



## Department of Distance Education Punjabi University, Patiala

---

**Class : B.A. III (Computer Applications) Semester : 6**  
**Paper : (BAP 303) INTRODUCTION TO COMPUTER  
NETWORK AND INTERNET PROGRAMMING**

**Medium : English**

**Unit : II**

---

***Lesson No.***

- 2.1 : The Internet : www and Multimedia
- 2.2 : File Transfer Protocol
- 2.3 : HTML - I
- 2.4 : HTML - II
- 2.5 : Advanced HTML

---

***Department website : [www.pbidde.org](http://www.pbidde.org)***

### **The Internet: WWW and Multimedia**

#### **1.0 World Wide Web**

- 1.1 Internet Protocols
- 1.2 Hypertext and Links: The Motion of the Web
- 1.3 Pages on the Web
- 1.4 Retrieving Documents on the Web
- 1.5 How to Access the World Wide Web: Web Browsers
- 1.6 Extending the Browser: Plug-Ins
- 1.7 Beyond Plug-Ins: Active X

#### **2.0 Multimedia**

- 2.1 Delivery of Multimedia
- 2.2 Advantages of Multimedia
- 2.3 Suitability of Multimedia
- 2.4 The Types of Media
- 2.5 Different Ways to View Multimedia
- 2.6 Data, Audio and Video compression

#### **3.0 Summary**

#### **4.0 Keywords**

#### **5.0 Self Check Exercise**

#### **6.0 Suggested Readings**

#### **Objective:**

In this Lesson we will study the concept of World Wide Web, which is one of the most important services provided by the Internet. We will also study the meaning of Multimedia and its use on the Internet.

#### **1.0 Introduction: World Wide Web (WWW):**

The World Wide Web was developed in 1989 by Tim Berners-Lee of the European Particle Physics Lab (CERN) in Switzerland. The initial purpose of the Web was to use networked hypertext to facilitate communication among its members, who were located in several countries. Web was soon spread beyond CERN, and a rapid growth in the number of both developers and users ensued. In addition to hypertext, the Web began to incorporate graphics, video and sound. The use of the Web has

reached global proportions and has become a defining aspect of human culture in an amazingly short period of time.

World Wide Web has accelerated the growth of internet by giving it easy to use, “**point and click**” graphical interface. Users are attracted to the World Wide Web because it is interactive, easy to use, combines graphics, text, sound, animation making and it is a rich communication medium. The World Wide Web is many things to its millions of users. It is used as a market place, art gallery, library, school, publishing house and what ever else its author creates. The WWW provides a network of interactive documents and software to access them. It is based on documents called pages that combine text, pictures, forms, sound, animation and hypertexts links called hyperlinks. To navigate the WWW users “surf” from one page to another by using “**point and click**” operation on the hypertext in text and graphics.

The World Wide Web also referred to as the WWW or W3 or simply “the web”, is the universe of information available via the **Hypertext Transfer Protocol (HTTP)**. The web presents information as a series of “document” called as web pages that are prepared using the Hyper Text Mark-up Language (HTML). By making use of HTML, the documents author can specially code sections of the document to “point” to other information resources. These specially coded sections often referred to as “hypertext- links”. The user viewing a web page can select the hypertext links and can easily connect to the information resource that the links points to. These links lead to other documents, images, sounds, databases (like library catalogs) etc. While accessing the WWW, it is not necessary to follow a hierarchical path to information resources. Thus we can easily:

1. Jump from one link to another.
2. Jump to specific part of a document.
3. Go directly to a resource, if you know a URL (uniform resource locator).

So as the web page is not hierarchical and handles graphics, it offers a great deal of flexibility to the user.

### **1.1 Internet Protocols**

Almost every protocol type available on the Internet is accessible on the Web. **Internet Protocols** are sets of rules that allow for intermachine communication on the Internet. The following is a sample of major protocols accessible on the Web:

**E-mail** (Simple Mail Transport Protocol or SMTP)

Distributes electronic messages and files to one or more electronic mailboxes

**Telnet** (Telnet Protocol)

Facilitates login to a computer host to execute commands

**FTP** (File Transfer Protocol)

Transfers text or binary files between an FTP server and client

**Usenet** (Network News Transfer Protocol or NNTP)

Distributes Usenet news articles derived from topical discussions on newsgroups

**HTTP** (Hyper Text Transfer Protocol)

Transmits hypertext over networks. This is the protocol of the Web.

The World Wide Web provides a single interface for accessing all these protocols. This creates a convenient and user-friendly environment. Once upon a time, it was necessary to be conversant in these protocols within separate, command-level environments. The Web gathers these protocols together into a single system. Because of this feature, and because of the Web's ability to work with multimedia and advanced programming languages, the Web is by far the most popular component of the Internet.

### **1.2 Hypertext and Links: The Motion of the Web**

The operation of the Web relies primarily on hypertext as its means of information retrieval. Hypertext is a document containing words that connect to other documents. These words are called links and are selectable by the user. A single hypertext document can contain links to many documents. In the context of the Web, words or graphics may serve as links to other documents, images, video, and sound. Links may or may not follow a logical path, as each connection is created by the author of the source document. Overall, the Web contains a complex virtual web of connections among a vast number of documents, graphics, videos and sounds.

Producing hypertext for the Web is accomplished by creating documents with a language called Hypertext Markup Language, or HTML. With HTML, tags are placed within the text to accomplish document formatting, visual features such as font size, italics and bold and the creation of hypertext links. Graphics may also be incorporated into an HTML document

HTML is a sub language of Standard Generalised Markup Language. It is a system that defines and standardizes the structure of the document. Both SGML and HTML utilize descriptive markup to define the structure of an area of text.

HTML is standardized and portable. A document that has been prepared using HTML markup "tags" can be viewed using a variety of web browsers such as Netscape and Lynx. A browser interprets the tags in an HTML file and presents the file as a formatted, readable web page. In addition, HTML documents can be viewed on all types of system, such as Macintosh, PC and UNIX machines.

### **1.3 Pages on the Web**

A web page is a single unit of information, often called a document that is available via the World Wide Web. A web page can be longer than one computer screen and can use more than one piece of paper when it is printed out. A web page

is created using HTML. It consists of standardized codes or “tags” that are used to define the structure of information on a web page. These codes enable web pages to have many features including bold text, italic text, headings, paragraph breaks and numbered or bulleted list. A web page can be created by user activity. For example, if you visit a Web search engine and enter keywords on the topic of your choice, a page will be created containing the results of your search. In fact, a growing amount of information found on the Web today is served from databases, creating temporary Web pages "on the fly" in response to user queries.

Access to Web pages may be accomplished by:

1. Entering an Internet address and retrieving a page directly
2. Browsing through pages and selecting links to move from one page to another
3. Searching through subject directories linked to organized collections of Web pages
4. Entering a search statement at a search engine to retrieve pages on the topic of your choice

#### **1.4 Retrieving Documents on the Web: The URL and Domain Name System**

URL stands for Uniform Resource Locator. The URL specifies the Internet address of a file stored on a host computer connected to the Internet. Every file on the Internet has a unique URL. Web browsers use the URL to retrieve the file from the host computer and the specific directory in which it resides. This file is downloaded to the user's client computer and displayed on the monitor connected to the machine.

URLs are translated into numeric addresses using the **Domain Name System (DNS)**. The DNS is a worldwide system of servers that stores location pointers to Web sites. The numeric address, called the IP (Internet Protocol) address, is actually the "real" URL. Since numeric strings are difficult for humans to use, alphanumeric addresses are employed by end users. Once the translation is made by the DNS, the browser can contact the Web server and ask for a specific file located on its site.

#### **Anatomy of a URL**

This is the format of the URL:

**Protocol://host/path/filename**

For example, this is a URL on the Web site of the U.S. House of Representatives:

**[http://www.house.gov/house/2004\\_House\\_Calendar.html](http://www.house.gov/house/2004_House_Calendar.html)**

This URL is typical of addresses hosted in domains in the United States.

Structure of this URL:

1. Protocol: **http**
2. Host computer name: **www**
3. Second-level domain name: **house**
4. Top-level domain name: **gov**

5. Directory name: **house**
6. File name: **2004\_House\_Calendar\_html**

Note how much information about the content of the file is present in this well-constructed URL.

Several top-level domains (TLDs) which are common:

.Com	commercial enterprise
.Edu	educational institution
.Gov	U.S. government entity
.mil	U.S. military entity
.net	network access provider
.org	usually nonprofit organizations
.biz	business
.info	information
.mobi	mobile devices
.coop	Cooperatives
.aero	air transport industry

New domain names were approved in November 2000 by the Internet Corporation for Assigned Names and Numbers (ICANN): **.biz**, **.museum**, **.info**, **.pro** (for professionals), **.name** (for individuals), **.aero** (for the aerospace industry), and **.coop** (for cooperatives). ICANN continues to investigate proposals for adding additional domain names, for example, **.mobi** for sites designed for mobile devices, and **.jobs** for the human resources community.

In addition, dozens of domain names have been assigned to identify and locate files stored on host computers in countries around the world. These are referred to as **two-letter Internet country codes**, and have been standardized by the International Standards Organization as ISO 3166. For example:

In	India
Au	Australia
Jp	Japan
Uk	United Kingdom

### 1.5 How to Access the World Wide Web: Web Browsers

To access the World Wide Web, you must use a Web browser. A browser is a software program that allows users to access and navigate the World Wide Web. There are two types of browsers:

**1. Graphical:** Text, images, audio, and video are retrievable through a graphical software program such as Internet Explorer, Firefox, Chrome, Mozilla and Opera. These browsers are available for Windows, Linux and other operating systems.

Navigation is accomplished by pointing and clicking with a mouse on highlighted words and graphics.

You can install a graphical browser on your computer. For example, Internet Explorer is a part of the Windows operating system, and is also available on the Microsoft site: <http://www.microsoft.com>.

**2. Text:** Lynx is a browser that provides access to the Web in text-only mode. Navigation is accomplished by highlighting emphasized words in the screen with the arrow up and down keys, and then pressing the forward arrow (or Enter) key to follow the link. In these days of graphical browsers, it may be hard to believe that Lynx was once very popular.

### **1.6 Extending the Browser: Plug-Ins**

When the browser encounters a sound, image or video file, it hands off the data to other programs, called **plug-ins**, to run or display the file. Working in conjunction with plug-ins, browsers can offer a seamless multimedia experience. Many plug-Ins are available for free. File formats requiring plug-ins are known as **MIME types**. MIME stands for Multimedia Internet Mail Extension and was originally developed to help e-mail software handle a variety of binary (non-ASCII) file attachments.

A common plug-in utilized on the Web is the Adobe Acrobat Reader. The Acrobat Reader allows you to view documents created in Adobe's Portable Document Format (PDF). These documents are the MIME type "application/pdf" and are associated with the file extension .pdf. When the Acrobat Reader has been downloaded to your computer, the program will open and display the file requested when you click on a hyperlinked file name with the suffix .pdf. The latest versions of the Acrobat Reader allow for the viewing of documents within the browser window. Web browsers are often standardized with a small suite of plug-ins, especially for playing multimedia content. Additional plug-ins may be obtained at the browser's Web site, at special download sites on the Web or from the Web sites of the companies that created the programs.

### **1.7 Beyond Plug-Ins: Active X**

**ActiveX** is a technology developed by Microsoft which makes plug-ins less necessary. ActiveX offers the opportunity to embed animated objects, data, and computer code on Web pages. A Web browser supporting ActiveX can render most items encountered on a Web page. As just one example, Active X allows you to view and edit PowerPoint presentations directly within your Web browser. ActiveX works best with Microsoft's Internet Explorer.

## **2. Multimedia**

Multimedia is the integration of multiple forms of media. This includes text, graphics, audio, video etc. It is a method of instruction that uses the computer to

present information with text, graphics, audio or video. It is referred to as Computer Based Training (CBT) or web based training (WBT). It can take the form of training, presentations, software simulation, job simulation, testing reference or online help. Since multimedia instruction is often delivered over the internet. It can look just like set of web pages, but it can also be delivered on CD-ROM, on diskette or over a Local Area Network (LAN) or company intranet.

To actually see, hear or otherwise experience these media, you need a computer equipped with the right hardware such as sound card, enough memory to make movies play smoothly, a big enough hard drive to store large file-formats files and so on. And you need software installed on your computer, either as a part of a web browser or as a stand alone program that can interpret and display—or just “play”—the various media file formats. If you can put together all the ingredients, then you can start to experience the internet as the world’s largest CD-ROM, with new content appearing online daily.

Every web browser can at least download files. So if you can get a stand-alone program to interpret a multimedia file you have downloaded, then you can experience that medium even without a sophisticated web browser.



## 2.1 Delivery of Multimedia

There are two basic methods for delivering multimedia.

1. **CD-ROM** has become the most cost effective method of delivery for multimedia materials. These devices are used to store large amounts of some combination of text, graphics, sound and moving video.
2. **Distributed networks** allow computers to be connected to other computers which can provide an almost endless array of multimedia material. The internet is currently providing a distributed network approach to the delivery of multimedia material.

## 2.2 Advantages of Multimedia:

The advantages of Multimedia are as under:

- **It's interactive:** Learners are actively engaged. Interactive may include multiple choice questions, fill-in-the-blanks, selecting all right answers or matching by dragging or dropping graphics. Feedback engages the learner even more. For example, an incorrect answer could provide a hint along with a "sorry that's incorrect" response.
- **It is tailored to learner needs:** Learner can take the training when they want, as often as they want, and at a pace that's right for them.
- **It is cost effective:** Companies don't need to pay for trainers, travel or classrooms. It is especially useful in situations where the users are spread over the wide area.
- **It is efficient and effective:** Learners typically learn more in a shorter period of time with multimedia instruction, as learners are more interested in multimedia messages which combine the element of text, audio, graphics and video.
- **It is easy to revise and update:** Especially with internet delivery, changes made to one copy are immediately available to all learners. For example, a company that manufactures car utility racks can easily alter installation instructions for consumers as car manufacturers introduce new car models every year.

## 2.3 Suitability of Multimedia

- **Cost:** The expense can vary widely. One can find off-the-shelf or off-the-web training at very reasonable cost, and sometime even for free. Custom development of computer based training can be expensive, but may be worth it if employee or customer training is crucial.
- **Number of Learners:** A very large audience, such as a large customer base, may reduce costs greatly and possibly allow delivering instruction more effectively and economically than with paper based instruction.
- **Location of Learners:** If the learners are spread over broad geographical areas, such as with sales reps or traveling technicians, CBT or WBT may be the only way to provide accurate and effective training to everyone.

- **Availability of Computers:** Equipment doesn't have to be sophisticated –all you may need is an internet connection.

## 2.4 The Types of Media

### 1. Pictures

The first graphical web browser, Mosaic, could display only one picture format when it first appeared-CompuServe's **GIF** (Graphic Interchange Format), which is a compressed file format. The other major picture format is called **JPEG** (named for the Joint Photographics Experts Group that designed the format). At first, Mosaic could only display JPEG in a helper application. When Netscape came along, it supported in-line JPEG, which now most browsers can now also handle. JPEG can be compressed to much smaller file sizes than equivalent GIFs, but more they are "squished" the worse the quality of the image becomes.

### 2. Sounds and Music

There are many different file formats available on the Net. The most common of these include Microsoft's **.WAV** (Wave) format, perhaps the most widespread; the Macintosh **AIFF** format; and the UNIX (originally NeXT) **.au** format. Many websites offer sounds in more than one format, in order to make it easier for each user to download a file format local to their type of computer.

Other sound format includes **MIDI** (Musical Instrument Digital Interface); the Amiga **SND** format and **MP2 or MPA**, which are MPEG formats. (MPEG is a movie format, but movies, of course, often also includes sounds, so the MPEG standard specifies a sound format as well.

A new approach to sound files (for movies files as well) is Streamlining. Streamlining is when files are sent a little at a time and start playing almost immediately. This model differs from those in which an entire file is sent and a start playing only after it has been completely downloaded. The most popular streaming format these days is called **Real Player or Real Audio**. This format allows sounds to be broadcast something like the way they are in radio. Microsoft has also developed a competing sound- streaming format called **Truesound**.

### 3. Movies and Animations

As with sounds and pictures, there are various competing movie and animation formats available on the Net. Technically the difference between the movie and an animation is that movies use video or film images (variations on photographic technology) while animations use drawn illustrations.

Probably the most widespread movie format is the **MPEG** (Motion Pictures Experts Group) format, a compressed format. Another popular format is **QuickTime**, which started on the Macintosh platform but can now be displayed on most computers. QuickTime files usually have a **.qt or .mov** extension. A third common movie format, native to the windows platform is **.avi**.

#### 4. 3D Environments

The future of the Net may be glimpsed in the still-evolving 3D formats with which real or imaginary spaces are depicted in perspective and the user has the ability to move around, viewing the space from multiple angles.

The most common format for 3D worlds on the web is called **VRML**, which stands for Virtual Reality Modeling Language. While VRML is an evolving standard, there are multiple competing implementations of it out there, including a version of QuickTime called, naturally enough, QuickTime VR. VRML files have usually **.wrl** extension. One of the future goals of VRML developers is to create worlds in which many users can meet and interact, as if in person. There are also some special browsers out there, such as one called WebFX, designed specifically for viewing and moving through 3D spaces.

### 2.5 Different Ways to View Multimedia

#### 1. Plug-ins

The most sophisticated way to work with multimedia files is to plug special add-on software directly into a browser. Such a program, usually called a plug-in, is an application that works in tandem with another program, enhancing its features as if you had taken a piece of hardware and added it to your computer to give it more features. Only Netscape Navigator has really exploited this approach fully. Its nearest competitor, Microsoft Internet Explorer, can handle only one specific add-in, what Microsoft, which always use its own term, called plug-ins. With plug-ins installed, a browser can then display an unusual format in-line, meaning inside the browser window, instead of launching an external program to display the file.

#### 2. Viewers and Players

A viewer program is one that can be used to view or play a specific type of file. Even with the character-based browser like the UNIX program Lynx, you can still download files. It's true that you will then have to get the file from UNIX account to your desktop computer, but after that, you can "play" the file downloaded if you have the appropriate software (such as sound recorder for windows) installed on your PC. This approach will also work with browsers. Such as AOL's, that are not equipped to launch external programs automatically

#### 3. Compressed Files and Streaming Media

One of the strategies used to address bandwidth limitations is to compress files as much as possible, most often using widely accepted compression standards. This might mean converting media files, such as images, to compressed formats or it might mean compressing the original files with a "Zip" type program, and requiring that the recipient decompress it herself.

#### 4. Helper Applications

Mosaic, Netscape Navigator and other browsers based on the graphical model have the ability to launch external programs called helper applications--when a non standard file format is selected. Helper applications will let your browser open files in formats it could not otherwise handle, such as a Sun audio file. They do however; have to be “taught” where to look for the helper application. You can either do this in advance, by entering the options or preferences area of the browser and looking for the helper applications or you can attempt to download a media file and then when the browser tells you it doesn’t recognize the file format, you can educate it about which viewer to use with that type of file. You do this simply by typing the path and file name for the correct program or by clicking on a browse button and rummaging around on your hard disk for program you need. After that, your browser will automatically launch the right helper application whenever you select that type of media file again.

#### 2.6 Data, Audio and Video compression

Multimedia refers to any computer-based presentation or application that integrates two or more of the following elements: text, graphics, animation, audio, video and virtual reality. A PowerPoint presentation combining text, graphics and video clips is an example of a multimedia application. It is a noninteractive multimedia presentation in which the slides are displayed in a predefined linear fashion. Interactive multimedia applications accept input from the user by means of a keyboard, voice or mouse and perform an action in response. An interactive multimedia program allows users to select the material, define the order in which it is presented and obtain feedback on their actions. The ability for users to interact with a multimedia application is one of its more unique and important features that enhances learning by engaging and challenging users.

**Text** is a fundamental element used to convey information. It can be enhanced by a variety of textual effects to emphasize and clarify information, e.g., font size, color, style or special effects, such as shadow, blinking, bouncing or fading.

A **graphic** is a digital representation of non-text information, such as a drawing, charts or photographs. Graphics were the first media used to enhance the text-based Internet. The introduction of graphical Web browsers allowed Web page developers to incorporate illustrations, logos and pictorial navigation into Web pages. Graphics files on the Web must be saved in a certain format. The two most common file formats for graphical files are **JPEG** (Joint Photographic Experts Group) and **GIF** (Graphics Interchange Format). Files are saved in both JPEG and GIF format using compression techniques to reduce the file size for faster downloading from the Web.

JPEG is designed for compressing full-color, gray scale images or continuous-tone artwork. Any smooth variation in color, such as occurring in highlighted or

shaded areas, will be represented more faithfully and in less space by JPEG than by GIF. GIF does significantly better on images with only a few distinct colors, such as line drawings and simple cartoons. Plain black-and-white images should never be converted to JPEG. There has to be at least 16 gray levels before JPEG is useful for gray-scale images.

JPEG is *lossy*, meaning that the decompressed image isn't quite the same as the original. A *lossless* compression algorithm is one that guarantees its decompressed output to be bit-for-bit identical to the original input. This scheme does not discard any data during the encoding process, while the lossy scheme throws useless data away during encoding. That is, in fact, how lossy schemes manage to obtain superior compression ratios over most lossless schemes. JPEG was designed specifically to discard information that the human eye cannot easily see. Because the human eye is much more sensitive to brightness variations in gray-scale than to color variations, JPEG can compress color data more heavily than brightness data. Gray-scale images do not compress well by large factors. It should be noted that GIF is lossless for gray-scale images of up to 256 levels, while JPEG is not. However, the more complex and subtly rendered the image, the more likely that JPEG will do well.

There are two good reasons for using JPEG: to make image files smaller and to store 24 bits/pixel color. Using smaller image files makes it easier to transmit files across networks and saves storage space when creating archiving libraries. JPEG stores full color information 24 bits/pixel (16 million colors) unlike GIF, which can store only 8 bits/pixel (256 or fewer colors). Hence JPEG is considerably more appropriate than GIF for use as a WWW standard photo format. However, it takes longer to decode and view a JPEG image than to view an image of a simpler format such as GIF. A useful property of JPEG is that adjusting compression parameters can vary the degree of lossiness. This means that the image maker can trade file size against output image quality.

JPEG is a useful format for compact storage and transmission of images, but it should not be used as an intermediate format for sequences of image manipulation steps. The disadvantage of lossy compression with JPEG is that if the user repeatedly compresses and decompresses an image, a little more quality is lost each time. A lossless 24-bit format such as PNG (Portable Network Graphics), TIFF (Tagged Image File Format) or PPM (Portable PixMap) should be used while working on the original image and then JPEG format can be used when the file is ready to store or send out on the net. If the user expects to edit the image again in the future, the original lossless master copy should be used. Both GIF and PNG support transparent backgrounds, while JPEG does not have this ability. This issue is currently making the GIF image very popular with Web page creators and designers.

Because graphics files can be time consuming to download, some Web sites use thumbnails on their pages. A **thumbnail** is a small version of a larger image that the user can click on to display the full-sized image. The graphical elements for computer applications can be obtained from a clip art/image gallery, which is a collection of previously created clip art and photographs grouped by themes. Graphics can also be created using paint/image editing software, for example, Windows Paint, PaintShop Pro or Adobe Photoshop. Photographs for multimedia applications can be obtained using a scanner to digitize photos, taking the photographs with a digital camera or buying them in a photograph collection on a CD-ROM. Graphics can also be downloaded from the Web.

Displaying a series of still graphics in rapid sequence creates an **animation**, which is a graphic that has the illusion of motion. Animated graphics make Web pages more visually attractive and draw attention to important information or links. There are graphics animation and authoring software packages that allow creation of animations. One popular type of animation, called an animated GIF, is created using special software (e.g., Ulead GIF Animator freeware) to combine several images into a single GIF file. With GIF Animator the user can create pictures, banners, buttons and even movies using the built-in image editor or any stand-alone graphics editor, to create image files that will be assembled into GIF animation. Also, the user can obtain previously created animations from a CD-ROM or download them from the Web.

**Audio** is music, speech or any other sound. Using audio in a multimedia application to supplement text and graphics enhances understanding. Audio for multimedia can be obtained in several ways. One method is to capture the sound using a microphone, CD-ROM, radio, musical device or any other audio input device that is plugged into a port on a sound card. As with graphics and animation, audio clips can be purchased on a CD-ROM, DVD-ROM or downloaded from the Web. Audio files can be quite large, and therefore they are frequently compressed to reduce the size and increase the speed of downloading. MP3 is a popular technology to compress audio. Files compressed using this format have an extension of .mp3 and have to be downloaded completely. Most currently used browsers contain a program called a **player** that can play the audio in MP3 files. Web applications also use streaming audio that allows the user to listen to the sound as it downloads to the computer. **Streaming** is the process of transferring data in a continuous and even flow, which allows users to access and use a file before it has been transmitted completely. Streaming is important because most users do not have fast enough Internet connections to download a large multimedia file quickly. Web-based audio can be also used for Internet telephone service, sometimes called Internet telephony. This technology allows users to talk to other people over the Internet. Internet

telephony uses the Internet, instead of a telephone network, so the cost of the communication is reduced to the cost of the Internet connection. Internet telephone software and the computer's sound card digitize and compress the conversation and then transmit the digitized audio over the Internet to the called parties.

**Video** consists of photographic images that are played back at speeds of 15-30 frames per second and provide the appearance of full motion. To use video in a multimedia application, the developer has to capture, digitize and edit the video segments using special video production hardware and software. Video can also be captured directly in digital format using a digital video camera. Due to the size of video files, incorporating video into a multimedia application is often a challenge. Files require large amounts of storage space, therefore they are often compressed. Video compression works by recognizing that only a small portion of the video image changes from frame to frame, and after storing the first reference frame only changes from one frame to the next are stored. The Motion Pictures Experts Group has defined a standard for video and audio compression and de-compression, called **MPEG**. MPEG compression can reduce the size of video files up to 95 percent, while retaining near TV quality. Video compression has allowed video to play a more important role in multimedia applications. Technologies such as streaming video made video a viable part of multimedia on the Web. As with streaming audio, streaming video allows the user to view longer or live video images as they are downloaded to the computer. The standard used for transmitting video data on the Internet is RealVideo, which is a component of RealPlayer supported by most current Web browsers. Streaming video also allows conducting Internet videoconferences that work like Internet telephony. A video camera, videoconferencing software and video capture card digitize and compress the images and sounds. After they are sent over the Internet, equipment and software at the receiving end assemble and decompress the data presenting the images and sound as video. The Synchronized Multimedia Integration Language (SMIL, pronounced "smile") enables simple authoring of interactive audiovisual presentations. SMIL is typically used for "rich media/multimedia" presentations, which integrate streaming audio and video with other media type. SMIL is an HTML-like language and many SMIL presentations are written using a simple text-editor.

Another important application of multimedia is to create **simulations**, which are computer-based models of real-life situations. Multimedia simulations often replace costly and sometimes dangerous demonstrations and training, such as in chemistry, nuclear physics, aviation and medicine. Also, multimedia simulations are used in the game industry. **Virtual reality** is the simulation of a real or imagined environment that appears as a three-dimensional (3-D) synthetic space that has dynamic properties specified by software. On the Web, virtual reality involves the

display of 3-D images that the user can explore and manipulate interactively. Most Web-based virtual reality applications are developed using virtual reality modeling language (VRML), which is a language that describes the geometry of the scene. Using VRML, a developer can create an entire 3-D site, called a virtual reality world containing infinite space and depth. Virtual reality has many practical applications in science, education, advertising, design and other fields.

Some of the multimedia on the Web is developed in **Java**, which is a programming language specifically designed for use on the Internet. Developers use Java to create stand-alone applications or programs called **applets** that can be downloaded and run in the window of any Java-enabled browser. An applet is a short program executed inside of another program that runs on the user's computer.

Most Internet browsers have the capability of displaying basic multimedia elements on a Web page. Sometimes the browser needs an additional program called a plug-in or helper application, which extends the capability of the browser. A **plug-in** runs multimedia elements within the browser window, while a **helper application** runs multimedia elements in a window separate from the browser. Plug-ins and helper applications can be downloaded or copied at no charge from many sites on the Web. Usually, Web pages that use multimedia include links to Web sites that contain the required plug-in or helper. Some browsers include commonly used plug-ins, such as Shockwave. To view the virtual world, the user needs a VRML browser or a VRML plug-in to a Web browser.

Combining media for Web applications brings Web pages to life, increasing the types of information available on the Web, expanding the Web's potential uses and making the Internet a more entertaining place to explore.

### **3.0 Summary**

In this Lesson we have studied the concept of World Wide Web, which is one of the most important services provided by the Internet. We have also studied the meaning of Multimedia and its use on the Internet.

### **4.0 Keywords**

Internet, DNS, IP address, TCP/IP, Protocols, Plug-ins, Multimedia, Browsers, HTTP.

### **5.0 Self Check Exercise**

- Q What is Multimedia? Explain the various types of media and different ways to view Multimedia.
- Q Write an essay on World Wide Web.
- Q Write down short notes on the following:
1. Domain Name System (DNS)
  2. IP address

3. URL
4. Multimedia
5. WWW

**6.0 Suggested Readings**

- "Internet for everyone" by Leon & Leon, APH publishing corporation, New Delhi
- "The ABCs of Internet" by Christian Crumlish, Sybex
- "The Internet Book" by Douglas Comer, Prentice Hall.
- Computer network & internets by D.E. Comer, Pearson Education

### **File Transfer Protocol**

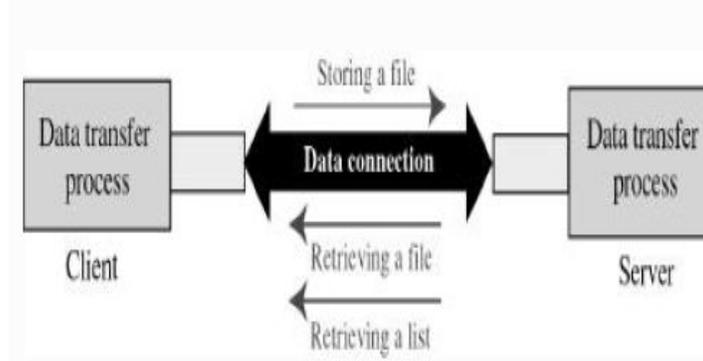
- 1.0 File Transfer Protocol**
  - 1.1 Objectives of FTP
  - 1.2 Models for FTP
  - 1.3 Criticisms of FTP
- 2.0 Data Transfer Function**
  - 2.1 Data Representation and Storage
  - 2.2 Data Type
  - 2.3 Establishing Data Connection
  - 2.4 FTP in Review
- 3.0 FTP Directories**
  - 3.1 Moving Around FTP Directories
  - 3.2 Some Useful FTP Commands
- 4.0 Modes of Ftp**
  - 4.1 Files modes
  - 4.2 FTP and web browsers
- 5.0 Generalized File Transfer**
- 6.0 Batch & Interactive Transfer**
  - 6.1 FTP Client-Server Interactions
  - 6.2 Characteristics of Clients and Servers
- 7.0 Summary**
- 8.0 Keywords**
- 9.0 Self Check Exercise**
- 10.0 Suggested Readings**

**Objective:**

In this lesson, we will discuss some fundamental concepts and terms used in the evolving language of File Transfer Protocol. We will also discuss some important concept about FTP as well as some important commands used in FTP login. In this lesson, we will also discuss the important concept of client and server connection.

### 1.0 File Transfer Protocol (FTP) :

**FTP or File Transfer Protocol** is a commonly used protocol for exchanging files over any network that supports the TCP/IP protocol (such as the Internet or an intranet). There are two computers involved in an FTP transfer: a server and a client. The FTP server, running FTP server software, listens on the network for connection requests from other computers. The client computer, running FTP client software, initiates a connection to the server. Once connected, the client can do a number of file manipulation operations such as uploading files to the server, download files from the server, rename or delete files on the server and so on. Any software company or individual programmer is able to create FTP server or client software because the protocol is an open standard. Virtually every computer platform supports the FTP protocol. This allows any computer connected to a TCP/IP based network to manipulate files on another computer on that network regardless of which operating systems are involved (if the computers permit FTP access). There are many existing FTP client and server programs, and many of these are free.



**Figure1:- File sharing**

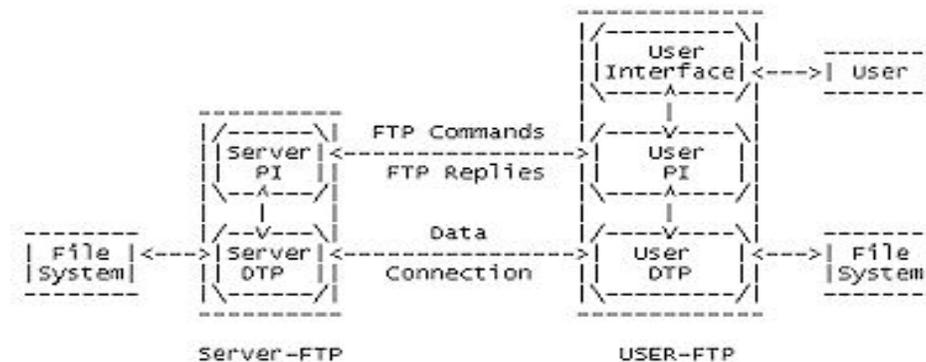
In other words “**File Transfer Protocol (FTP)** is a standard network protocol used to exchange and manipulate files over a TCP/IP based network, such as the Internet. FTP is built on a client –server architecture and utilizes separate control and data connections between the client and server applications. Client applications were originally interactive command-line tools with standardized command syntax, but graphical user interfaces have been developed for all desktop operating systems in use today. FTP is also often used as an application component to automatically transfer files for program internal functions. ”

#### 1.1 Objectives of FTP are as follows:

- 1) To promote sharing of files (computer programs and/or data)
- 2) To encourage indirect or implicit (via programs) use of remote computers.
- 3) To shield a user from variations in file storage systems among hosts.
- 4) To transfer data reliably and efficiently.

**Some basic FTP terms.**

- **Anonymous FTP:** Transfers files from the public portion of an FTP server. "Anonymous" means that you don't have to have an account on the server. In most cases, use anonymous as your user name and your email address as your password.
- **Archive:** An FTP site that contains a selection of files for download.
- **Download:** Also called "Get". Copy a file from an FTP site to another computer. If you're merely downloading shared files an anonymous account is usually sufficient. However, if you're downloading Web pages for update, a password and user privileges is usually required.
- **FTP site:** A Web site that stores files for download. You can access the sites with a Web browser by typing in the address. All FTP site addresses begin with ftp:// (instead of http://).
- **Upload:** Also called "Put". Place files on an FTP server. Upload privileges are usually password protected to keep unauthorized users from placing files that could contain viruses or other malicious code on the server.

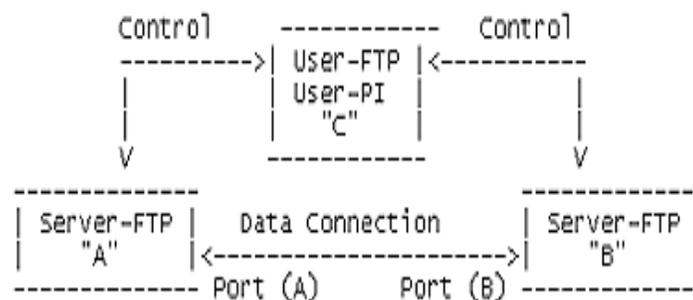
**1.2 Models for FTP:****Figure 2:- Model for FTP Use**

In the model described in Figure 2, the user-protocol interpreter initiates the control connection. The control connection follows the Telnet protocol. At the initiation of the user, standard FTP commands are generated by the user-PI and transmitted to the server process via the control connection. (The user may establish a direct control connection to the server-FTP, from a TAC terminal for example, and generate standard FTP commands independently, bypassing the user-FTP process.) Standard replies are sent from the server-PI to the user-PI over the control connection in response to the commands.

The FTP commands specify the parameters for the data connection (data port, transfer mode, representation type, and structure) and the nature of file system

operation (store, retrieve, append, delete, etc.). The user-FTP or its designate should "listen" on the specified data port and the server initiate the data connection and data transfer in accordance with the specified parameters. It should be noted that the data port need not be in the same host that initiates the FTP commands via the control connection, but the user or the user-FTP process must ensure a "listen" on the specified data port. It ought to also be noted that the data connection may be used for simultaneous sending and receiving.

In another situation a user might wish to transfer files between two hosts, neither of which is a local host. The user sets up control connections to the two servers and then arranges for a data connection between them. In this manner, control information is passed to the user-PI but data is transferred between the server data transfer processes. Following is a model of this server-server interaction.



**Figure 3:- FTP**

The protocol requires that the control connections be open while data transfer is in progress. It is the responsibility of the user to request the closing of the control connections when finished using the FTP service, while it is the server who takes the action. The server may abort data transfer if the control connections are closed without command.

### 1.3 Criticisms of FTP:

1. Passwords and file contents are sent in clear text, which can be intercepted by eavesdroppers.
2. No integrity check on the receiver side. If transfer is interrupted the receiver has no way to know if the received file is complete or not. It is necessary to manage this externally for example with MD5 sums or cyclic redundancy checking.
3. FTP is an extremely high latency protocol due to the number of commands needed to initiate a transfer.
4. Multiple TCP/IP connections are used, one for the control connection, and one for each download, upload, or directory listing. Firewall software needs additional logic to account for these connections.

5. It is hard to filter active mode FTP traffic on the client side by using a firewall, since the client must open an arbitrary port in order to receive the connection. This problem is largely resolved by using passive mode FTP.

### **Security problems**

FTP is an inherently insecure method of transferring files because there is no way for the original FTP specification to transfer data in an encrypted fashion. What this means is that under most network configurations, user names, passwords, FTP commands and transferred files can be "sniffed" or viewed by someone else on the same network using a protocol analyzer (or "sniffer"). It should be noted that this is a problem common to many Internet protocols written prior to the creation of SSL such as HTTP, SMTP and Telnet. The common solution to this problem is to use SFTP (SSH File Transfer Protocol) which is based on SSH or FTPS (FTP over SSL), which adds SSL or TLS encryption to FTP.

### **2.0 Data Transfer Function:**

Files are transferred only via the data connection. The control connection is used for the transfer of commands, which describe the functions to be performed, and the replies to these commands. Several commands are concerned with the transfer of data between hosts. These data transfer commands include the MODE command which specify how the bits of the data are to be transmitted and the STRUCTURE and TYPE commands, which are used to define the way in which the data are to be represented. The transmission and representation are basically independent but the "Stream" transmission mode is dependent on the file structure attribute and if "Compressed" transmission mode is used, the nature of the filler byte depends on the representation type.

### **2.1 Data Representation and Storage:**

Data is transferred from a storage device in the sending host to a storage device in the receiving host. Often it is necessary to perform certain transformations on the data because data storage representations in the two systems are different. For example NVT-ASCII has different data storage representations in different systems. DEC TOPS-20s's generally store NVT-ASCII as five 7-bit ASCII characters, left-justified in a 36-bit word. It is desirable to convert characters into the standard NVT-ASCII representation when transmitting text between dissimilar systems. The sending and receiving sites would have to perform the necessary transformations between the standard representation and their internal representations.

A different problem in representation arises when transmitting binary data (not character codes) between host systems with different word lengths. It is not always clear how the sender should send data and the receiver store it. For example, when transmitting 32-bit bytes from a 32-bit word-length system to a 36-bit word-length system, it may be desirable (for reasons of efficiency and usefulness) to store

the 32-bit bytes right-justified in a 36-bit word in the latter system. In any case, the user should have the option of specifying data representation and transformation functions. It should be noted that FTP provides for very limited data type representations. Transformations desired beyond this limited capability should be performed by the user directly.

### **2.2 Data Type:**

Data representations are handled in FTP by a user specifying a representation type. This type may implicitly (as in ASCII or EBCDIC) or explicitly (as in Local byte) define a byte size for interpretation which is referred to as the "logical byte size."

### **2.3 Establishing Data Connection:**

The mechanics of transferring data consists of setting up the data connection to the appropriate ports and choosing the parameters for transfer. Both the user and the server-DTPs have a default data port. The user-process default data port is the same as the control connection port (i.e., U). The server-process default data port is the port adjacent to the control connection port. The transfer byte size is 8-bit bytes. This byte size is relevant only for the actual transfer of the data; it has no bearing on representation of the data within a host's file system. The passive data transfer process (this may be a user-DTP or a second server-DTP) shall "listen" on the data port prior to sending a transfer request command. The FTP request command determines the direction of the data transfer. The server, upon receiving the transfer request, will initiate the data connection to the port. When the connection is established, the data transfer begins between DTP's, and the server-PI sends a confirming reply to the user-PI. Every FTP implementation must support the use of the default data ports, and only the USER-PI can initiate a change to non-default ports.

It is possible for the user to specify an alternate data port by use of the PORT command. The user may want a file dumped on a TAC line printer or retrieved from a third party host. In the latter case, the user-PI sets up control connections with both server-PI's. One server is then told (by an FTP command) to "listen" for a connection which the other will initiate. The user-PI sends one server-PI a PORT command indicating the data port of the other. Finally, both are sent the appropriate transfer commands. The exact sequence of commands and replies sent between the user-controller and the servers is defined in the Section on FTP Replies.

In general, it is the server's responsibility to maintain the data connection--to initiate it and to close it. The exception to this is when the user-DTP is sending the data in a transfer mode that requires the connection to be closed to indicate EOF. The server MUST close the data connection under the following conditions:

1. The server has completed sending data in a transfer mode that requires a close to indicate EOF.

2. The server receives an ABORT command from the user.
3. The port specification is changed by a command from the user.
4. The control connection is closed legally or otherwise.
5. An irrecoverable error condition occurs.

Otherwise the close is a server option, the exercise of which the server must indicate to the user-process by either a 250 or 226 reply only.

#### **2.4 FTP in Review:**

As well as web servers, mail servers and chat servers, the Internet houses FTP servers. These are servers that are like libraries of software or files that you can go to and "download". "Downloading" means copying a file to your computer. Conversely, "uploading" means to transmit a file from your computer to another computer on the Internet. Typically, when you visit an FTP server with a WYSIWYG FTP program (like CuteFTP) you will be presented with a list of directories. A directory is identified by the characters "dir", for "directory" or a little yellow folder icon. If you click on this line you will go down the directory tree to more directories or to the files in that directory or both.

Today most FTP sites can be viewed using a standard web browser. The web browser will list the directories and files in blue (denoting a link below). You can click on these links to start a download. Simply type the URL of the server into the location box. For example: ftp://name.of.site/ will give you a listing of all the directories of the FTP server; ftp://name.of.site/directory/ which will give you a listing of all the files available in that directory; and ftp://name.of.site/ directory/filename will download the actual file to your computer.

Many FTP servers are 'anonymous FTP' servers, which means you can log in with the username 'anonymous' and your e-mail address as the password. In the past you needed a special piece of software to view the files on an FTP server. You still need file transfer software if you want to "upload" files to an FTP or web server. This is required if you want to put pages on the web. There are a number of FTP programs around. The two most used packages are CuteFTP and WSFTP LE. These can be downloaded from many web sites. Inside an FTP program you will notice that the window is divided into two panes. The pane on the left shows directories and files on your computer. The window on the right is the public access area of the FTP server you are visiting.

#### **3.0 FTP Directories:**

##### **3.1 Moving Around FTP Directories:**

You can move through levels of directories in the same way you do in Windows Explorer in Windows. In other words, double mouse click on a directory to see its contents. Move up the directories by clicking on the up arrow and down directories by clicking on the directory's icon.

Let's say you had built a web site and you wanted to upload all your files to your ISP's web server. In the left pane you would go to the directory with your HTML files and picture files in it. In the right pane you would find your "work space" (supplied by your ISP, Usually 5MB). In the left pane you would highlight the files you want to move to the web server and then click the left pointing arrow. Note that you can delete and rename files in your FTP program.

### 3.2 Some Useful FTP Commands:

- **ascii**  
Changes to text mode. This is necessary for uploading a homepage file (index.html).
- **binary**  
Changes to binary mode. Necessary to upload a binary file, such as a picture.
- **dir**  
List files, with more info than "Ls"
- **put** (filename)  
Upload (filename) from your machine.
- **get** (filename)  
Download (filename) to your machine.
- **CD** (directoryname)  
Changes directory on the remote machine.
- **LCD** (directoryname)  
Changes directory on your local (home) computer.
- **mget** (files)  
Download multiple files to your machine.
- **mput** (files)  
Upload multiple files from your machine.
- **bye**  
Close the ftp session.
- **quit/close/bye/disconnect**: to disconnect from the FTP server.
- **help**  
Shows a list of available ftp commands.

### 4.0 Modes of Ftp:

There are two modes, basically for using FTP. In one mode, you sign on with your own account or ID and transfer files. In the other, you sign on as a guest or as an anonymous user and transfer files. With the anonymous FTP, your privileges on the host system will be severely limited for security reasons. That's because host that provides anonymous FTP do so specifically to make certain files available to anyone who wants them.

#### 4.1 Files modes:

There are two modes for transferring files in FTP.

1. ASCII mode is used for plain text, BinHex (Macintosh Transfer Format), uuencoded (Unix Transfer Encoding) and postscript files.
2. Binary mode (also, confusingly, called I mode on some servers) is for everything else.

#### **4.2 FTP and web browsers:**

Most recent web browsers and file managers can connect to FTP servers, although they may lack the support for protocol extensions such as FTPS. This allows manipulation of remote files over FTP through an interface similar to that used for local files. This is done via an FTP URL, which takes the form ftp(s)://<ftpserveraddress> (e.g., [1]). A password can optionally be given in the URL, e.g.: ftp(s)://<login>:<password>@<ftpserveraddress>:<port>. Most web-browsers require the use of passive mode FTP, which not all FTP servers are capable of handling.

#### **5.0 Generalized File Transfer:**

A method is disclosed for providing a file transfer interface for all types of transfer emulators/protocols. A set of common performed by all file transfer utilities is into a file transfer application layer. Another set protocol specific functions is grouped into an emulator. An efficient interface is used to interact between two layers. The external manifestation of this is the user sees file transfer as a common function of a lot of emulator/protocol specific utilities.

Interface for all types of transfer emulators/protocols. A set of common performed by all file transfer utilities is into a file transfer application layer. Another set protocol specific functions is grouped into an emulator An efficient interface is used to interact between two layers. The external manifestation of this is the user sees file transfer as a common function of a lot of emulator/protocol specific utilities. The one common file transfer application layer consists the following functions:

1. Menu/Operator interface
2. Message interface
3. File I/O interface
4. ASCII/EBCDIC translations

Multiple emulator layers can exist, and each consists of following functions:

1. File transfer protocol handling
2. Communications I/O
3. Emulator functions

The file transfer application layer provides a common for different file transfer protocols. When the transfer application layer connects to an emulator, it the protocol to be used during the connection. the file transfer application layer has connected, with the emulator proceeds in the same way for protocols, except that

different message numbers are and the number of units of data transferred are for each protocol. The emulator layer provides services to the file application layer for transporting information to from a host computer or another PC with the appropriate transfer protocol.

### **6.0 Batch & Interactive Transfer**

- **Batch transfer:**

- A user creates a list of files to be transferred through an interface program.
- The interface program drops the request into a queue.
- A transfer program reads the request and performs the transfer.
- Good for slow or unreliable transfers.

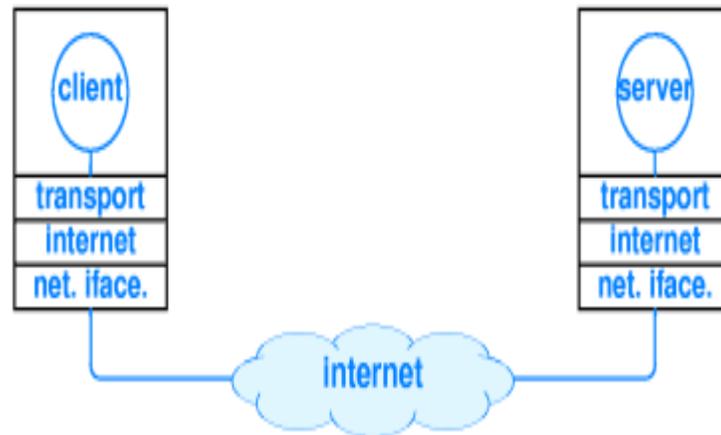
Batch & Interactive Transfer

- **Interactive transfer:**

- A user starts a transfer program.
- Actions include listing contents of directories and transferring files.
- The user can find and transfer files immediately.
- More convenient.
- Provides quick feedback in case of errors.

### **6.1 FTP Client-Server Interaction:**

- FTP uses the client-server model.
  - When a user starts an FTP session, the FTP client requests and the FTP server accepts a TCP control connection on port 21. The client side uses the control connection to send the user ID, password, and commands to the server. The control connection remains open during the entire session. When a user requests a file transfer, the FTP server opens a TCP data connection on port 20. FTP sends exactly one data file over a data connection.
- After a file is sent, a data connection is closed. If another file is to be sent, another data Connection is opened. The two functions of sending commands and data transfer are separated.



**Figure : 4 A client and server architecture**

FTP is a powerful protocol that allows files to be transferred from "computer A" to "computer B" or vice versa. FTP works on the principal of a client/server. A client program enables the user to interact with a server in order to access information and services on the server computer. Files that can be transferred are stored on computers called FTP servers.

The File Transfer Protocol is used to send files from one system to another under user commands. Both text and binary files are accommodated and the protocol provides features for controlling user access. When a user wishes to engage in File transfer, FTP sets up a TCP connection to the target system for the exchange of control messages. These allow user ID and password to be transmitted and allow the user to specify the file and file action desired. Once file transfer is approved, a second TCP connection is set up for data transfer. The file is transferred over the data connection, without the overhead of headers, or control information at the application level. When the transfer is complete, the control connection is used to signal the completion and to accept new file transfer commands.

## **6.2 Characteristics of Clients and Servers**

**In general, client software has the following characteristics.**

- It is an application program that becomes a client temporarily when remote access is needed, but performs other computation locally.
- It is invoked by a user and executes for one session.
- It runs locally on the user's computer.
- It actively initiates contact with a server (CONNECT primitive).
- It can access multiple services as needed.

**In general, server software has the following characteristics.**

- It is a special-purpose program dedicated to providing one service.

- It is invoked automatically when a system boots, and continues to execute through many sessions.
- It runs on a shared computer.
- It waits passively for contact from arbitrary remote clients (LISTEN primitive).
- It accepts contact from arbitrary clients, but offers a single service.

### 7.0 Summary:

In this lesson we have studied fundamental concepts used in file transfer protocol that binds whole world in single rope. We have also discussed the concept of client server with some important commands.

### 8.0 Keywords

TCP/IP protocol, Anonymous FTP, Interactive, FTP Directories, Data Connection, FTP commands

### 9.0 Self Check Exercise

- Q1. What do you mean by client - server architecture?
- Q2. What is generalized file transfer?
- Q3. Write down the importance of file transfer protocol.
- Q4. When we need file transfer protocol?

### 10.0 Suggested Readings:

- “Using Ftp (User Friendly Reference)” by [Mary Ann Pike](#) and [Noel Estabrook](#).  
“FTP For The Rest of Us” by [Eugene Khmelevsky](#).

**HTML - I**

- 1. Objectives of the lesson**
- 2. Introduction**
- 3. About HTML**
- 4. Getting Started**
  - 4.1. Creating a webpage**
- 5. HTML Elements**
  - 5.1. Closing your tags**
  - 5.2. UPPERCASE or lowercase**
- 6. HTML Formatting**
  - 6.1. Headings**
  - 6.2. Bold**
  - 6.3. Italics**
  - 6.4. Line Breaks**
  - 6.5. Horizontal Rule**
  - 6.6. Unordered (un-numbered) List**
  - 6.7. Ordered (numbered) List**
- 7. HTML Attributes**
  - 7.1. Width**
  - 7.2. Alignment**
- 8. HTML Colors**
  - 8.1. Foreground Color**
  - 8.2. Background Color**
  - 8.3. Border Color**
- 9. HTML Links**
  - 9.1. Link Targets**
  - 9.2. Email Links**
- 10. Summary**
- 11. Exercise**
- 12. Suggested Reading**

**1. Objective of the lesson**

- Design the first webpage
- Formatting of the webpage
- Links between different webpage

**2. Introduction**

**H-T-M-L** are initials that stand for **HyperText Markup Language** (computer people love initials and acronyms -- you'll be talking acronyms ASAP). Let me break it down for you:

- **Hyper** is the opposite of linear. It used to be that computer programs had to move in a linear fashion. This before this, this before this, and so on. HTML does not hold to that pattern and allows the person viewing the World Wide Web page to go anywhere, any time they want.
- **Text** is what you will use. Real, honest to goodness English letters.
- **Mark up** is what you will do. You will write in plain English and then mark up what you wrote. More to come on that in the next Primer.
- **Language** because they needed something that started with "L" to finish HTML and Hypertext Markup *Louie* didn't flow correctly. Because it's a language, really -- but the language is plain English.

**3. About HTML**

HTML, which stands for HyperText Markup Language, is a markup language used to create web pages. The web developer uses "HTML tags" to format different parts of the document. For example, you use HTML tags to specify headings, paragraphs, lists, tables, images and much more.

HTML is a subset of Standard Generalized Markup Language (SGML) and is specified by the World Wide Web Consortium (W3C).

**3.1 What do I need to create HTML?**

You don't need any special equipment or software to create HTML. In fact, you probably already have everything you need. Here is what you need:

- Computer
- Text editor. For example, Notepad (for Windows), Pico (for Linux), or Simple text (Mac). You could use a HTML editor if you like but it's not needed.
- Web Browser. For example, Internet Explorer or Firefox.

**3.2 Do I need to be online?**

No, you do not need to be online to create web pages. You can create web pages on your local machine. You only need to go online when you want to publish your web page to the web - this bit comes later.

## 4. Getting Started

OK, lets get straight into it. Here, you will learn just how easy it is to create a web page. In fact, by the time you've finished with *this* web page, you will have created *your own* web page!

When you create a web page you will usually do something like this:

1. Create an HTML file
2. Type some HTML code
3. View the result in your browser
4. Repeat the last 2 steps (if necessary)

### 4.1 Creating a Webpage

OK, let's walk through the above steps in more detail.

#### 1. Create an HTML file

An HTML file is simply a text file saved with an .html or .htm extension (i.e. as opposed to a .txt extension).

1. Open up your computer's normal plain text editor (this will probably be "Notepad" if you're using Windows or "SimpleText" if you're using Macs). You could use a specialized HTML editor such as DreamWeaver or FrontPage if you prefer.
2. Create a new file (if one wasn't already created)
3. Save the file as html\_tutorial\_example.html

#### 2. Type some HTML code

Type the following code:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>HTML Tutorial Example</title>
</head>
<body>
<p>Less than 5 minutes into this HTML tutorial and
I've already created my first homepage!</p>
</body>
</html>
```

#### 3. View the result in your browser

Either...

1. Navigate to your file then double click on it

...OR...

2. Open up your computer's web browser (for example, Internet Explorer, Firefox, Netscape etc).

3. Select File > Open, then click "Browse". A dialogue box will appear prompting you to navigate to the file. Navigate to the file, then select "Open".

#### 4. Repeat the last 2 steps until you're satisfied with the result

It's unrealistic to expect that you will always get it right the first time around. Don't worry - that's OK! Just try again and again - until you get it right.

#### Explanation of code

OK, before we get too carried away, I'll explain what that code was all about.

We just coded a bunch of *HTML tags*. These tags tell the browser what to display and where. You may have noticed that for every "opening" tag there was also a "closing" tag, and that the content we wanted to display appeared in between. Most HTML tags have an opening and closing tag.

All HTML documents should at least contain all of the tags we've just coded and in that order.

The next lesson goes into a bit more detail about HTML tags.

#### 5. HTML Elements

*HTML elements* are the fundamentals of HTML. HTML documents are simply a text file made up of HTML elements. These elements are defined using *HTML tags*. HTML tags tell your browser which *elements* to present and how to present them. Where the element appears is determined by the order in which the tags appear.

HTML consists of almost 100 tags. Don't let that put you off though - you will probably find that most of the time, you only use a handful of tags on your web pages. Having said that, I highly recommend learning all HTML tags eventually - but we'll get to that later.

OK, lets look more closely at the example that we created in the previous lesson.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>HTML Tutorial Example</title>
</head>
<body>
<p>Less than 5 minutes into this HTML tutorial and
I've already created my first homepage!</p>
</body>
</html>
```

Explanation of the above code:

- The `<!DOCTYPE... >` element tells the browser which version of HTML the document is using.
- The `<html>` element can be thought of as a container that all other tags sit inside (except for the !DOCTYPE tag).
- The `<head>` tag contains information that is not normally viewable within your browser (such as meta tags, JavaScript and CSS), although the `<title>` tag is an exception to this. The content of the `<title>` tag is displayed in the browser's title bar (right at the very top of the browser).
- The `<body>` tag is the main area for your content. This is where most of your code (and viewable elements) will go.
- The `<p>` tag declares a paragraph. This contains the body text.

### 5.1 Closing your tags

As mentioned in a previous lesson, you'll notice that all of these tags have opening and closing tags, and that the content of the tag is placed in between them. There are a few exceptions to this rule.

You'll also notice that the closing tag is slightly different to the opening tag - the closing tag contains a forward slash (/) after the <. This tells the browser that this tag closes the previous one.

### 5.2 UPPERCASE or lowercase?

Although most browsers will display your page regardless of the case you use, you should always code in lowercase. This helps keep your code XML compliant (but that's another topic).

Therefore... **Good:** `<head>`

**Bad:** `<HEAD>`

In the next lesson, we learn about some of the more common formatting tags.

## 6. HTML Formatting

You may be familiar with some of the formatting options that are available in word processing applications such as Microsoft Office, and desktop publishing software such as QuarkXpress. Well, many of these formatting features are available in HTML too! This lesson contains some of the more common formatting options.

### 6.1 Headings

There is a special tag for specifying headings in HTML. There are 6 levels of headings in HTML ranging from h1 for the most important, to h6 for the least important.

Typing this code:

```
<h1>Heading 1</h1>
```

<h2>Heading 2</h2>

<h3>Heading 3</h3>

<h4>Heading 4</h4>

<h5>Heading 5</h5>

<h6>Heading 6</h6>

Results in this:

**Heading 1**

**Heading 2**

**Heading 3**

**Heading 4**

Heading 5

*Heading 6*

## 6.2 Bold

You specify bold text with the <b> tag.

Typing this code:

<b>This text is bold.</b>

Results in this:

**This text is bold.**

## 6.3 Italics

You specify italic text with the <i> tag.

Typing this code:

<i>This text is italicised.</i>

Results in this:

*This text is italicised.*

## 6.4 Line Breaks

Typing this code:

<p>Here is a...<br />line break.</p>

Results in this:

Here is a  
line break.

## 6.5 Horizontal Rule

Typing this code:

Here's a horizontal rule... <hr /> ...that was a horizontal rule :)

Results in this:

Here's a horizontal rule...

---

...that was a horizontal rule :)

## 6.6 Unordered (un-numbered) List

Typing this code:

```
<ul>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ul>
```

Results in this:

- List item 1
- List item 2
- List item 3

## 6.7 Ordered (numbered) List

Note, that the only difference between an ordered list and an unordered list is the first letter of the list definition ("o" for ordered, "u" for unordered).

Typing this code:

```
<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
```

Results in this:

1. List item 1
2. List item 2
3. List item 3

We will be covering more HTML tags throughout this tutorial, but before we do that, you should know about *attributes*.

## 7. HTML Attributes

HTML tags can contain one or more *attributes*. Attributes are added to a tag to provide the browser with more information about how the tag should appear or behave. Attributes consist of a *name* and a *value* separated by an equals (=) sign.

Consider this example:

```
<body style="background-color:orange">
```

OK, we've already seen the body tag in previous lessons, but this time we can see that something extra has been added to the tag - an attribute. This particular attribute statement, `style="background-color:orange"`, tells the browser to *style* the body element with a *background color* of *orange*.

### 7.1 Width

Here's another example of adding an attribute to an HTML tag:

```
<hr width="60%" />
```

This results in:

## 7.2 Alignment

You can align HTML elements to the left, right, or center:

```
<h3 align="left">Aligned left</h3>
```

```
<h3 align="center">Aligned center</h3>
```

```
<h3 align="right">Aligned right</h3>
```

This results in:

**Aligned left**

**Aligned center**

**Aligned right**

Many attributes are available to HTML elements, some are common across most tags, others can only be used on certain tags. Some of the more common attributes are:

Attribute	Description	Possible Values
width	Specifies the width of an element	(Numeric value)
height	Specifies the height of an element	(Numeric value)
align	Specifies the horizontal alignment of an element	left, center, or right
class	Used with Cascading Style Sheets (CSS)	(the name of a class)
style	Used with Cascading Style Sheets (CSS)	(style information)
title	Can be used to display a "tooltip" for your elements.	(You supply the text)

You don't need to fully comprehend these just yet. The good thing about attributes is that, generally, they are optional. Many HTML elements assign a *default* value to its attributes - meaning that, if you don't include that attribute, a value will be assigned anyway. For example, if you don't specify the alignment for the `p` element, it will always be aligned left. This is handy - you will be using that tag a lot, so the less coding you have to do, the better!

You will see more attributes being used as we cover off some of the more advanced HTML elements.

## 8. HTML Colors

In HTML, colors can be added by using the `style` attribute. You can specify a color name (eg, *blue*), a hexadecimal value (eg, *#0000ff*), or an RGB value (eg *rgb(0,0,255)*).

### Syntax

#### 8.1 Foreground Color

To add color to an HTML element, you use `style="color:{color}"`, where `{color}` is the color value. For example:

```
<h3 style="color:blue">HTML Colors</h3>
```

This results in:

HTML Colors

## 8.2 Background Color

To add a background color to an HTML element, you use `style="background-color:{color}"`, where `{color}` is the color value. For example:

```
<h3 style="background-color:blue">HTML Colors</h3>
```

This results in:

**HTML Colors**

## 8.3 Border Color

To add a colored border to an HTML element, you use `style="border:{width} {color} {style}"`, where `{width}` is the width of the border, `{color}` is the color of the border, and `{style}` is the style of the border. For example:

```
<h3 style="border:1px blue solid;">HTML Colors</h3>
```

This results in:

**HTML Colors**

### Color Names

The most common methods for specifying colors are by using the color name or the hexadecimal value. Although color names are easier to remember, the hexadecimal values and RGB values provides you with more color options.

Hexadecimal values are a combination of letters and numbers. The numbers go from 0 - 9 and the letters go from A to F. When using hexadecimal color values in your HTML/CSS, you precede the value with a hash (#). Although hexadecimal values may look a little weird at first, you'll soon get used to them.

There are 16 color names (as specified in the HTML 4.0 specification). The chart below shows these color names and their corresponding hexadecimal value.

Color	Color Name	Hexadecimal Value	Color	Color Name	Hexadecimal Value
	Black	#000000		Green	#008000
	Silver	#c0c0c0		Lime	#00ff00
	Gray	#808080		Olive	#808000
	White	#ffffff		Yellow	#ffff00
	Maroon	#800000		Navy	#000080

	Red	#ff0000		Blue	#0000ff
	Purple	#800080		Teal	#008080
	Fuchsia	#ff00ff		Aqua	#00ffff

You can make up your own colors by simply entering any six digit hexadecimal value (preceeded by a hash). In the following example, we're using the same code as above. The only difference is that, instead of using "blue" as the value, we're using its hexadecimal equivalent (which is #0000ff):

```
<h3 style="color:#0000ff">HTML Colors</h3>
```

This results in:

**HTML Colors**

If we wanted to change to a deeper blue, we could change our hexadecimal value slightly, like this:

```
<h3 style="color:#000069">HTML Colors</h3>
```

This results in:

**HTML Colors**

### Choosing Colors - The Easy Way

By using hexadecimal or RGB color values, you have a choice of over 16 million different colors. You can start with 000000 and increment by one value all the way up to FFFFFFFF. Each different value represents a slightly different color.

But don't worry - you won't need to remember every single hexadecimal color! The [HTML color picker and color chart](#) make it easy for you to choose colors for your website.

## 9. HTML Links

*Links*, otherwise known as *hyperlinks*, are defined using the `<a>` tag - otherwise known as the *anchor* element.

To create a hyperlink, you use the `a` tag in conjunction with the `href` attribute (`href` stands for Hypertext Reference). The value of the `href` attribute is the *URL*, or, location of where the link is pointing to.

Example HTML Code:

```
Visit the <a href="http://www.natural-environment.com/blog/">Natural Environment Blog</a>
```

This results in:

Visit the [Natural Environment Blog](#)

Hypertext references can use *absolute* URLs, *relative* URLs, or *root relative* URLs.

### **absolute**

This refers to a URL where the full path is provided. For example, `http://www.quackit.com/html/tutorial/index.cfm`

**relative**

This refers to a URL where only the path, *relative to the current location*, is provided. For example, if we want to reference the `http://www.quackit.com/html/tutorial/index.cfm` URL, and our current location is `http://www.quackit.com/html`, we would use `tutorial/index.cfm`

**root relative**

This refers to a URL where only the path, *relative to the domain's root*, is provided. For example, if we want to reference the `http://www.quackit.com/html/tutorial/index.cfm` URL, and the current location is `http://www.quackit.com/html`, we would use `/html/tutorial/index.cfm`. The forward slash indicates the domain's root. This way, no matter where your file is located, you can always use this method to determine the path, even if you don't know what the domain name will eventually be.

**9.1 Link Targets**

You can nominate whether to open the URL in a new window or the current window. You do this with the target attribute. For example, `target="_blank"` opens the URL in a new window.

The target attribute can have the following possible values:

<code>_blank</code>	Opens the URL in a <i>new</i> browser window.
<code>_self</code>	Loads the URL in the <i>current</i> browser window.
<code>_parent</code>	Loads the URL into the <i>parent</i> frame (still within the current browser window). This is only applicable when using frames.
<code>_top</code>	Loads the URL in the <i>current</i> browser window, but cancelling out any frames. Therefore, if frames were being used, they aren't any longer.

Example HTML Code:

Visit the `<a href="http://www.natural-environment.com" target="_blank">Natural Environment</a>`

This results in:

Visit the [Natural Environment](http://www.natural-environment.com)

**Named Anchors**

You can make your links "jump" to other sections within the same page. You do this with *named anchors*.

To use named anchors, you need to create two pieces of code - one for the hyperlink (this is what the user will click on), and one for the named anchor (this is where they will end up).

This page uses a named anchor. I did this by performing the steps below:

1. I created the named anchor first (where the user will end up)

Example HTML Code:

```
<h2>Link Targets<a name="link_targets"></a></h2>
```

2. I then created the hyperlink (what the user will click on). This is done by linking to the *name* of the named anchor. You need to precede the name with a hash (#) symbol.

Example HTML Code:

```
<a href="#link_targets">Link Targets</a>
```

This results in:

Link Targets

When you click on the above link, this page should jump up to the "Link Targets" section (above). You can either use your back button, or scroll down the page to get back here.

You're back? Good, now lets move on to email links.

## 9.2 Email Links

You can create a hyperlink to an email address. To do this, use the `mailto` attribute in your anchor tag.

Example HTML Code:

```
<a href="mailto:king_kong@quackit.com">Email King Kong</a>
```

This results in:

Email King Kong

Clicking on this link should result in your default email client opening up with the email address already filled out.

You can go a step further than this. You can auto-complete the subject line for your users, and even the body of the email. You do this appending subject and body parameters to the email address.

Example HTML Code:

```
<a href="mailto:king_kong@quackit.com?subject=Question&body=Hey there">Email King Kong</a>
```

This results in:

Email King Kong

## Base href

You can specify a default URL for all links on the page to start with. You do this by placing the `base` tag (in conjunction with the `href` attribute) in the document's *head*.

Example HTML Code:

```
<head>  
<base url="http://www.quackit.com">  
</head>
```

## 10 Summary

- We started with an introduction to HTML, and what it's used for.

- We then created a basic webpage, before moving on to some of the basic tags. We learned that each tag can have different attributes, and that we can supply the value for these attributes.
- We learned that we can create hyperlinks using the anchor tag (a) with the href attribute.

**11 Self Check Exercise**

1. What are the Attributes of HTML?
2. What are the links?
3. Define Colors?

**12 Suggested Reading**

1. The Complete Reference-HTML by Powell Thomas, Tata McGraw Hill
2. Khurana R., "HTML" APH Publishing
3. HTML-E. Stephen Mack and Janam Platt, BPB Publications

**HTML – II**

- 1. Objectives of the lesson**
- 2. Introduction**
- 3. HTML Images**
  - 3.1. Image Alignment**
  - 3.2. Image Links**
  - 3.3. Removing the Border**
- 4. HTML Meta Tags**
- 5. HTML Comments**
- 6. HTML Forms**
  - 6.1. The Input Tag**
- 7. HTML Tables**
- 8. HTML Image Maps**
- 9. Summary**
- 10. Self Check Exercise**
- 11. Suggested Reading**

- 1. Objective of the lesson**
  - To attachment the images
  - To create Tables with different attribute
  - To create Form with different input tags

**2. Introduction**

To display an image on a page, you need to use the src attribute. Src stands for "source". The value of the src attribute is the URL of the image you want to display on your page. Meta elements are typically used to specify page description, keywords, author of the document, last modified, and other metadata. The <meta> tag always goes inside the head element.

The <input> tag is used to select user information. An input field can vary in many ways, depending on the type attribute. An input field can be a text field, a checkbox, a password field, a radio button, a button, and more.

### 3. HTML Images

Images make up a large part of the web - most websites contain images. HTML makes it very easy for you to embed images into your web page.

To embed an image into a web page, the image first needs to exist in either .jpg, .gif, or .png format. You can create images in an image editor (such as Adobe Photoshop) and save them in the correct format.

Once you've created an image, you need to embed it into your web page. To embed the image into your web page, use the `<img />` tag, specifying the actual location of the image.

#### Example of Image Usage

HTML Code:

```

```

This results in:



The img above contains a number of attributes. These attributes tell the browser all about the image and how to display it. Here's an explanation of these attributes:

src	Required attribute. This is the path to the image. It can be either an <i>absolute</i> path, or a <i>relative</i> path (remember these terms from our last lesson?)
width	Optional attribute (but recommended). This specifies the width to display the image. If the actual image is wider, it will shrink to the dimensions you specify here. Likewise, if the actual image is smaller it will expand to your dimensions. I don't recommend specifying a different size for the image, as it will lose quality. It's better to make sure the image is the correct size to start with.
height	Optional attribute (but recommended). This specifies the height to display the image. This attribute works similar to the width.
alt	Alternate text. This specifies text to be used in case the browser/user agent can't render the image.

#### 3.1 Image Alignment

You can determine how your images will be aligned, relative to the other content on the page (such as a paragraph of text). You do this using the align attribute.

HTML Code:

```
<p>
```

```

```

Here is a paragraph of text to demonstrate HTML images and how they can be aligned to the right of a paragraph (or paragraphs) if you so desire.</p> </p>This can be used to produce some nice layout effects, especially if you have a lot of text, and it runs right past the image.</p><p> Otherwise, the image will just hang below the text and may look funny.</p>

This results in:

Here is a paragraph of text to demonstrate HTML images and how they can be aligned to the right of a paragraph (or paragraphs) if you so desire.

This can be used to produce some nice layout effects, especially if you have a lot of text, and it runs right past the image.

Otherwise, the image will just hang below the text and may look funny.



### 3.2 Image Links

You can make your images "clickable" so that when a user clicks the image, it opens another URL. You do this by simply wrapping the image with hyperlink code.

HTML Code:

```
<a href="http://www.quackit.com/html/tutorial">

</a>
```

This results in:



### 3.3 Removing the Border

You might notice that this has created a border around the image. This is default behaviour for most browsers. If you don't want the border, specify border="0".

HTML Code:

```
<a href="http://www.quackit.com/html/tutorial">

</a>
```

This results in:



#### 4. HTML Meta Tags

Meta tags allow you to provide *metadata* about your HTML pages. This can be very useful for search engines and can assist the "findability" of the information on your website.

##### What is Metadata?

Metadata is information about, or that describes, other data or information.

If we relate this to a web page, if you think about it for a moment, you could probably come up with a lot more information about a web page than what you're actually displaying to the reader. For example, there could be a number of *keywords* that are related to the page. You could probably give the page a *description*. The page also has an *author* - right? All these could be examples of metadata.

##### Metadata on the Web

Metadata is a very important part of the web. It can assist search engines in finding the best match when a user performs a search. Search engines will often look at any metadata attached to a page - especially keywords - and rank it higher than another page with less relevant metadata, or with no metadata at all.

##### Adding Meta Tags to Your Documents

You can add metadata to your web pages by placing <meta> tags between the <head> and </head> tags. They can include the following attributes:

Attribute	Description
Name	Name for the property. Can be anything. Examples include, keywords, description, author, revised, generator etc.
content	Specifies the property's value.
scheme	Specifies a scheme to use to interpret the property's value (as declared in the content attribute).
http-equiv	Used for http response message headers. For example http-equiv can be used to refresh the page or to set a cookie. Values include content-type, expires, refresh and set-cookie.

##### Example HTML Code:

Keywords:

```
<meta name="keywords" content="HTML, meta tags, metadata" />
```

Description:

```
<meta name="description" content="Contains info about meta tags" />
```

Revision date (last time the document was updated):

```
<meta name="revised" content="Quackit, 6/12/2002" />
```

Refresh the page every 10 seconds:

```
<meta http-equiv="refresh" content="10" />
```

The above examples are some of the more common uses for the meta tag.

## 5. HTML Comments

Before we go any further, I'd like to introduce the concept of *comments*. Although comments are optional, they can assist you greatly.

Comments are a part of the HTML code and is used to explain the code. This can be helpful for other HTML coders when trying to interpret someone else's code. It can also be useful for yourself if you have to revisit your code in many months, or even years time. Comments aren't displayed in the browser - they are simply there for the programmer's benefit.

You write comments like this:

```
<!-- Write your comment here -->
```

Comments always start with `<!--` and end with `-->`. This tells the browser when a comment begins and ends.

Example HTML Code:

```
<!-- Display a happy image --->
```

```

```

This results in:



As you can see, the comment is invisible to the user viewing the page in the browser. It is there, simply for the HTML coder's benefit.

Well, we have now finished the first part of this tutorial. We have covered a lot, and by now, you have learned enough to build a website.

If you'd like to learn some of the more advanced HTML techniques, such as, how to create tables, frames, forms etc, please continue!

## 6. HTML Forms

HTML enables us to create forms. This is where our websites can become more than just a nice advertising brochure. Forms allow us to build more dynamic websites that allow our users to interact with it.

An HTML form is made up of any number of *form elements*. These elements enable the user to do things such as enter information or make a selection from a preset options.

In HTML, a form is defined using the `<form></form>` tags. The actual form elements are defined between these two tags.

### 6.1 The Input Tag

This is the most commonly used tag within HTML forms. It allows you to specify various types of user input fields such as text, radio buttons, checkboxes etc.

#### Text

Text fields are used for when you want the user to type text or numbers into the form.

```
<input type="text" />
```

This results in:

#### Radio Buttons

Radio buttons are used for when you want the user to select *one* option from a pre-determined set of options.

```
<input type="radio" name="lunch" value="pasta" /><br /> <input type="radio" name="lunch" value="rissotto" />
```

This results in:

#### Checkboxes

Checkboxes are similar to radio buttons, but enable the user to make multiple selections..

```
<input type="checkbox" name="lunch" value="pasta" /><br /> <input type="checkbox" name="lunch" value="rissotto" />
```

This results in:

#### Submit

The submit button allows the user to actually submit the form.

```
<input type="submit" />
```

This results in:

### Select Lists

A select list is a dropdown list with options. This allows the user to select one option from a list of pre-defined options.

The select list is created using the select in conjunction with the option tag.

```
<select>
  <option value ="sydney">Sydney</option>
  <option value ="melbourne">Melbourne</option>
  <option value ="cromwell">Cromwell</option>
  <option value ="queenstown">Queenstown</option>
</select>
```

This results in:



### Form Action

Usually when a user submits the form, you need the system to do something with the data. This is where the *action* page comes in. The action page is the page that the form is submitted to. This page could contain advanced scripts or programming that inserts the form data into a database or emails an administrator etc.

Creating an action page is outside the scope of this tutorial. In any case, many web hosts provide scripts that can be used for action page functionality, such as emailing the webmaster whenever the form has been completed. For now, we will simply look at how to submit the form to the action page.

You nominate an action page with the action attribute.

Example HTML Code:

```
<form action="/html/tags/html_form_tag_action.cfm" method="get">
```

First name:

```
<input type="text" name="first_name" value="" maxlength="100" />
```

```
<br />
```

Last name:

```
<input type="text" name="last_name" value="" maxlength="100" />
```

```
<input type="submit" value="Submit" />
```

```
</form>
```

This results in:

First name:

Last name:

Oh, one last thing. You may have noticed the above example uses a method attribute. This attribute specifies the HTTP method to use when the form is submitted.

Possible values are:

- get (the form data is appended to the URL when submitted)
- post (the form data is not appended to the URL)

Providing this attribute is optional. If you don't provide it, the method will be *post*.

## 7. HTML Tables

In HTML, the original purpose of tables was to present tabular data. Although they are still used for this purpose today, many web designers tended to use tables for advanced layouts. This is probably due to the restrictions that HTML has on layout capabilities - it wasn't really designed as a layout language.

Anyway, whether you use tables for presenting tabular data, or for page layouts, you will use the same HTML tags.

### Basic table tags

In HTML, you create tables using the table tag, in conjunction with the tr and td tags. Although there are other tags for creating tables, these are the basics for creating a table in HTML.

HTML Code:

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

This results in:

Table cell 1	Table cell 2
--------------	--------------

You'll notice that we added a border attribute to the table tag. This particular attribute allows us to specify how thick the border will be. If we don't want a border we can specify 0 (zero). Other common attributes you can use with your table tag include width, width, cellspacing and cellpadding.

You can also add attributes to the tr and td tags. For example, you can specify the width of each table cell.

Widths can be specified in either pixels or percentages. Specifying the width in pixels allows you to specify an exact width. Percentages allows the table to "grow" or "shrink" depending on what else is on the page and how wide the browser is.

HTML Code:

```
<table border="1" cellpadding="5" cellspacing="5" width="100%">
<tr>
<td width="20%">Table cell 1</td><td>Table cell 2</td>
</tr>
</table>
```

This results in:

Table cell 1	Table cell 2
--------------	--------------

### Table Headers

Many tables, particularly those with tabular data, have a header row or column. In HTML, you can use the `th` tag.

Most browsers display table headers in bold and center-aligned.

HTML Code:

```
<table border="1" cellpadding="5" cellspacing="5" width="100%">
<tr>
<th>Table header</th>
<th>Table header</th>
</tr>
<tr>
<td width="20%">Table cell 1</td><td>Table cell 2</td>
</tr>
</table>
```

This results in:

<b>Table header</b>	<b>Table header</b>
Table cell 1	Table cell 2

### Colspan

You can use the `colspan` attribute to make a cell span multiple columns.

HTML Code:

```
<table border="1" cellpadding="5" cellspacing="5" width="100%">
<tr>
<th colspan="2">Table header</th>
</tr>
<tr>
<td width="20%">Table cell 1</td><td>Table cell 2</td>
</tr>
</table>
```

This results in:

<b>Table header</b>	
Table cell 1	Table cell 2

**Rowspan**

Rowspan is for rows just what colspan is for columns (rowspan allows a cell to span multiple rows).

HTML Code:

```
<table border="1" cellpadding="5" cellspacing="5" width="100%">
<tr>
<th rowspan="2">Table header</th><td>Table cell 1
</tr>
<tr>
<td>Table cell 2</td>
</tr>
</table>
```

This results in:

<b>Table header</b>	Table cell 1
	Table cell 2

**Color**

You can apply color to your table using CSS. Actually, you can apply any applicable CSS property to your table - not just colors. For example, you can use CSS to apply width, padding, margin, etc

HTML Code:

```
<table border="1" cellpadding="5" cellspacing="5" width="100%">
<tr>
<th style="color:blue;background-color:yellow;" rowspan="2">Table
header</th><td>Table cell 1
</tr>
<tr>
<td>Table cell 2</td>
</tr>
</table>
```

This results in:

<b>Table header</b>	Table cell 1
	Table cell 2

## 8. HTML Image Maps

Image maps are images with clickable areas (sometimes referred to as "hotspots") that usually link to another page. If used tastefully, image maps can be really effective. If not, they can potentially confuse users.

To create an image map:

1. First, you need an image. Create an image using the usual method (i.e. via an image editor, then save as a gif or jpeg into your website's image folder).
2. Use the HTML map tag to create a map with a name. Inside this tag, you will specify where the clickable areas are with the HTML area tag
3. Use the HTML img tag to link to this image. In the img tag, use with the usemap attribute to define which image map to use (the one we just specified).

### Image Map Example

HTML Code:

```
<img src ="/pix/mueller_hut/mueller_hut_t.jpg"
width="225" height="151" border="0"
alt="Mueller Hut, Mount Cook, and I"
usemap ="#muellermap" />
<map id ="muellermap"
name="muellermap">
  <area shape ="rect" coords ="90,80,120,151"
href ="javascript:alert('Me');"
alt="Me" />
  <area shape ="poly" coords ="55,55,120,80,90,80,90,100,70,100,20,80,55,55"
href ="http://en.wikipedia.org/wiki/Mount_Cook"
alt="Mount Cook" />
  <area shape ="poly" coords ="145,80,145,100,215,90,215,80,180,60,145,80"
href ="http://www.natural-environment.com/places/mueller_hut.php"
alt="Mueller Hut" />
</map>
```

This results in:



OK, compared to our previous lessons, this code is looking quite complex. However, once you really study it, it is not that complex. All we are doing, is specifying an image, then we are creating a map with coordinates. The hardest part is getting all the coordinates right.

In our example, we use the area in conjunction with the shape and coord attributes. These accept the following attributes:

shape	<p>Defines a shape for the clickable area. Possible values:</p> <ul style="list-style-type: none"> <li>• default</li> <li>• rect</li> <li>• circle</li> <li>• poly</li> </ul>
coords	<p>Specifies the coordinates of the clickable area. Coordinates are specified as follows:</p> <ul style="list-style-type: none"> <li>• rect: left, top, right, bottom</li> <li>• circle: center-x, center-y, radius</li> <li>• poly: x1, y1, x2, y2, ...</li> </ul>

You can use the above attributes to configure your own image map with as many shapes and clickable regions as you like.

### 9. Summary

- We embedded an image using the img tag with the src attribute, and applied several other attributes to the tag
- We then learned about the importance of commenting our code to make site maintenance easier
- Armed with the above knowledge, we tackled some more advanced HTML such as image maps, tables, forms, and frames.

### 10. Self Check Exercise

1. How to put the Comment in HTML?
2. What are the Forms?
3. What are the Components of Forms?

### 11. Suggested Reading

1. The Complete Reference-HTML by Powell Thomas, Tata McGraw Hill
2. Khurana R., "HTML" APH Publishing
3. HTML-E. Stephen Mack and Janam Platt, BPB Publications

**ADVANCE HTML**

- 1. Objectives of the lesson**
- 2. Introduction**
- 3. HTML Frames**
- 4. HTML Entities**
- 5. HTML Layouts**
  - 5.1. HTML layout with tables**
  - 5.2. HTML layout with DIV, SPAN and CSS**
  - 5.3. Advanced Layouts**
- 6. HTML Styles**
- 7. HTML Scripts**
  - 7.1. Adding a Script**
  - 7.2. Triggering a Script**
  - 7.3. Calling an External Script**
  - 7.4. Hide Scripts from Older Browsers**
  - 7.5. Alternate Information for Older Browsers**
  - 7.6. Set a Default Scripting Language**
- 8. HTML Website Templates**
- 9. Website Hosting**
- 10. Summary**
- 11. Self Check Exercise**
- 12. Suggested Reading**

**1. Objective of the lesson**

To create advance webpage, using scripting language and with the help of templates.

**2. Introduction**

HTML frames allow to present documents in multiple views, which may be independent windows or subwindows. Multiple views offer designers a way to keep

certain information visible, while other views are scrolled or replaced. For example, within the same window, one frame might display a static banner, a second a navigation menu, and a third the main document that can be scrolled through or replaced by navigating in the second frame.

With HTML scripts you can create dynamic web pages, make image rollovers for really cool menu effects, or even validate your HTML form's data before you let the user submit. However, javascript and vbscript are very complicated compared to HTML. It may be simpler just to download someone else's scripting code and use it on your web page

### 3. HTML Frames

In HTML, frames enable you present multiple HTML documents within the same window. For example, you can have a left frame for navigation and a right frame for the main content.

Frames are achieved by creating a *frameset* page, and defining each frame from within that page. This frameset page doesn't actually contain any content - just a reference to each frame. The HTML [frame](#) tag is used to specify each frame within the frameset. All frame tags are nested with a [frameset](#) tag.

So, in other words, if you want to create a web page with 2 frames, you would need to create 3 files - 1 file for each frame, and 1 file to specify how they fit together.

#### Creating Frames

##### Two Column Frameset

HTML Code:

The frameset (frame\_example\_frameset\_1.html):

```
<html>
<head>
<title>Frameset page</title>
</head>
<frameset cols = "25%, *">
  <frame src ="frame_example_left.html" />
  <frame src ="frame_example_right.html" />
</frameset>
</html>
```

The left frame (frame\_example\_left.html):

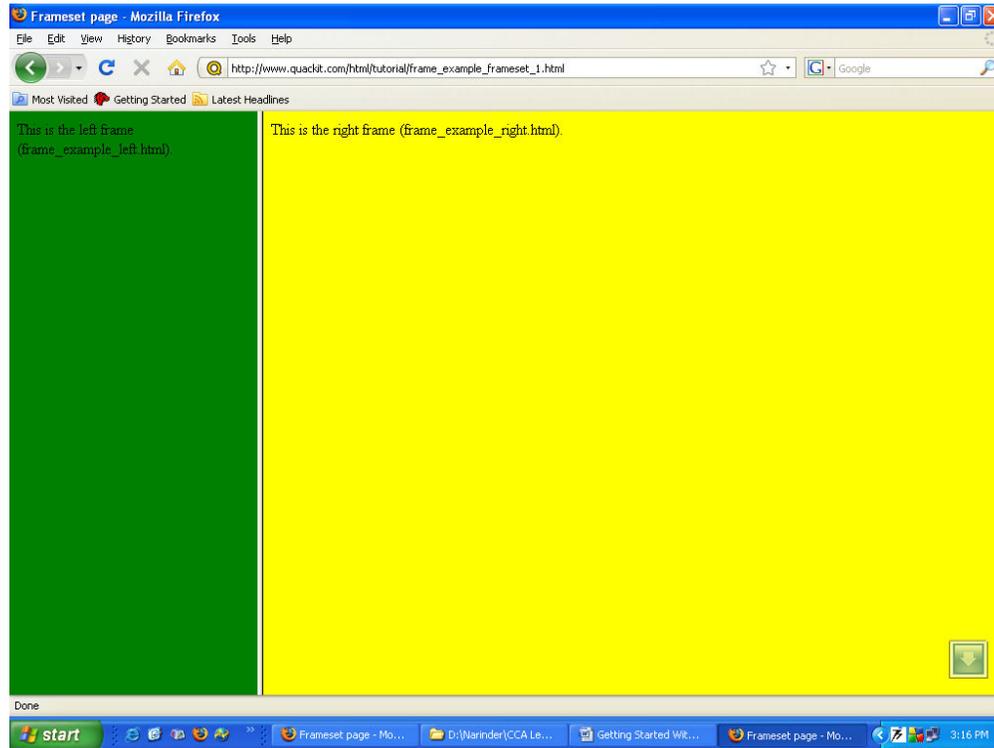
```
<html>
<body style="background-color:green">
<p>This is the left frame (frame_example_left.html).</p>
</body>
</html>
```

The right frame (frame\_example\_right.html):

```

<html>
<body style="background-color:yellow">
<p>This is the right frame (frame_example_right.html).</p>
</body>
</html>

```

**Output :-****Add a Top Frame**

You can do this by "nesting" a frame within another frame.

HTML Code:

The frameset (frame\_example\_frameset\_2.html):

```

<html>
<head>
<title>Frameset page</title>
</head>
<frameset rows="20%,*">
  <frame src="/html/tutorial/frame_example_top.html">
</frameset cols = "25%, *">
  <frame src = "/html/tutorial/frame_example_left.html" />
  <frame src = "/html/tutorial/frame_example_right.html" />

```

```
</frameset>
```

```
</frameset>
```

```
</html>
```

The top frame (frame\_example\_top.html):

```
<html>
```

```
<body style="background-color:maroon">
```

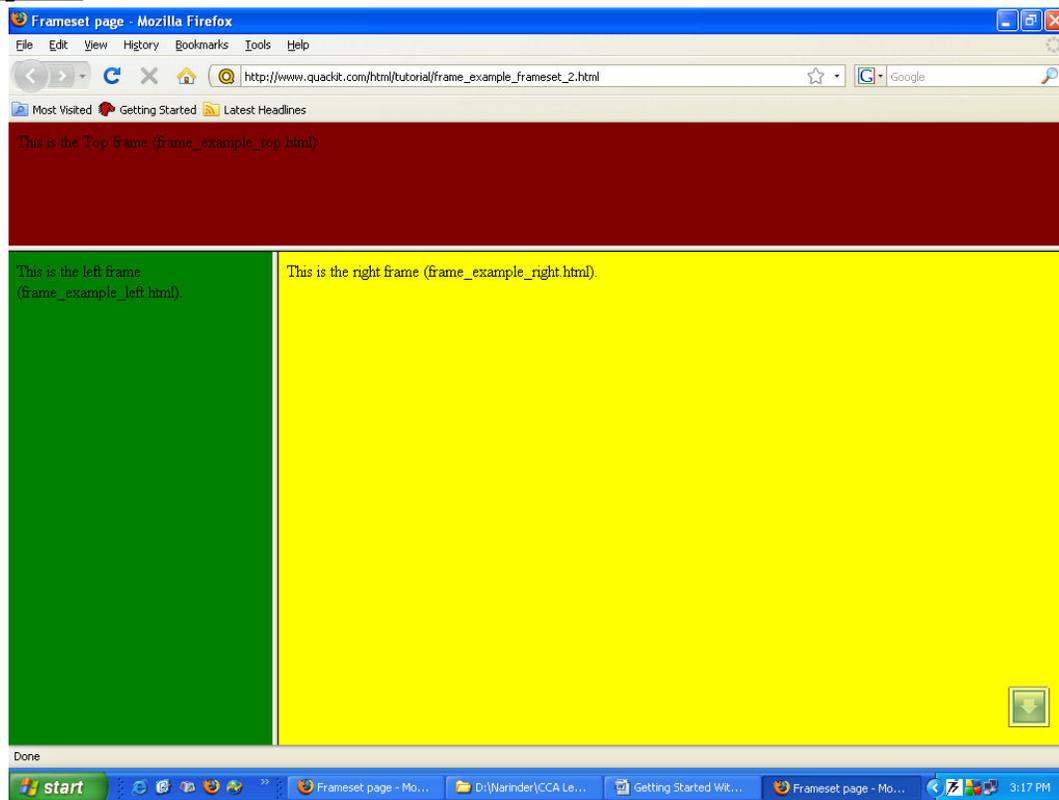
```
<p>This is the Top frame (frame_example_top.html).</p>
```

```
</body>
```

```
</html>
```

(The left and right frames don't change)

### **Output :-**



### **Remove the Borders**

You can get rid of the borders if you like. Officially, you do this using `frameborder="0"`. I say, officially because this is what the HTML specification specifies. Having said that, different browsers support different attributes, so for maximum browser support, use the `frameborder`, `border`, and `framespacing` attributes.

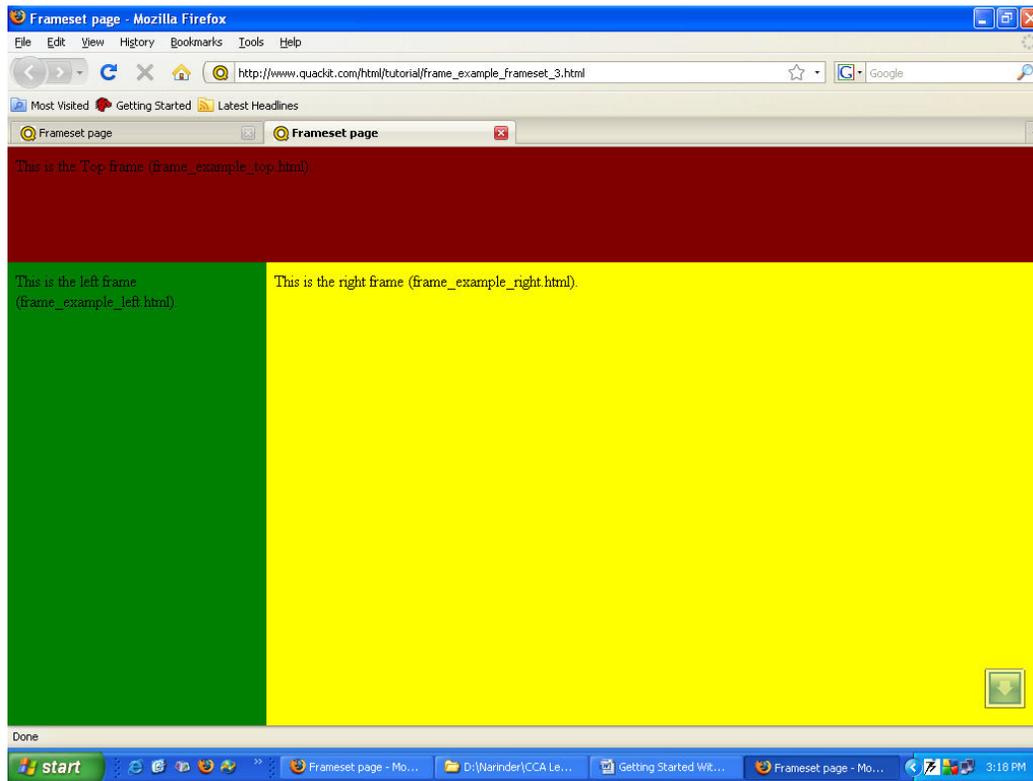
HTML Code:

The frameset (frame\_example\_frameset\_3.html):

```
<html>
<head>
<title>Frameset page</title>
</head>
<frameset border="0" frameborder="0" framespacing="0" rows="20%,*">
  <frame src="/html/tutorial/frame_example_top.html">
<frameset cols = "25%, *">
  <frame src = "/html/tutorial/frame_example_left.html" />
  <frame src = "/html/tutorial/frame_example_right.html" />
</frameset>
</frameset>
</html>
```

(The left, right, and top frames don't change)

### **Output:-**



### **Load Another Frame**

Most websites using frames are configured so that clicking a link in one frame loads another frame. A common example of this is having a menu in one frame, and the main body in the other (like our example).

This is achieved using the name attribute. You assign a name to the target frame, then in your links, you specify the name of the target frame using the target attribute.

Tip: You could use base target="content" at the top of your menu file (assuming all links share the same target frame). This would remove the need to specify a target frame in each individual link.

HTML Code:

The frameset (frame\_example\_frameset\_4.html):

```
<html>
<head>
<title>Frameset page</title>
</head>
<frameset border="0" frameborder="0" framespacing="0" cols = "25%, *">
  <frame src ="/html/tutorial/frame_example_left_2.html" />
  <frame name="content" src ="/html/tutorial/frame_example_yellow.html" />
</frameset>
</html>
```

The left frame (frame\_example\_left\_2.html):

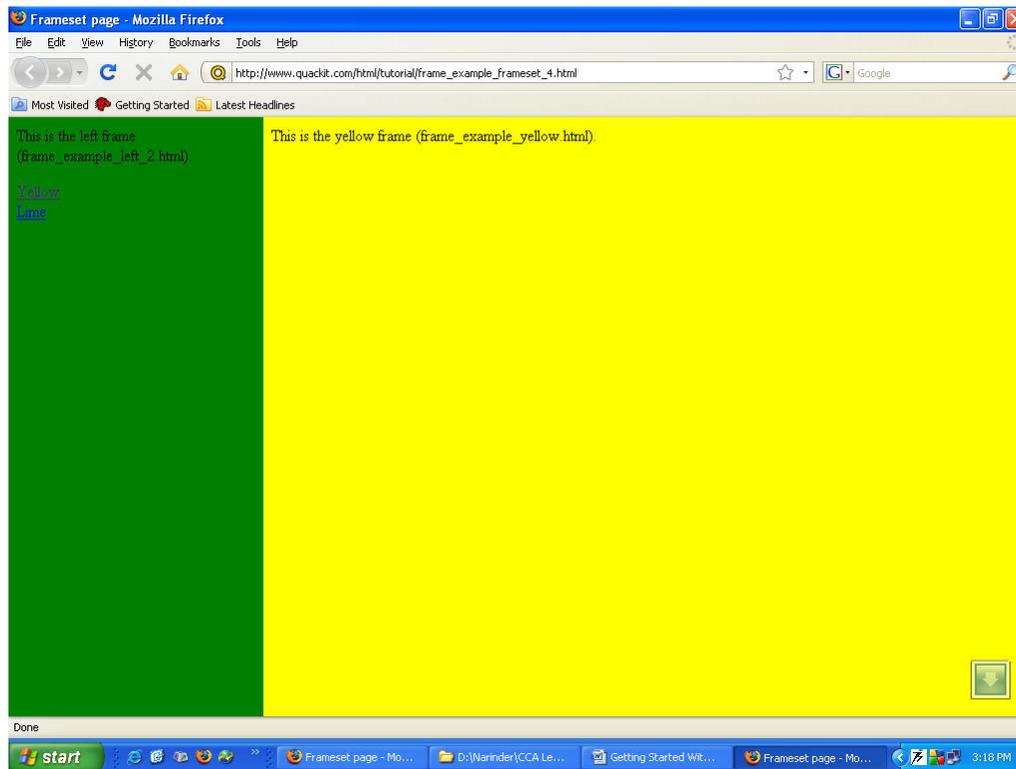
```
<html>
<body style="background-color:green">
<p>This is the left frame (frame_example_left_2.html).</p>
<p>
<a target="content" href="frame_example_yellow.html">Yellow</a><br />
<a target="content" href="frame_example_lime.html">Lime</a>
</p>
</body>
</html>
```

The yellow frame (frame\_example\_yellow.html):

```
<html>
<body style="background-color:yellow">
<p>This is the yellow frame (frame_example_yellow.html).</p>
</body>
</html>
```

The lime frame (frame\_example\_lime.html):

```
<html>
<body style="background-color:Lime">
<p>This is the lime frame (frame_example_lime.html).</p>
</body>
</html>
```

**Output:-****Tag Reference**

Here's some more info on the above tags.

**The frameset Tag**

In your frameset tag, you specify either cols or rows, depending on whether you want frames to go vertically or horizontally.

Attribute	Description
rows	Specifies the number of rows and their height in either pixels, percentages, or relative lengths. Default is 100%
cols	Specifies the number of columns and their width in either pixels, percentages, or relative lengths. Default is 100%

**The frame Tag**

For each frame you want to display, you specify a frame tag. You nest these within the frameset tag.

Attribute	Description
name	Assigns a name to a frame. This is useful for loading contents into one frame from another.

longdesc	A long description - this can elaborate on a shorter description specified with the title attribute.
src	Location of the frame contents (for example, the HTML page to be loaded into the frame).
noresize	Specifies whether the frame is resizable or not (i.e. whether the user can resize the frame or not).
scrolling	Whether the frame should be scrollable or not (i.e. should scrollbars appear). Possible values: <ul style="list-style-type: none"> <li>• auto</li> <li>• yes</li> <li>• no</li> </ul>
frameborder	Whether the frame should have a border or not. Possible values: <ul style="list-style-type: none"> <li>• 1 (border)</li> <li>• 0 (no border)</li> </ul>
marginwidth	Specifies the margin, in pixels, between the frame's contents and its left and right margins.
marginheight	Specifies the margin, in pixels, between the frame's contents and its top and bottom margins.

### The noframe Tag

The noframes tag is used if the user's browser doesn't support frames. Anything you type in between the noframes tags is displayed in their browser.

HTML Code:

```
<html>
<head>
<title>Frameset page</title>
</head>
<frameset cols = "25%, *">
  <noframes>
    <body>Your browser doesn't support frames.
    Therefore, this is the noframe version of the site.</body>
  </noframes>
  <frame src ="frame_example_left.html" />
  <frame src ="frame_example_right.html" />
</frameset>
</html>
```

#### 4. HTML Entities

HTML entities (also known as *character entity references*) enable you to use characters that aren't supported by the document's character encoding or your keyboard. For example, you can type `&copy;` to output a copyright symbol.

You can also use *numeric character references*. Numeric character references can be defined with either a decimal or hexadecimal value (although decimal is more common). The numeric character reference for the copyright symbol is `&#169;` (decimal) and `&#xA9;` (hexadecimal).

A key benefit that numeric character references have over entities is that they can be used to specify any unicode character, whereas entities can only specify some.

##### Entity Example

Many web developers use entities for outputting HTML code to the browser (or at least, outputting one of the characters used by the HTML language).

Lets say you have a website about web design. And lets say you want to display the tag required for creating a horizontal rule 20% wide. Well, if you type the tag (`<hr width="20%" />`) into your code, the browser is simply going to render it as it should be (rather than output the actual code example). Like this...

---

... you see, instead of **displaying** the code, the browser **rendered** the code.

What you should have typed in is this:

```
&lt;hr width="50%" /&gt;
```

The `&lt;` replaces the `<` and the `&gt;` replaces the `>`.

#### 5. HTML Layouts

You may have noticed that many websites have multiple columns in their layout - they are formatted like a magazine or newspaper. Many websites achieved this HTML layout using tables.

##### 5.1 HTML layout with tables

Tables have been a popular method for achieving advanced layouts in HTML. Generally, this involves putting the whole web page inside a big table. This table has a different column or row for each main section.

For example, the following HTML layout example is achieved using a table with 3 rows and 2 columns - but the header and footer column spans both columns (using the `colspan` attribute):

This HTML code...

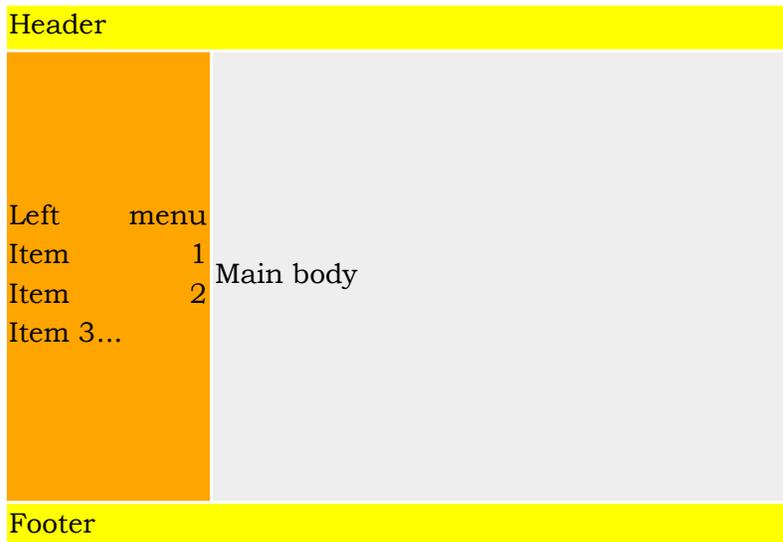
```
<table width="400px" border="0">
  <tr>
    <td colspan="2" style="background-color:yellow;">
      Header
    </td>
```

```

</tr>
<tr>
  <td style="background-color:orange;width:100px;text-align:top;">
    Left menu<br />
    Item 1<br />
    Item 2<br />
    Item 3...
  </td>
  <td style="background-color:#eeeeee;height:200px;width:300px;text-align:top;">
    Main body
  </td>
</tr>
<tr>
  <td colspan="2" style="background-color:yellow;">
    Footer
  </td>
</tr>
</table>

```

produces this layout...



## 5.2 HTML layout with DIV, SPAN and CSS

Although we can achieve pretty nice layouts with HTML tables, tables weren't really designed as a layout tool. Tables are more suited to presenting tabular data. The [div](#) element is a block level element used for grouping HTML elements. Once grouped, formatting can be applied to the div element and everything contained

within it. While the `div` tag is a block-level element, the HTML [span](#) element is used for grouping elements at an *inline* level.

Using our previous HTML layout, we can use DIV and CSS to achieve a similar effect. The following code...

```
<div style="width:400px">
  <div style="background-color:yellow">
    Header
  </div>
  <div style="background-color:orange;height:200px;width:100px;float:left;">
    Left menu<br />
    Item 1<br />
    Item 2<br />
    Item 3...
  </div>
  <div style="background-color:#eeeeee;height:200px;width:300px;float:right;">
    Main body
  </div>
  <div style="background-color:yellow;clear:both">
    Footer
  </div>
</div>
```

produces this layout...

Header	
Left	menu
Item	1
Item	2
Item 3...	
Main body	
Footer	

One change to this that I'd recommend is that you place the CSS into an external style sheet.

One major benefit of using CSS is that, if you place your CSS in a separate location (i.e. an external style sheet), your site becomes much easier to maintain.

### 5.3 Advanced Layouts

As you build more websites you'll be able to develop more advanced layouts. Having said this, you may still prefer to use a professional designer for your basic site layout. A good designer can develop a professional looking website template that you can use as a starter for your website. From there, you can use your HTML knowledge to customize it as you see fit.

Finding a good designer can be a bit of a challenge (and costly). Fortunately, you can still get a professionally designed website without hiring a designer. Quackit offers you over 15,000 professionally designed templates at a very reasonable cost - most from around \$50 - \$60 per template.

## 6. HTML Styles

HTML is quite limited when it comes to the appearance of its elements. This is not so good if you're trying to give a website a unique look and feel, or you're trying to adhere to a corporate identity.

But, never fear - CSS is here!

CSS stands for Cascading Style Sheets, and its purpose is to enable web authors/designers to apply styles across their websites. With CSS, you can specify a number of style properties for a given HTML element. Each property has a name and a value, separated by a colon (:). Each property declaration is separated by a semi-colon (;).

### Example of Style Sheet Usage

HTML Code:

```
<p style="color:olive;font-size:24px;">HTML Styles with CSS</p>
```

This results in:

HTML Styles with CSS

The above code is an example of *inline styles*. It is called *inline* because we declared the styles within the HTML tag itself. We could also use *embedded* styles or even *external* styles.

Embedded styles refers to declaring all styles in one place (usually the head of the document). Each element then knows to use the style that was previously declared.

External styles refers to creating a separate file that contains all style information. This file is then linked to from as many HTML pages as you like - even the whole site.

## 7. HTML Scripts

A *script* is a small, embedded program that can add interactivity to your website. For example, a script could generate a pop-up alert box message, or provide a dropdown menu.

Because HTML doesn't actually have scripting capability, you need to use the `<script>` tag to generate a script, using a scripting language.

The `<script>` tags tell the browser to expect a script in between them. You specify the language using the `type` attribute. The most popular scripting language on the web is JavaScript.

### 7.1 Adding a Script

You can specify whether to make a script run automatically (as soon as the page loads), or after the user has done something (like click on a link).

In either case, a generally accepted convention is to place your scripts between the <head></head> tags. This ensures that the script is ready to run when it is called.

HTML Code:

```
<script type="text/javascript">
  alert("I am a script. I ran first!")
</script>
```

This would open a JavaScript alert as soon as the page loads.

### 7.2 Triggering a Script

In many cases, you won't want the script to run automatically. You might only want the script to run if the user does something (like hover over a link), or once the page has finished loading.

These actions are called *intrinsic events* (*events* for short). There are 18 pre-defined intrinsic events that can trigger a script to run. You use *event handlers* to tell the browser which event should trigger which script. These are specified as an attribute within the HTML tag.

Lets say you want a message to display in the status bar whenever the user hovers over a link. The act of hovering over the link is an event which is handled by the onmouseover event handler. You add the onmouseover attribute to the HTML tag to tell the browser what to do next.

HTML Code:

```
Treat yourself to a <a href="http://www.great-workout.com/killer-ab-workout.cfm"
onmouseover="window.status='Go on, you know you want to'; return true">Killer
Ab Workout</a>
```

This results in:

Treat yourself to a [Killer Ab Workout](#)

Status bar messages aren't supported by all browsers. If you see no change to the status bar, it's likely that your browser doesn't support this piece of JavaScript.

Before we move on, check out the [list of intrinsic events](#) as specified by HTML 4.01

### 7.3 Calling an External Script

You can also place your scripts into their own file, then call that file from your HTML document. This is useful if you want the same scripts available to multiple HTML documents - it saves you from having to "copy and paste" the scripts into each HTML document. This makes it much easier to maintain your website.

HTML Code:

```
<script type="text/javascript" src="external_scripts.js"></script>
```

#### 7.4 Hide Scripts from Older Browsers

Although most (if not all) browsers these days support scripts, some older browsers don't. If a browser doesn't support JavaScript, instead of running your script, it would display the code to the user. To prevent this from happening, you can simply place HTML comments around the script. Older browsers will ignore the script, while newer browsers will run it.

HTML Code:

```
<script type="text/javascript">
  <-- Hide from older browsers
  alert("I am a script. I ran first!")
  Unhide -->
</script>
```

#### 7.5 Alternate Information for Older Browsers

You can also provide alternative info for users whose browsers don't support scripts (and for users who have disabled scripts). You do this using the `<noscript>` tag.

HTML Code:

```
<script type="text/javascript">
  <-- Hide from older browsers
  alert("I am a script. I ran first!")
  Unhide -->
</script>
<noscript>
You need JavaScript enabled to view this page.
</noscript>
```

#### 7.6 Set a Default Scripting Language

You can specify a default scripting language for all your script tags to use. This saves you from having to specify the language everytime you use a script tag within the page.

HTML Code:

```
<meta http-equiv="Content-Script-Type" content="text/JavaScript" />
```

Note that you can still override the default by specifying a language within the script tag.

### 8. HTML Website Templates

An HTML website template is basically a prebuilt website, where you, the developer, can modify as required. All the HTML/CSS code and images are included in the template, and a good template will be compatible with most modern browsers. Some website templates come with a flash version and a non-flash version (i.e. HTML version).

### Benefits of Using a Website Template

Here are some benefits of using an HTML website template:

- Speeds up development time - you don't need to spend hours designing your website, then converting that into code, then checking for browser compatibility etc.
- More professional looking website - unless you're a very good designer, it will be difficult to achieve a great looking website every time. And, if you're not a good designer, your websites will reflect this.
- Learning - using an HTML website template is a great way to learn from other developers. Once you've downloaded the template, you have full access to the code, so you can learn the tricks the developers used in achieving an effect.
- Cheaper & easier than hiring a professional designer.

### 9. Website Hosting

Knowledge of website hosting is essential if you plan to make your website available for the world to see.

It's one thing to build a website using HTML, but so far in this tutorial, everything we've done has been on our own local machine. Unless you have your own web server with a permanent high-speed connection to the Internet, you'll need a website hosting provider to host your website.

If you already have a hosting provider that's great! But, if you don't, you'll need to start looking for one.

#### Basic Web Hosting Concepts

Whether you already have a web host or you're just beginning, you should know at least a little bit about the following hosting concepts:

- The [different types of web hosting](#) (i.e the differences between shared hosting, dedicated servers, and managed hosting)
- [How to choose a good web hosting provider](#)
- [Domain names](#) (i.e. *yourwebsite.com*)
- [FTP](#) - for uploading your website to the hosting provider
- [Availability](#) - how to monitor your web site's availability
- [Technical support](#) - what to expect from a hosting provider

It can be very difficult for a beginner to find information about this. Furthermore, there are many experienced web developers who don't know exactly what it is their web host does.

### 10. Summary

- We finished how to extend HTML with stylesheets and scripts.

### 11. Self Check Exercise

1. How to Add Script in HTML?
2. What are the Templates?

3. How to Host Website?

**12. Suggested Reading**

1. The Complete Reference-HTML by Powell Thomas, Tata McGraw Hill
2. Khurana R., "HTML" APH Publishing
3. HTML-E. Stephen Mack and Janam Platt, BPB Publications

**DEPARTMENT OF DISTANCE EDUCATION**

**PUNJABI UNIVERSITY, PATIALA**  
**STUDENT'S RESPONSE-SHEET**

Roll No.....  
**Class : B.A Part-III**

**Computer : Paper : BAP-303**  
**Introduction to Computer**  
**Network**  
**and Internet Programming**  
RESPONSE SHEET NO.1 and 2

Date of receipt of lesson.....  
Date of submission of Response-  
Sheet by the student....  
No. of pages attached.....  
Date of receipt in the Department  
.....

Marks obtained.....  
Date & Signature of the Examiner  
.....  
Name & address of the student  
below in BLOCK LETTERS:  
.....  
.....  
.....

---

**Long answer type Questions**

Max Marks = 40

(Attempt any 2 from the following)

2 x 10 = 20 marks

1. What is a computer network ? Why do we need a network ? What are the advantages of the network ?
2. What is a network Topology ? Explain different types of network topologies.
3. EWhat is Internet ? Explain the various services provided by the Internet.
4. What is HTML ? Explain the structure of the HTML page.

**Short answer type Questions** (Attempt any 5 from the following) 5 x 4 = 20 marks

- Twisted pair cable
- Fiber-optic cable
- Email
- FTP
- WWW
- Multimedia
- Tables in HTML
- Anchor tag