>
  <TITLE>Tables</TITLE>
</HEAD>
<BODY>
<TABLE BORDER="0">
<TR>
  <TD WIDTH="50"><BR></TD>
  <TD WIDTH="400">
    <H1 ALIGN="CENTER">Layout is here!</H1>
<HR>
```

```

<P>This is a very simple layout that would
have nearly been impossible to do without tables</P>
</TD><TD width="100"><BR></TD>
</TR>
</TABLE><TABLE border="1">
<TR>
<TD width="50"><BR></TD>
<TD width="400"><BR></TD>
<TD width="100"><BR></TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Example 6: Table Layout

Step 2: Open the page in the browser and verify if the output is as per the figure.

Step 2:

Open it in the browser and check the output with the required output.

Problem 2: Fun With Food

Problem Statement:

Create a web page, which uses a table with columns *Fruit*, *Color* and *Cost per pound* as shown in the figure that follows.

Fruit	Color	Cost per pound
Grapes	Purple	1.25
Cherries	Red	154.79
Kiwi	Brown	10.00

This is the footer area

Figure 13: Fruits Table

Solution

Step 1:

Write the following code in **Notepad** and save the file.

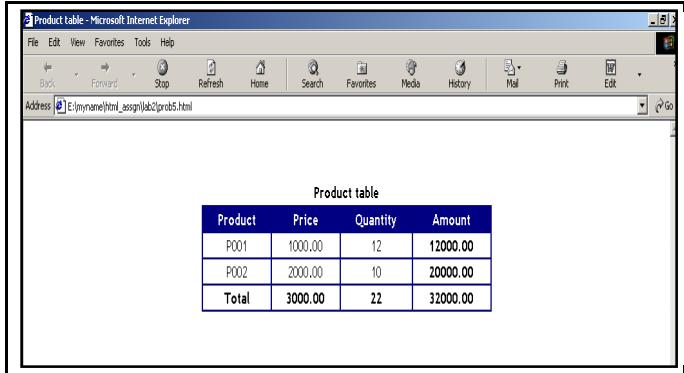
```
<HTML>
<HEAD>
    <TITLE>Fruits Table</TITLE>
</HEAD>
<BODY>
<TABLE border="1" frame="BOX" rules="groups">
<CAPTION>Fun with food</CAPTION>
<COLGROUP>
    <COL>
</COLGROUP>
<COLGROUP>
    <COL align="center">
        <COL>
</COLGROUP>
<THEAD>
<TR>
    <TH bgcolor="yellow">Fruit</TH>
    <TH bgcolor="yellow">Color</TH>
    <TH bgcolor="yellow">Cost per pound</TH>
</TR>
</THEAD>
<TBODY>
<TR>
    <TD>Grapes</TD>
    <TD>Purple</TD>
    <TD>1.25</TD>
</TR>
<TR>
    <TD>Cherries</TD>
    <TD>Red</TD>
    <TD>154.79</TD>
</TR>
<TR>
    <TD>Kiwi</TD>
    <TD>Brown</TD>
    <TD>10.00</TD>
</TR>
</TBODY>
<TFOOT>
<TR>
    <TH colspan="3">This is the footer area</TH>
</TR>
</TFOOT>
</TABLE>
</BODY>
</HTML>
```

Example 7: Fruit Table

Step 2: Open the file page in the browser to check the required output.

Problem 3: Table Heading << To Do>>

Problem Statement: Create a html page . When this page is opened in a browser, it should appear as shown in the following figure



Product table			
Product	Price	Quantity	Amount
P001	1000.00	12	12000.00
P002	2000.00	10	20000.00
Total	3000.00	22	32000.00

Figure 14: Product table

Note: Table heading - Background color is : navy and font color is : white.

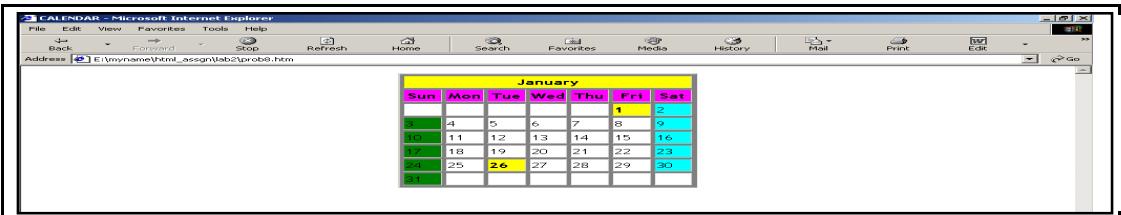
Solution

1. Open **Editor**. Type the code and save the file.
2. Open the page in browser
3. Check the page shown in the browser and verify that it is as per the requirement.

Problem 4: Calendar <<To Do>>

Problem Statement:

Design a web page to display a calendar for a month using html table.



January						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Figure 15: Calendar

Note: Background colors to be used: For all the Sundays: **green**, for all the Saturdays: **aqua**, for 1, 26 Jan: **yellow**

Solution

1. Open **NotePad**. Type the code and save the file.
2. Open the page in browser.
3. Verify that the output is as per requirements.

Lab 3: Working with Lists

Goals	At the end of this lab session you will be able to use following types of lists: <ul style="list-style-type: none"> • Numbered List • Bulleted List • Directory List • Glossary List
Time	30 minutes

Problem 1: Types of Lists

Problem Statement:

Design a web page as shown in figure 3.1(a) and 3.1(b):

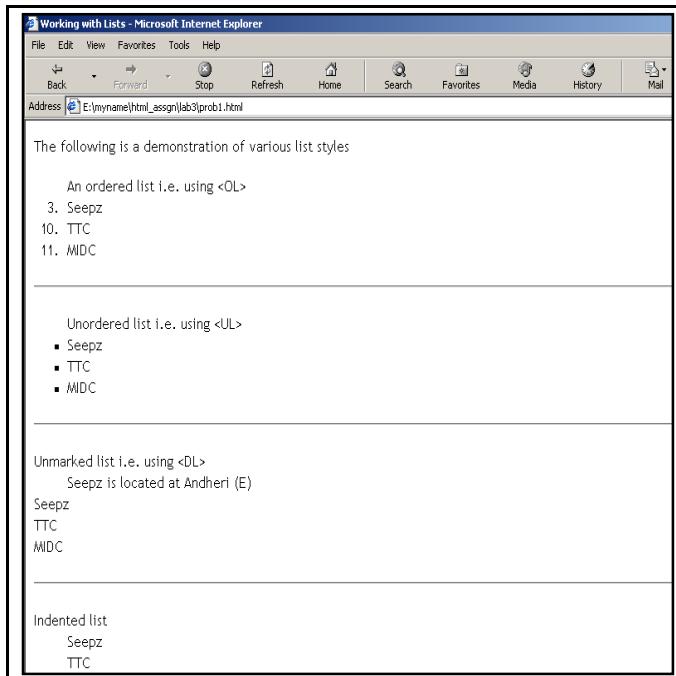


Figure 16: Part (a)

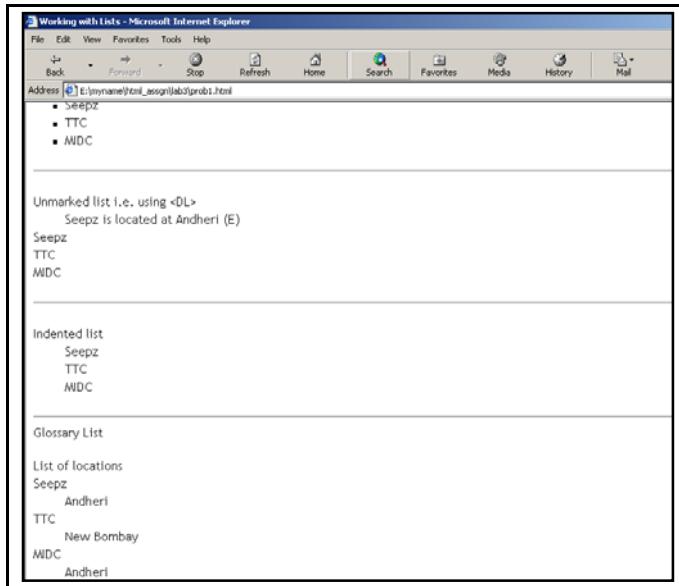


Figure 17: Part (b)

Solution

Step 1: Write the following code in **Notepad** and save.

```

<html>
<head><title>Working with Lists</title></head>
<body>
<p> The following is a demonstration of various list styles
<ol start="3" >An ordered list i.e. using &ltOL&gt
<li>Seepz
<li value="10">TTC
<li>MIDC
</ol>
<hr>
<ul type="square">Unordered list i.e. using &ltUL&gt
<li>Seepz
<li>TTC
<li>MIDC
</ul>
<hr>
<dl>Unmarked list i.e. using &ltDL&gt
<dd>Seepz is located at Andheri (E)
<dt>Seepz
<dt>TTC
<dt>MIDC
</dl>
<hr>
<dl>Indented list
<dd>Seepz
<dd>TTC

```

<dd>MIDC

```
</dl>
<hr>
Glossary List
<dl>List of locations
<dt>Seepz
<dd>Andheri
<dt>TTC
<dd>New Bombay
<dt>MIDC
<dd>Andheri
</dt>
</body>
</html>
```

Example 8: Types of Lists

Step 2: Check the page shown in the browser and verify that it is as per the requirement.

Problem 2: Subjects <>To Do>>

Create a web page to display a list as shown in the figure that follows.

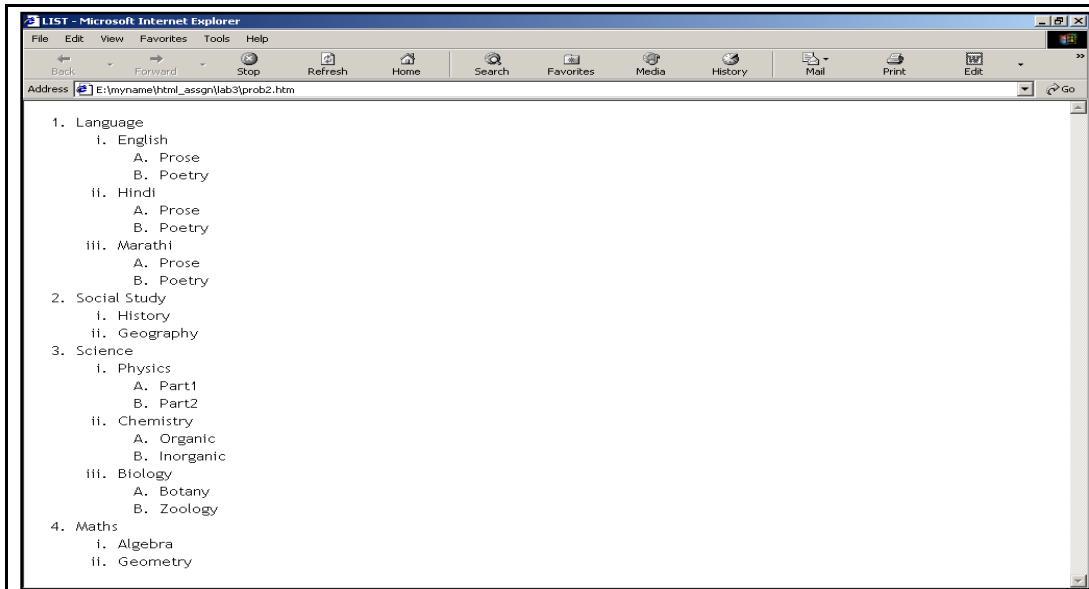


Figure 18: Subject list

Solution

1. Open **Notepad**. Type the code and save the file.

2. Open the page in browser.
3. Check the page shown in the browser and verify that it is as per the requirement.

Lab 4: Working with Links

Goals	At the end of this lab session you will be able to: <ul style="list-style-type: none">• Create links to web documents.• Create links to email.• Create hyperlinks for lists and table data.• Provide target for hyperlink.
Time	30 minutes

Problem 1: Welcome to Big Company <<To Do>>

Problem Statement:

Design a simple home page for a company with a heading and 3 links – About, Products, Contact as given in the figure below.



Figure 19: Big Company home page

When you click the “About” hyperlink, following page should be displayed.

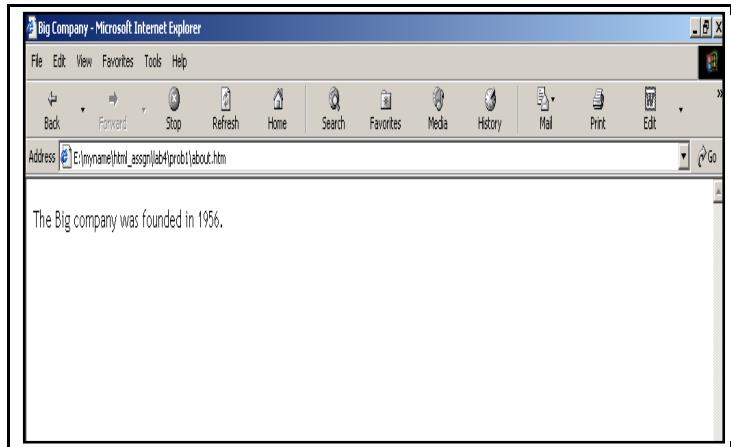


Figure 20: About

When you click the **Back** button on the browser toolbar, they should be redirected to the page *prob1.html*. Click the “Products” hyperlink to reach the following page:

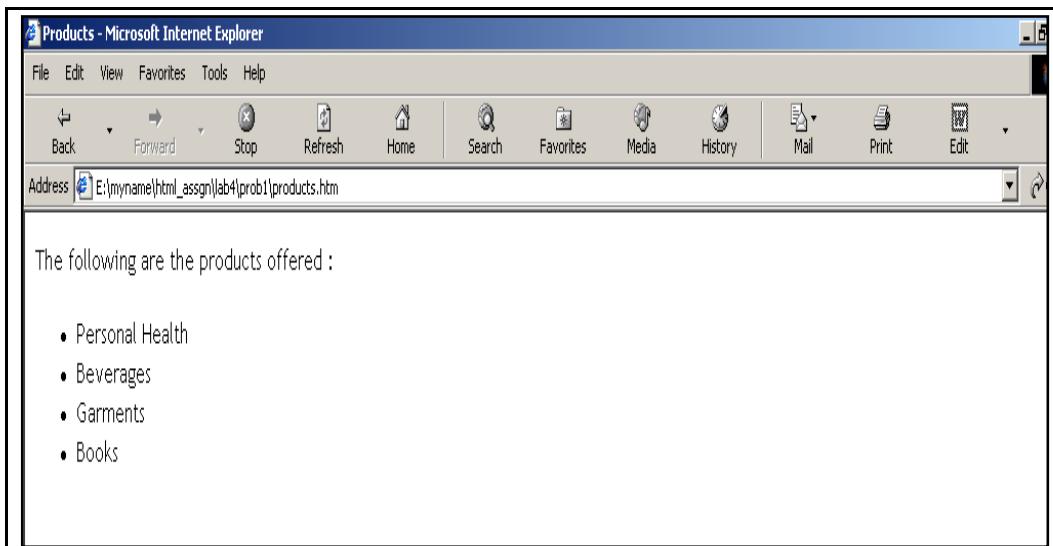


Figure 21: Products

When you click the **Back** button on the browser toolbar, they are redirected to page *prob1.html*. Click the “Contact” hyperlink. It opens Outlook Express and the e-mail address given in the *To* field, which is edude@patni.com in the following illustration, is displayed in the New message window. This email address is specified in the *mailto* attribute.

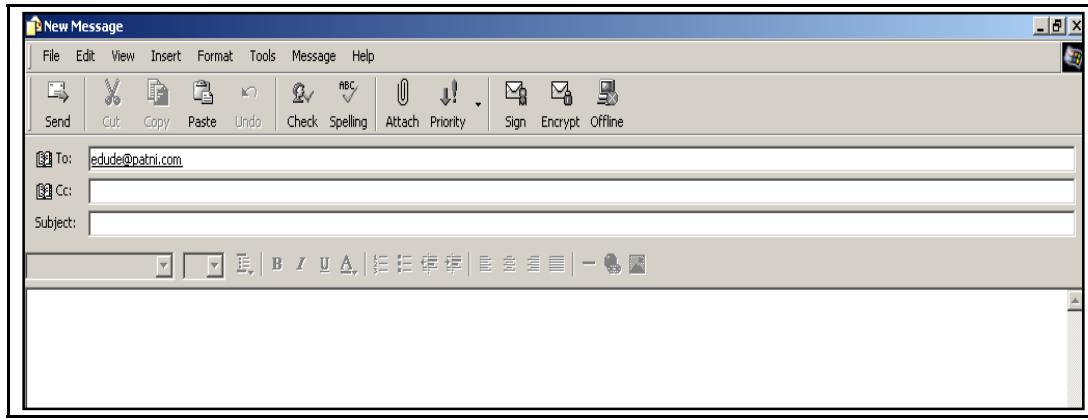


Figure 22: Contact

Solution

Step 1: Write the following code in **Notepad** and save the file.

```
<HTML>
<HEAD>
    <TITLE>Big Company</TITLE>
</HEAD>
<BODY>
    The Big company was founded in 1956.
</BODY>
</HTML>
```

Example 9: Big Company

Step 2: Write the following code in **Notepad** and save it as the file.

```
<HTML>
<HEAD>
    <TITLE>Products</TITLE>
</HEAD>
<BODY>
    The following are the products offered :

    <UL>
        <LI>Personal Health
        <LI>Beverages
        <LI>Garments
        <LI>Books
    </UL>
</BODY>
</HTML>
```

Example 10: Products

Step 3: Write the following code in **Notepad** and save the file.

```
<HTML>
<HEAD>
</HEAD>
<BODY>
<TABLE width="650">
<TR>
    <TD WIDTH="100">
        <A href="about.htm">About</A><BR><BR>
        <A href="products.htm">Products</A><BR><BR>
        <A href="mailto:edude@patni.com">Contact</A><BR><BR>
    </TD>

    <TD WIDTH="450">
        <H1 ALIGN="CENTER">Welcome to Big Company</H1>
        <HR>
        <P>This is a very simple layout that would
        have nearly been impossible to do without tables</P>
    </TD>
<TD width="100"><BR></TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

Example 11: Welcome to Big Company

Step 4: Start the Internet Explorer. On the **File** menu, click **Open**. **Open** dialog box appears. Click the **Browse** button and open the page with links file. Verify if the links on the page are working as per the requirement.

Lab 5: Image Handling

Goals	At the end of this lab session you will be able to: <ul style="list-style-type: none">• Understand the use of inline images.• Attributes of an inline image.• Text and image aligning.• Use of an image as a hyperlink.
Time	30 minutes

Problem 1: Images with Clickable Areas <<To do>>

Problem Statement:

Create a web page with an image map of a picture (*bike.gif*) with four clickable rectangles. Four clickable areas for the bike image are *engine*, *seat*, *fork* and *headlight*. When you click within a rectangular area it needs to display information about selected part.

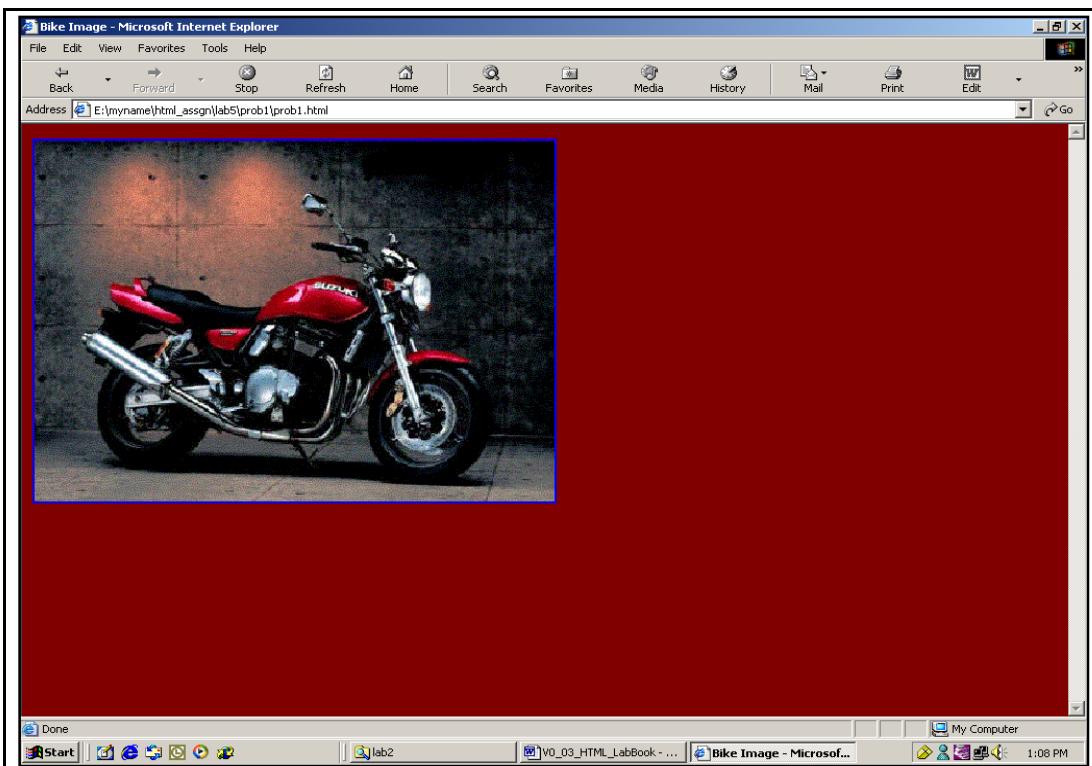


Figure 23 : Imagemap

When you click the engine portion of the image, it should display the following:

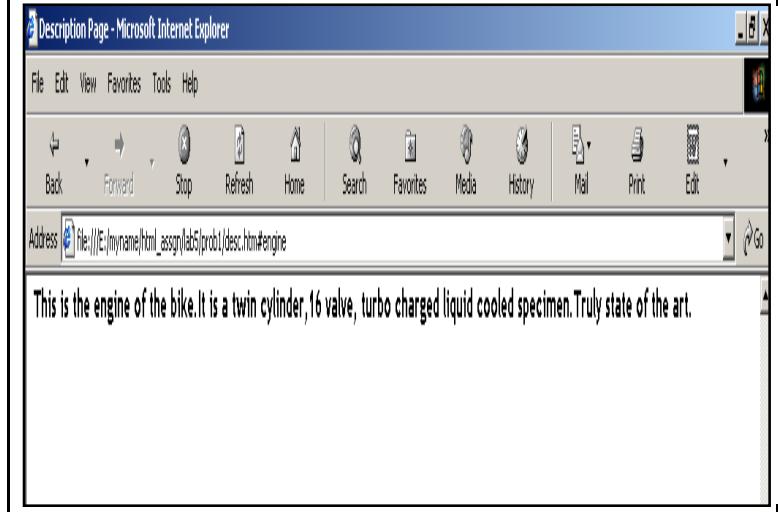


Figure 24: Engine

Solution

Step 1: Write the following code in **Notepad** and save the file.

```
<HTML>
<HEAD>
  <TITLE>Description Page</TITLE>
</HEAD>
<BODY>
  <H2>Description of the various sections of the bike</H2>
  <TABLE WIDTH="100%" CELLSPACING="250">
  <TR>
    <TD><A name="engine"></A>
      <B> This is the engine of the bike. It is a twin cylinder, 16 valve, turbo charged liquid cooled specimen. Truly state of the art.</B>
    </TD></TR>
  <TR>
    <TD><A name="seat"></A>
      <B> This is the seat of the bike. Designed to seat two comfortably and upholstered in calf leather.</B>
    </TD></TR>
  <TR>
    <TD><A name="fork"></A>
      <B> This is the front wheel section of the bike. The air-suspension forks are designed to absorb even the harshest of shocks while riding. The disc-brakes on the front-wheel facilitate immediate braking even at break-neck speeds.</B>
    </TD></TR>
  <TR>
    <TD><A name="head"></A>
      <B> This is the head section of the bike. The front panel is carefully designed to provide easy access to the various functions of the bike. The headlight is powered by 16 volt batteries to provide clear road vision even
  </TABLE>
</BODY>
</HTML>
```

```
in pitch darkness.</B></TD></TR>
</TABLE>
</BODY></HTML>
```

Example 12: Image Maps

Step 2: Write the following code in **Notepad** and save the file

```
<HTML>
<HEAD>
  <TITLE>Bike Image</TITLE>
</HEAD>
<BODY BGCOLOR = "maroon">
<MAP NAME = "bikemap">
<AREA SHAPE = "RECT" COORDS = "198,228,246,273"
      HREF = "desc.htm#engine">
<AREA SHAPE = "RECT" COORDS = "105,140,219,193"
      HREF = "desc.htm#seat">
<AREA SHAPE = "RECT" COORDS = "341,221,396,284"
      HREF = "desc.htm#fork">
<AREA SHAPE = "RECT" COORDS = "334,130,377,171"
      HREF = "desc.htm#head">
</MAP>
<IMG src = "bike.gif" ALT = "Imgmap" USEMAP = "#bikemap">
</BODY>
</HTML>
```

Example 13: Bike Image

Lab 6: HTML Forms for User Input

Goals	At the end of this lab session you will be able to: <ul style="list-style-type: none"> • Understand the role of forms in web pages. • Understand various HTML elements used in forms. • Develop HTML forms in web pages.
Time	120 minutes

Problem 1: Form

Problem Statement:

Design a web page *prob1.html* in the directory *lab6*. When *prob1.html* is opened in the browser, the page is displayed as shown in the figure that follows.

The screenshot shows a Microsoft Internet Explorer window with the title "Form Methods - Microsoft Internet Explorer". The address bar shows "C:\myname\html_assign\lab6\prob1.htm". The form itself has the following fields:

- Text input field: "Enter the password" (empty)
- Text input field: "Enter your surname" (value: "Sukuru")
- Text area: "Enter ur address" (empty)
- Text input field: "Address" (empty)
- Text input field: "Select the training programs attended:" (empty)
- Checkboxes: COBOL (unchecked), IDMS (checked), Java (unchecked)
- Text input field: "Select the group you belong to:" (empty)
- Radio buttons: Group 1 (checked), Group 2 (unchecked), Group 3 (unchecked), Others (unchecked)
- Text input field: "Which training program would you like to attend?" (value: "JavaScript")
- File upload button: "Browse..."
- Buttons: "Exit", "Save", "Reset"

Figure 25: Forms

Solution

Step 1: Write the following code in **Notepad** and save it as *lab6\prob1.html*.

```

<HTML>
<HEAD>
<TITLE> Form Methods</TITLE>
</HEAD>
<BODY bgcolor="cyan">
<FORM action="mailto:edude@patni.com" name="ab" method="post"
enctype="multipart/form-data">
<P>
<LABEL>Enter the password
<INPUT type="password" name="USERNAME" size = "20" value="abc"
tabindex="3"></LABEL>
<INPUT type="hidden" name="coname" value="PCS">
Enter your surname :
<INPUT type="text" name="surname" SIZE = "20" readonly value="Sukuru
" tabindex="2" maxlength="30">
<BR><BR>
Address : <TEXTAREA name="addr" Rows="5" cols="40" tabindex="0"
accesskey="A">Enter ur address </TEXTAREA>
<BR><BR>
Select the training programs attended : <BR>
<INPUT type="checkbox" name="s-cobol"> COBOL
<INPUT type="checkbox" name="s-idms" checked> IDMS
<INPUT type="checkbox" name="s-java"> Java
<BR><BR>
Select the group you belong to :
<SPACE type="VERTICAL" size="15">
<INPUT type="radio" name="s-grp" value="grp1"> Group 1
<INPUT type="radio" name="s-grp" value="grp2" checked> Group 2
<INPUT type="radio" name="s-grp" value="grp3"> Group 3
<INPUT type="radio" name="s-grp" value="oth"> Others
<BR><BR>
Which training program would you like to attend ?
<SELECT Name="pref">
<OPTION value="JS">JavaScript
<OPTION value="CORBA">CORBA
<OPTION value="VB6">Visual Basic 6
</SELECT>
</P>
<INPUT type="file" name="fnm" size=15 value="fonts.html">
<BR>
<INPUT type="button" name="but" value="Exit">
<INPUT type="Submit" Value="Save" name="s-but">
<INPUT type="reset" Value="Reset">
</FORM>
</BODY>
</HTML>

```

Example 14: Forms

Step 2: Open *prob1.html* in the browser and verify if the form is displayed as per the requirement.

Problem 2: Employee Details <<To Do>>

Problem Statement:

Design a web page *prob2.html* to accept the following employee details:

- Employee Name (Max 20 characters).
- Employee Code (Max 4 characters).
- Department (Use radio buttons).
- Date Of Join (Use the format dd/mm/yyyy).
- Address.
- Training programs attended (Use check boxes).
- Training programs need to attend (Use select box).
- Send the information at empinfo@patni.com.

Name	
Employee Code	
Department	<input type="radio"/> ADMIN <input type="radio"/> HR <input type="radio"/> TECHNICAL <input type="radio"/> ACCOUNTS
Date of Joining	dd mm yyyy
Address	
Training programs Attended	<input type="checkbox"/> HTML/DHTML <input type="checkbox"/> JAVA <input type="checkbox"/> CLIENT/SERVER <input type="checkbox"/> .NET
Training Programmes need to attend	JAVASCRIPT
Send the information at	empinfo@patni.com

Figure 26: Employee Details

Solution

1. Open **Editor**. Type the code and save the file as lab6\prob2.html.
2. Open the page in the browser.
3. Verify if the output is as per the figure.

Lab 7: Working with Frames

Goals	At the end of this lab session you will be able to: <ul style="list-style-type: none">• Understand the need for frames in web pages.• Create and work with frames.• Manage large content with frame.
Time	60 minutes

Problem 1: Frames

Problem Statement:

Create a web page which allows you to click on 2 hyperlinks courses, menu in the frame on the left. When you click a link, the details are displayed in the frame on the right. The file layout.html is loaded in the frame on the left.

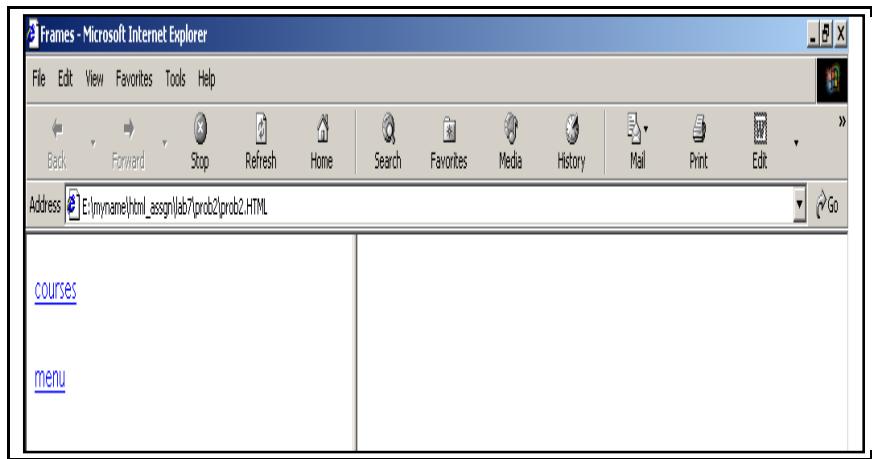


Figure 27: Frames

When you click the link “courses”, the details are displayed in the frame on the right.

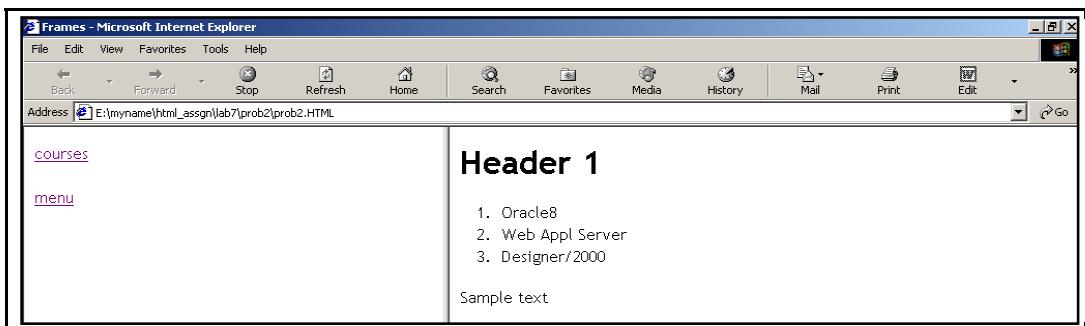
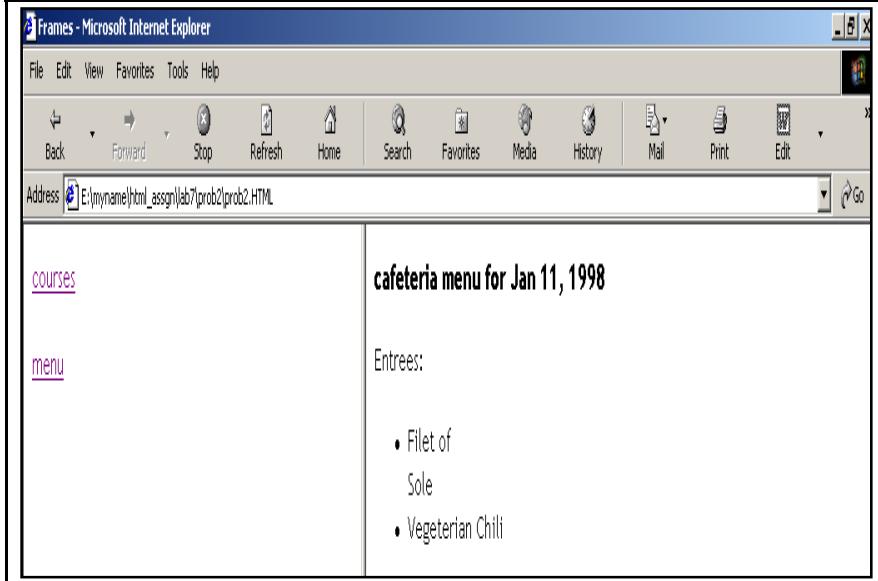


Figure 28: Courses

When you click the link "menu", the details are displayed in the frame on the right

**Figure 29: Menu**

Solution

Step 1: Write the following code in **Notepad** and save the file.

```
<html>
<head>
<title>Frames</title>
</head>

<frameset cols="40%, *" >
    <frame SRC="layout.html" >
    <frame SRC="" name="side-2">
</frameset>
</html>
```

Example 15: Frames (1)

Step 2: Write the following code in **Notepad** and save it as *lab7\prob2\layout.html*.

```
<html>
<head>
<title>frames</title>
</head>
<body>
<a href="courses.html" target="side-2"> courses </a>
<br>
```

```
<br>
<a href="menu.html" target="side-2"> menu </a>
</body>
</html>
```

Example 16: Frames (2)

Step 3: Write the following code in **Notepad** and save.

```
<html>
<head>
    <title>Sample Page</title>
</head>
<body >
<h1>Header 1</h1>
<ol>
    <li>Oracle8</li>
    <li>Web Appl Server</li>
    <li>Designer/2000</li>
</ol>

Sample text
</body>
</html>
```

Example 17: Courses

Step 4: Write the following code in **Notepad** and save.

```
<html>
<head>
<title>Cafeteria Menu Application</title>
</head>
<body>
<h3>cafeteria menu for Jan 11, 1998</h3>
<p>
Entrees:<br>
<ul>
<li>Filet of <br>Sole
<li>Vegeterian Chili
</ul>
</body>
</html>
```

Example 18: Menu

Step 5: Open the file *prob2.html* in the browser and check if the page works as per the requirement.

Lab 8: Cascading Style Sheets

Goals	At the end of this lab session you will be able to: <ul style="list-style-type: none">• Understand the basics of Cascading Style Sheets.• Use following types of style sheets:<ul style="list-style-type: none">◦ Inline◦ Embedded◦ Linked• Understand advanced Style Sheet Features.
Time	100 minutes

Problem 1: Inline Style Sheet

Problem Statement:

Design a web page using inline style sheet to format the contents of a page as shown in the figure below.

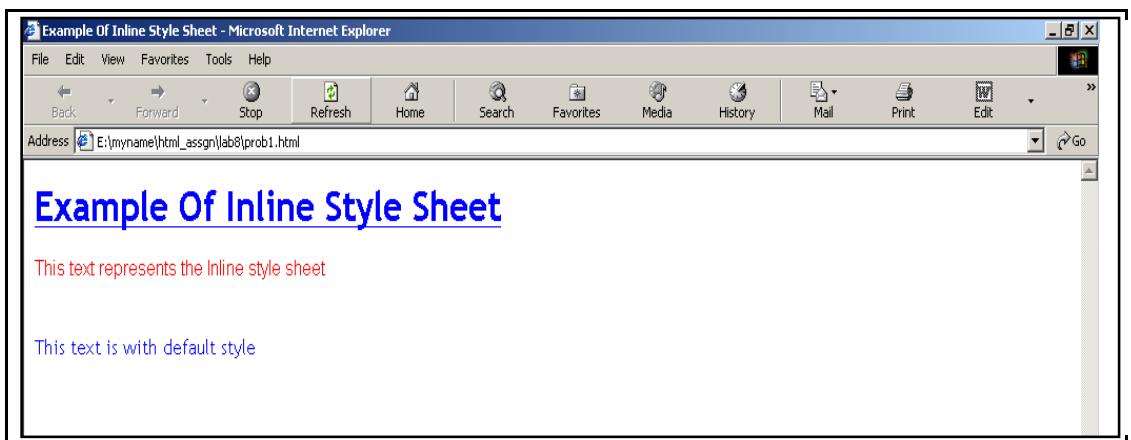


Figure 30: Inline Stylesheet

Solution

Step 1: Write the following code in **Notepad** and save.

```
<HTML>
<HEAD>
    <TITLE>Example Of Inline Style Sheet</TITLE>
<BODY bgcolor="black" text="yellow">
<H1 style="text-decoration: underline">Example Of Inline Style Sheet</H1>
<P style="font-family: arial, impact, sans serif; color:red">
This text represents the Inline style sheet
</P>
<BR>
<P>This text is with default style</P>
</BODY>
</HTML>
```

Example 19: Inline Style Sheet

Step 2: Open *prob1.html* in the browser and verify if the page is displayed as per the requirement.

Problem 2: Embedded Style Sheets

Problem Statement:

Design a web page using embedded style sheets to format the contents of a page as shown in the figure below.

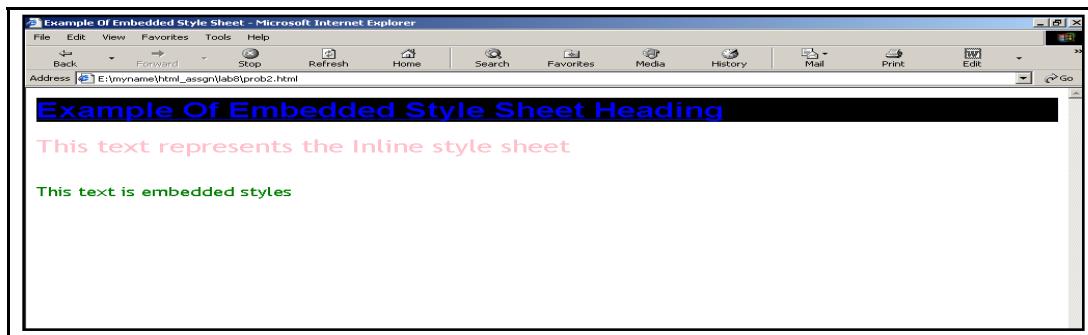


Figure 31: Embedded Stylesheet

Solution

Step 1: Write the following code in **Notepad** and save it as *lab8\prob2.html*.

```
<HTML>
<HEAD>
    <TITLE>Example Of Embedded Style Sheet</TITLE>
</HEAD>
<STYLE>
H1 {text-decoration: underline;
    font-family: arial;
    background: black;
    color: blue
}
P { font-size: 14pt; color: green}
</STYLE>

<BODY>

<H1>Example Of Embedded Style Sheet Heading</H1>

<P style="color: pink; font-size: 20pt" >
This text represents the Inline style sheet
</P>
<BR>
<P>This text is embedded styles
</P>
</BODY>
</HTML>
```

Example 20: Embedded Style Sheet

Step 2: Open *page* in the browser and verify if the page is displayed as per the requirement.

Problem 3: Linked Style Sheets**Problem Statement:**

Design a web page using linked style sheet to format the contents of the page shown in the example on Embedded Style Sheet (Problem 2). CSS file should be named “test.css”.

Text “Example of Embedded Style Sheet Heading” appears with a black background and text color as blue.

Use a *<P>* tag to place the text “This text represents the Inline style sheet”. The font size should be 20 points, and text color pink.

Using a *<P>* tag place text “This text is embedded styles”. The font size should be 14 points and text color should be green. This is the default for all *<P>* tags.

Solution

1. Open **Notepad**. Type the code and save the file.
2. Open **Notepad**. Type the code and save the file as lab8\test.css.
3. Start the Internet Explorer. On the **File** menu, click **Open**. **Open** dialog box appears. Click the **Browse** button and select *prob3.html* file.

<<Stretched Assignments>>

Problem 4: XHTML and Stylesheets

Design the webpage (800 X 600 px) as shown in figure using XHTML 1.0 and CSS with the sliced images given to you. (Images will be shared by the faculty) Webpage should cross browser complaint. Validate the webpage with XHTML strict.dtd using <http://validator.w3.org>

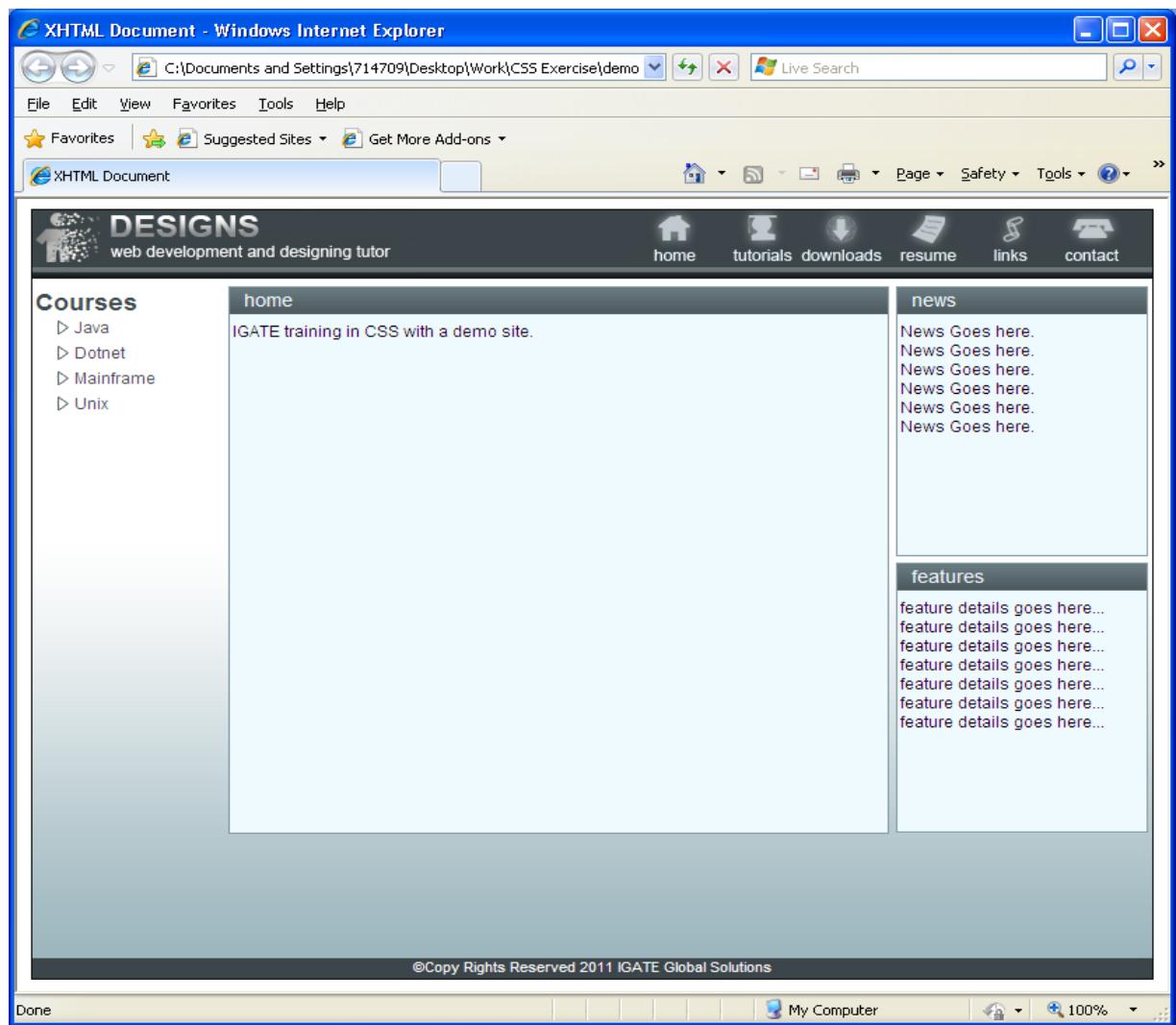


Figure 32: XHTML and Stylesheets

Appendix A: HTML Standards

Key Things To Keep In Mind:

- HTML standards help you reach the widest possible audience.
- There are many technologies that are *associated* with HTML because they are used on a Web page or in conjunction with HTML. But these technologies are *not* HTML:
 - CGI (Common Gateway Interface)
 - Java
 - JavaScript (JavaScript is also *not* Java)
 - Dynamic HTML (DHTML)
 - XML (Extensible Markup Language)
 - A variety of other emerging technologiesFor each of it, please follow the coding conventions, specified by that technology.
- Sometimes you need to break the rules and use non-standard syntax for good reasons. Try to keep this to a minimum.

How to Follow HTML Standards

Identify which version of HTML you are using in your document through the DOCTYPE line at the top of your file.

See the W3C site for more information on document types and DOCTYPE statements.

The important thing to remember is that a DOCTYPE statement is essential to assist validation software in checking your document.

- Use tools (supported by W3C) that support standards. In particular, install and use the *Tidy* program or *Tidy GUI* on your computer.
- Use W3C validation markup service to check the syntax of documents you create.
- Refer to W3C for technical and syntax information.

Some Simple HTML standards:

- The names of HTML files should always end with the ".html" extension.

Example:

Good: foo.html

Bad: foo.bar

- Always include a <HTML> tag at the very beginning and a </HTML> tag at the very end of your HTML documents.
- Always use the <HEAD> and </HEAD> tags to define a header section in your HTML documents.

- Always give your documents a title by using the <TITLE> and </TITLE> tags in the header section of your HTML documents.
- Always use the <BODY> and </BODY> tags to define the body in your HTML documents, which is everything in your document between the <HTML> and </HTML> that is not contained in your header section.
- Use the horizontal line tag <HR> to place a horizontal line beneath any prominent headers in your documents to help them stand out from the surrounding information.

Example:

```
<H1>My Document's Title</H1>
<HR>
```

- Always include a LINK with REV="MADE" in the header section of your HTML documents identifying you as the author.

Example:

```
<LINK REV="MADE" HREF="mailto:your_logonid@cs.niu.edu">
```

- Reasonable line lengths (no greater than 80 characters).
- Attributes associated with tags must be enclosed in quotes.

Example:

```

<table border= "0" cellpadding= "0" cellspacing="0">
<td width="200">
```

- Code is written in a consistent case. All command tags should be completely capitalized, in order for the tags to stand out better from the surrounding text.

Example:

Good: This text is emphasized.
Bad: This text is emphasized.

- All code should include comment tags for readability, particularly when nested tables are used.
- Images have *alt*, *height*, and *width* attributes. They must be placed in the same directory as the HTML files. These images must be referenced in the code as:

Example:

Good:
bad: .

- Links are coded correctly. All "HREF=" fields in anchor tags should always be enclosed in quotes.

Example:

Good:

Bad:

- Confirm that ©, ®, ™, and ℠ marks are coded correctly. These special characters should always be coded using their respective ASCII codes. It should also be confirmed that the superscription of these characters is done in a consistent manner.

Example :

Please code these special characters as follows:

and Ampersand: andamp;

© Copyright: and#169;

® Registration: and#174;

™ Trademark: and#153;

- Check links. There is nothing more frustrating to users than a broken link (except possibly the blink tag). If the review is of an entire site or a complete section of a site, it is helpful to use an automated link checker. Because there may be hundreds, or even thousands of links, the chance of missing one when checking them by hand is unacceptably high. Since Quality Assurance is not involved in the actual construction of a site, the producer/webmaster needs to verify that links are pointing to the correct pages that those pages still exist, etc.
- If you code a URL which does not specify a file name, always end the URL with a front slash (some browsers choke if you do not do this).

Example:

Good:

Good:

Bad:

Bad:

- Whenever possible, use logical formatting tags instead of physical —one. Let the client's browser figure out the best way to display the information.

Preferred: You should read the book <CITE>Neuromancer</CITE>

Preferred: This text should stand out

Discouraged: You should read the book <l>Neuromancer</l>

Preferred: This text should <BOLD>stand out</BOLD>

- Always "sign" any HTML documents that you create. Include a horizontal line and a link to your homepage (using the ADDRESS style) at the very bottom.

Example:

...and this is the end of my document's text.<P>

<HR>

 <ADDRESS>

WWW</ADDRESS>

</BODY>

</HTML>

- View pages on various browsers, including versions 3.x and 4.x. While it is prohibitively time consuming to check every page on every browser, several pages need to be reviewed on a wide variety of browsers to assure that the general design holds up.

Appendix B: Table of Figures

Figure 1: Open Notepad	5
Figure 2: HTML Code	5
Figure 3: Save contents as an .html file	6
Figure 4: Open the .html file from the browser	6
Figure 5: First.html in a browser	7
Figure 6: <i>MyFirstPage.html</i> Output.....	8
Figure 7: <i>Headers.html</i> Output	9
Figure 8: <i>Address.html</i> Output.....	10
Figure 9: <i>HorizontalRuler.html</i> Output.....	11
Figure 10: <i>PreFormattedText.html</i> Output.....	11
Figure 11: Resume Page.....	13
Figure 12: Table layout.....	14
Figure 13: Fruits Table	15
Figure 14: Product table	17
Figure 15: Calendar	17
Figure 16: Part (a)	19
Figure 17: Part (b)	20
Figure 18: Subject list	21
Figure 19: Big Company home page.....	23
Figure 20: About	24
Figure 21: Products	24
Figure 22: Contact	25
Figure 23 : Imagemap	27
Figure 24: Engine	28
Figure 25: Forms	30
Figure 26: Employee Details	32
Figure 27: Frames	33
Figure 28: Courses	34
Figure 29: Menu.....	34
Figure 30: Inline Stylesheet	37
Figure 31: Embedded Stylesheet	38
Figure 32: XHTML and Stylesheets.....	41

Appendix C: Table of Examples

Example 1: MyFirstPage.html.....	7
Example 2: Headers.html	8
Example 3: Address.html.....	9
Example 4: HorizontalRuler.html	10
Example 5: PreFormattedText.html.....	11
Example 6: Table Layout.....	15
Example 7: Fruit Table	17
Example 8: Types of Lists	21
Example 9: Big Company	25
Example 10: Products	25
Example 11: Welcome to Big Company	26
Example 12: Image Maps.....	29
Example 13: Bike Image.....	29
Example 14: Forms	31
Example 15: Frames (1)	34
Example 16: Frames (2)	35
Example 17: Courses	35
Example 18: Menu.....	35
Example 19: Inline Style Sheet.....	38
Example 20: Embedded Style Sheet.....	39