Vijay Institute of Management

(Approved by AICTE, New Delhi & Affiliated to Anna University)

MBA – TANCET Counselling Code 683

Phone: 852 6396 369, 852 6396 469, 852 6396 569

B.Com (CA)

FUNDAMENTALS OF INTERNET AND WEB TECHNOLOGIES

(UG Course Material for MK University - Madurai)

Computer Network

A computer network is a set of computers connected together for the purpose of sharing resources. The most common resource shared today is connection to the Internet. Other shared resources can include a printer or a file server. The Internet itself can be considered a computer network.

Computer Network Defined

A **computer network** is a set of connected computers. Computers on a network are called **nodes**. The connection between computers can be done via cabling, most commonly the Ethernet cable, or wirelessly through radio waves. Connected computers can share resources, like access to the Internet, printers, file servers, and others. A network is a multipurpose connection, which allows a single computer to do more.

Types of Network Connections

Computer networks can be broken down historically into **topologies**, which is a technique of connecting computers. The most common topology today is a **collapsed ring**. This is due to the success of a network protocol called the Ethernet. This protocol, or network language, supports the Internet, Local Area Networks, and Wide Area Networks.

Star Topology

A **star topology** is a design of a network where a central node extends a cable to each computer on the network. On a star network, computers are connected independently to the center of the network. If a cable is broken, the other computers can operate without problems. A star topology requires a lot of cabling.

Bus Topology

A **bus topology** is another type of design where a single cable connects all computers and the information intended for the last node on the network must run through each connected computer. If a cable is broken, all computers connected down the line cannot reach the network. The benefit of a bus topology is a minimal use of cabling.

Ring Topology

A similar topology is called a **ring**. In this design, computers are connected via a single cable, but the end nodes also are connected to each other. In this design, the signal circulates through the

network until it finds the intended recipient. If a network node is not configured properly, or it is down temporarily for another reason, the signal will make a number of attempts to find its destination.

A **collapsed ring** is a topology where the central node is a network device called a hub, a router, or a switch. This device runs a ring topology internally and features plugins for cables. Next, each computer has an independent cable, which plugs into the device. Most modern offices have a **cabling closet**, or a space containing a switch device that connects the network. All computers in the office connect to the cabling closet and the switch. Even if a network plug is near a desk, the plug is connected via a cable to the cabling closet.

Types of Networks

There are several different types of computer networks. Computer networks can be characterized by their size as well as their purpose.

The size of a network can be expressed by the geographic area they occupy and the number of computers that are part of the network. Networks can cover anything from a handful of devices within a single room to millions of devices spread across the entire globe.

Some of the different networks based on size are:

- Personal area network, or PAN
- Local area network, or LAN
- Metropolitan area network, or MAN
- Wide area network, or WAN

In terms of purpose, many networks can be considered general purpose, which means they are used for everything from sending files to a printer to accessing the Internet. Some types of networks, however, serve a very particular purpose. Some of the different networks based on their main purpose are:

- Storage area network, or SAN
- Enterprise private network, or EPN
- Virtual private network, or VPN

Let's look at each of these in a bit more detail.

Personal Area Network

A **personal area network**, or **PAN**, is a computer network organized around an individual person within a single building. This could be inside a small office or residence. A typical PAN would include one or more computers, telephones, peripheral devices, video game consoles and other personal entertainment devices.

If multiple individuals use the same network within a residence, the network is sometimes referred to as a home area network, or HAN. In a very typical setup, a residence will have a single wired Internet connection connected to a modem. This modem then provides both wired and wireless connections for multiple devices. The network is typically managed from a single computer but can be accessed from any device.

This type of network provides great flexibility. For example, it allows you to:

- Send a document to the printer in the office upstairs while you are sitting on the couch with your laptop.
- Upload a photo from your cell phone to your desktop computer.
- Watch movies from an online streaming service to your TV.

If this sounds familiar to you, you likely have a PAN in your house without having called it by its name.

Local Area Network

A **local area network**, or **LAN**, consists of a computer network at a single site, typically an individual office building. A LAN is very useful for sharing resources, such as data storage and printers. LANs can be built with relatively inexpensive hardware, such as hubs, network adapters and Ethernet cables.

The smallest LAN may only use two computers, while larger LANs can accommodate thousands of computers. A LAN typically relies mostly on wired connections for increased speed and security, but wireless connections can also be part of a LAN. High speed and relatively low cost are the defining characteristics of LANs.

LANs are typically used for single sites where people need to share resources among themselves but not with the rest of the outside world. Think of an office building where everybody should be able to access files on a central server or be able to print a document to one or more central

printers. Those tasks should be easy for everybody working in the same office, but you would not want somebody just walking outside to be able to send a document to the printer from their cell

phone! If a local area network, or LAN, is entirely wireless, it is referred to as a wireless local area network, or WLAN.

Metropolitan Area Network

A **metropolitan area network**, or **MAN**, consists of a computer network across an entire city, college campus or small region. A MAN is larger than a LAN, which is typically limited to a single building or site. Depending on the configuration, this type of network can cover an area from several miles to tens of miles. A MAN is often used to connect several LANs together to form a bigger network. When this type of network is specifically designed for a college campus, it is sometimes referred to as a campus area network, or CAN.

Wide Area Network

A **wide area network**, or **WAN**, occupies a very large area, such as an entire country or the entire world. A WAN can contain multiple smaller networks, such as LANs or MANs. The Internet is the best-known example of a public WAN.

- **SAN: SAN** (**storage area network**) is a high-speed **network** of storage devices that also connects those storage devices with servers. It provides block-level storage that can be accessed by the applications running on any networked servers. Storage Area Networks help attach remote computer storage devices, such as disk arrays, tape libraries, and optical jukeboxes, to servers in such a manner that that they appear to be locally attached to the operating system.
- **CAN:** A **Controller Area Network** (**CAN** bus) is a vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer.
- **PAN:** A **personal area network** (**PAN**) is a computer **network** used for data transmission amongst devices such as computers, telephones, tablets, personal digital assistants, fax machines and printers, that are located close to a single user.
- **GAN:** A global area network (GAN) is a network used for supporting mobile across an arbitrary number of wireless LANs, satellite coverage areas, etc.

The Open Systems Interconnection (OSI) reference model describes how information from a software application in one computer moves through a network medium to a software application in another computer. The OSI reference model is a conceptual model composed of seven layers,

each specifying particular network functions. The model was developed by the International Organization for Standardization (ISO) in 1984, *and i*t is now considered the primary architectural model for inter computer communications. The OSI model divides the tasks involved with moving information between networked computers into seven smaller, more manageable *task* groups. A task or group of tasks is *then* assigned to each of the seven OSI layers. Each layer is reasonably self-contained, so that the tasks assigned to each layer can be implemented independently. This enables the solutions offered by one layer *to be updated* without adversely affecting the other layers. The *following list details the seven layers of the* Open System Interconnection (OSI) reference model:

- Layer 7--Application layer
- Layer 6--Presentation layer
- Layer 5--Session layer
- Layer 4--Transport layer
- Layer 3--Network layer
- Layer 2--Data Link layer
- Layer 1--Physical layer

Characteristics of the OSI Layers

The seven layers of the OSI reference model can be divided into two categories: *upper layers* and *lower layers*.

The *upper layers* of the OSI model *deal* with application issues *and* generally *are* implemented only in software. The highest layer, *application,* is closest to the end user. Both users and application-layer processes interact with software applications *that* contain a communications component. The term upper layer is sometimes used to refer to any layer above another layer in the OSI model.

The *lower layers* of the OSI model handle data transport issues. The physical layer and data-link layer are implemented in hardware and software. The other lower layers generally *are* implemented only in software. The lowest layer, *the physical layer,* is closest to the physical network medium (the network cabling, for example, *and* is responsible for actually placing information on the medium.

Feature of OSI Model :

1. Big picture of communication over network is understandable through this OSI model.

2. We see how hardware and software work together.
3. We can understand new technologies as they are developed.
4. Troubleshooting is easier by separate networks.
5. Can be used to compare basic functional relationships on different networks.

Functions of Different Layers :

Layer 1: The Physical Layer :

1. It is the lowest layer of the OSI Model.
2. It activates, maintains and deactivates the physical connection.
3. It is responsible for transmission and reception of the unstructured raw data over network.
4. Voltages and data rates needed for transmission is defined in the physical layer.
5. It converts the digital/analog bits into electrical signal or optical signals.
6. Data encoding is also done in this layer.

Layer 2: Data Link Layer :

1. Data link layer synchronizes the information which is to be transmitted over the physical layer.
2. The main function of this layer is to make sure data transfer is error free from one node to another, over the physical layer.
3. Transmitting and receiving data frames sequentially is managed by this layer.
4. This layer sends and expects acknowledgements for frames received and sent respectively. Resending of non-acknowledgement received frames is also handled by this layer.
5. This layer establishes a logical layer between two nodes and also manages the Frame traffic control over the network. It signals the transmitting node to stop, when the frame buffers are full.

Layer 3: The Network Layer :

1. It routes the signal through different channels from one node to other.
2. It acts as a network controller. It manages the Subnet traffic.
3. It decides by which route data should take.
4. It divides the outgoing messages into packets and assembles the incoming packets into messages for higher levels.

Layer 4: Transport Layer :

1. It decides if data transmission should be on parallel path or single path.
2. Functions such as Multiplexing, Segmenting or Splitting on the data are done by this layer
3. It receives messages from the Session layer above it, convert the message into smaller units and passes it on to the Network layer.
4. Transport layer can be very complex, depending upon the network requirements.

Transport layer breaks the message (data) into small units so that they are handled more efficiently by the network layer.

Layer 5: The Session Layer :

1. Session layer manages and synchronize the conversation between two different applications.
2. Transfer of data from source to destination session layer streams of data are marked and are resynchronized properly, so that the ends of the messages are not cut prematurely and data loss is avoided.

Layer 6: The Presentation Layer :

1. Presentation layer takes care that the data is sent in such a way that the receiver will understand the information (data) and will be able to use the data.
2. While receiving the data, presentation layer transforms the data to be ready for the application layer.
3. Languages(syntax) can be different of the two communicating systems. Under this condition presentation layer plays a role of translator.
4. It perfroms Data compression, Data encryption, Data conversion etc.

Layer 7: Application Layer :

1. It is the topmost layer.
2. Transferring of files disturbing the results to the user is also done in this layer. Mail services, directory services, network resource etc are services provided by application layer.

3. This layer mainly holds application programs to act upon the received and to be sent data.

Merits of OSI reference model:

1. OSI model distinguishes well between the services, interfaces and protocols.
2. Protocols of OSI model are very well hidden.
3. Protocols can be replaced by new protocols as technology changes.
4. Supports connection oriented services as well as connectionless service.

Demerits of OSI reference model:

1. Model was devised before the invention of protocols.
2. Fitting of protocols is tedious task.
3. It is just used as a reference model.

INTERNET

The Internet is a global collection of computer networks that are linked together by devices called routers and use a common set of protocols for data transmission known as TCP/IP (transmission control protocol / Internet protocol). The primary purpose of the Internet is to facilitate the sharing of information. There are many different tools used on the Internet to make this possible. Some of the more common tools include email, listservs, newsgroups, telnet, gopher, FTP, and the World Wide Web. Probably the most popular of all Internet tools is the world wide web.

History of Internet:

- ARPANET is the first Internet network. ARPANET stands for Advanced Research Projects Agency Networks.

- ARPANET was introduced by the United States. ARPANET has made the TCP/IP correspondences standard, which characterizes information exchange on the web. In 1972, E-mail was adapted by Ray Tomlinson of BBN to ARPANET. In this, Ray has included @ symbol as address. TCP/IP was introduced in 1982.

- In 1978, the British post office telenet, DATAPAC, and TRANSPAC teamed up to make the main worldwide packet-switched system service, and this was referred to as the IPSS.
- The abbreviation of IPSS is International Packet Switched Service. Network control program was developed by a group called "the network working group".
- NSF, abbreviated as the National Science Foundation, is mainly used to create the similar and parallel network called NSF net.

Internet Protocol: A system must contain the IP address and it should be unique.

- The Internet protocol consists of two protocols: Transmission Control Protocol (TCP) and the Internet Protocol (IP).
- The Internet Protocol determines basic applications, for example, electronic mail, terminal imitating, and record exchange.
- Internet protocol architecture consists of four layers, they are:
- **Data link layer**: The data link layer is used for the encoding, decoding and logical organization of data bits. Data packets are framed and addressed by this layer.
- **Network layer**: The main aim of this layer is to deliver packets from source to destination across multiple links (networks). If two computers (system) are connected to the same link then there is no need for a network layer. It routes the signal through different channels to the other end and acts as a network controller.
- **Transport layer**: The main aim of the transport layer is to be delivered the entire message from source to destination. It decides if data transmission should be on the parallel path or single path.
- **Application layer**: It consists of protocols that focus on process-to-process communication across an IP network and provides a firm communication interface and end-user services.
- **World Wide Web**: A web server is a computer that provides web services to the client. A page hosted on the internet is known as web page. It can be viewed by a browser. A browser can help locate a website on the internet. The World Wide Web (WWW) permits user to view multi-media based documents like graphics, animations, audios and/or videos and any subject. In 1990, the World Wide Web was introduced by Tim Berners-Lee of CERN.

- **E-Mail**: Email is an electronic mail. It is used to send and receive the messages. It consists of two components like message header and message body. The message header contains added addresses and the body contains any information and sends any attached contents. The Internet makes your work easy by communication technologies.

Uses of the internet:

1. Checking emails: It is reported that the first thing that 90 % of Internet users do when they connected to the net is to check if they have received any emails.

2. Download: It is hard to find an Internet user, who has not used it to download music and movies. Apart from it, there are lots of other things that can be downloaded using the internet as well.

3. Search for relevant information: If you are not sure about something, then one of the easiest ways to know more about it is by searching it on the internet.

4. Book Tickets: Online booking has made things really easy. it may be movie ticket or flight ticket, you can book just about anything using the Internet. The process is very easy, convenient and super-fast.

5. Shop Online: The use of internet is not limited merely to booking tickets. With help of net, you can do a full-fledged online shopping. On popular e-stores like Ebay, Amazon etc., one can find almost anything that he needs to buy.

6. Make Friends: One of the best things about internet is that is has made communication very easy and convenient. We can make friends through social networks like Facebook, twitter etc.

7. Socialize: Social networks have got really big since last decade. Facebook and Twitter are the new online sites who like to share all the latest happening of their life on these social networks and keep their profile duly updated.

8. E Banking: Banking was never so easy and convenient before! Right from opening an account to operating it, E-Banking has really useful for everyone. We can also do online transactions from the other accounts sitting at the home.

9. Data Sharing: Data sharing was never so easy and quick before! It doesn't matter if you have to send a file or song, a picture or an eBook,; with help of internet, you can send just about anything at anywhere and that too, in no time.

Web Concepts

1. **BROWSER** -- A WWW browser is software on your computer that allows you to access the World Wide Web. Examples include *Netscape Navigator* and *Microsoft Internet Explorer*. Please know that a browser can't work its magic unless you are somehow connected to the Internet. At home, that is normally accomplished by using a modem that is attached to your computer and your phone line and allows you to connect to, or dial-up, an Internet Service Provider (ISP). At work, it may be accomplished by connecting your workplace's local area network to the Internet by using a router and a high speed data line.

2. **HYPERTEXT AND HYPERMEDIA** -- Hypertext is text that contains electronic links to other text. In other words, if you click on hypertext it will take you to other related material. In addition, most WWW documents contain more than just text. They may include pictures, sounds, animations, and movies. Documents with links that contain more than just text are called hypermedia.

3. **HTML (HYPERTEXT MARKUP LANGUAGE)** -- HTML is a set of commands used to create world wide web documents. The commands allow the document creator to define the parts of the document. For example, you may have text marked as headings, paragraphs, bulleted text, footers, etc. There are also commands that let you import images, sounds, animations, and movies as well as commands that let you specify links to other documents. If you wanted to create your own web page, you would need to know HTML or be able to use a tool that can generate HTML such as Claris *HomePage* or Adobe *PageMill*.

4. **URL (UNIFORM RESOURCE LOCATOR)** -- Links between documents are achieved by using an addressing scheme. That is, in order to link to another document or item (sound, picture, movie), it must have an address. That address is called its URL. The URL identifies the host computer name, directory path, and file name of the item. It also identifies the protocol used to locate the item such as hypertext, gopher, ftp, telnet or news. For example, the URL for the main page of the OPEN (Oregon Public Education Network) website http://www.open.k12.or.us

5. **HTTP (HYPERTEXT TRANPORT PROTOCOL)** -- HTTP is the protocol used to transfer hypertext or hypermedia documents.

6. **HOME PAGE** -- A home page is usually the starting point for locating information at a WWW site. Currently, the home page for Roseburg High School's web site is located at http://schools.rosenet.net/roseburg/rhs/

7. **CLIENTS AND SERVERS** -- If a computer has a web browser installed, it is known as a client. A host computer that is capable of providing information to

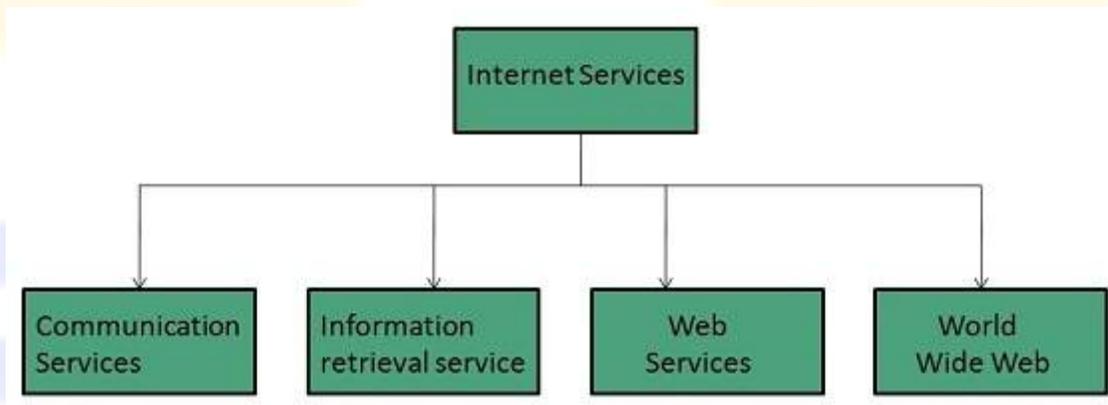others is called a server. A server requires special software in order to provide web documents to others.

Internet Services

Internet Services allows us to access huge amount of information such as text, graphics, sound and software over the internet. Following diagram shows the four different categories of Internet Services.

Communication Services

There are various Communication Services available that offer exchange of information with individuals or groups. The following table gives a brief introduction to these services:

S.N.	Service Description
1	**Electronic Mail** Used to send electronic message over the internet.
2	**Telnet** Used to log on to a remote computer that is attached to internet.
3	**Newsgroup** Offers a forum for people to discuss topics of common interests.

4	**Internet Relay Chat (IRC)** Allows the people from all over the world to communicate in real time.
5	**Mailing Lists** Used to organize group of internet users to share common information through e-mail.
6	**Internet Telephony (VoIP)** Allows the internet users to talk across internet to any PC equipped to receive the call.
7	**Instant Messaging** Offers real time chat between individuals and group of people. Eg. Yahoo messenger, MSN messenger.

Information Retrieval Services

There exist several Information retrieval services offering easy access to information present on the internet. The following table gives a brief introduction to these services:

S.N.	Service Description
1	**File Transfer Protocol (FTP)** Enable the users to transfer files.
2	**Archie** It's updated database of public FTP sites and their content. It helps to search a file by its name.
3	**Gopher** Used to search, retrieve, and display documents on remote sites.
4	**Very Easy Rodent Oriented Netwide Index to Computer Achieved (VERONICA)**

> VERONICA is gopher based resource. It allows access to the information resource stored on gopher's servers.

Web Services

Web services allow exchange of information between applications on the web. Using web services, applications can easily interact with each other.

The web services are offered using concept of **Utility Computing.**

World Wide Web (WWW)

WWW is also known as W3. It offers a way to access documents spread over the several servers over the internet. These documents may contain texts, graphics, audio, video, hyperlinks. The hyperlinks allow the users to navigate between the documents.

HTML (Hypertext Markup Language)

HTML (Hypertext Markup Language) is the set of markup symbols or codes inserted in a file intended for display on a World Wide Web browser page. The markup tells the Web browser how to display a Web page's words and images for the user. Each individual markup code is referred to as an element (but many people also refer to it as a tag). Some elements come in pairs that indicate when some display effect is to begin and when it is to end.

HTML is a formal Recommendation by the World Wide Web Consortium (W3C) and is generally adhered to by the major browsers, Microsoft's Internet Explorer and Netscape's Navigator, which also provide some additional non-standard codes. The current version of HTML is HTML 4.0. However, both Internet Explorer and Netscape implement some features differently and provide non-standard extensions. Web developers using the more advanced features of HTML 4 may have to design pages for both browsers and send out the appropriate version to a user. Significant features in HTML 4 are sometimes described in general as dynamic HTML. What is sometimes referred to as HTML 5 is an extensible form of HTML called Extensible Hypertext Markup Language (XHTML).

SGML (Standard Generalized Markup Language)

SGML (Standard Generalized Markup Language) is a standard for how to specify a document markup language or tag set. Such a specification is itself a document type definition (DTD). SGML is not in itself a document language, but a description of how to specify one. It is metadata.

SGML is based on the idea that documents have structural and other semantic elements that can be described without reference to how such elements should be displayed. The actual display of such a document may vary, depending on the output medium and style preferences. Some advantages of documents based on SGML are:

- They can be created by thinking in terms of document structure rather than appearance characteristics (which may change over time).
    - They will be more portable because an SGML compiler can interpret any document by reference to its document type definition (DTD).
    - Documents originally intended for the print medium can easily be re-adapted for other media, such as the computer display screen.

The language that this Web browser uses, Hypertext Markup Language (HTML), is an example of an SGML-based language. There is a document type definition for HTML (and reading the HTML specification is effectively reading an expanded version of the document type definition). In today's distributed networking environment, many documents are being described with the Extensible Markup Language (XML) which is a data description language (and a document can be viewed as a collection of data) that uses SGML principles.

SGML is based somewhat on earlier generalized markup languages developed at IBM, including General Markup Language (GML) and ISIL.

HTML - Elements: body, head, title

HTML elements have many ranks. All you see: paragraphs, banner, the navigation links from the left side and the right side , all are elements of this page.

An element has three parts: an opening tag, element's content and an closing.

1. **<p>** - the tag that opens a **paragraph**
2. Element's **content** - the paragraph itself.
3. **</p>** - the closing tag.

***Note:

All the web pages will have at least the base elements: ***html, head, title and body***.

<html> element

An HTML document will always begin with a *<html>* tag and will end with *</html>*. This is the standard structure of an HTML. Please open a Notepad and copy the next code:

```
HTML<html>
</html>
```

Save the document from the "File/Save As menu". Select All Files and name the new created file "*index.html*".Hit "Save". Now open the file in a browser so that you have the possibility to refresh the page (F5).

If you did everything well you will be able to see your web page white!

<head> element

The *<head>* element is the one that comes next. While you put it in between *html* and *body*everything should be just fine. "***Head***" has no visible function, so we will talk about this aspect in the next tutorials. Even though I want to mention that *<head>* can offer to the browser very useful information. You can introduce here other cods like JavaScript or CSS.

For the moment we will let this tag empty, and we will introduce the next element from the list, but first lets take a look without him:

```
HTML<html>
        <head>
        </head>
</html>
```

If you save the document and try to refresh the initial page from the browser you will not see any difference. Just have a little patience, because next we will study some visible elements.

<title> element

So that every thing goes well you must put the title tag inside the head tag. what you write between those two title tags (<title> and </title>) will be seen as browser's name, usually in the upper right side. Next we have the source code:

```
HTML<html>
        <head>
                <title> My first web page!</title>
        </head>
</html>
```

Now save the file and open it in your browser. You will be able to see the title in the upper left or right side as like the majority of the browsers.

You can put any name you want, just remember that the descriptive names are the ones that are easier to find later.

<body> element

The body element is the one that defines the beginning of the page content itself (titles, paragraphs, photos, music and any other content). As you can see in the menu from the left, we will talk about all elements one by one.

For now all you need to keep in mind is that body covers all the content of the page.

```
HTML<html>
        <head>
                <title> My first web page!</title>
        </head>


        <body>
```

```
                    Hey guys! Here we will put the content later !
          </body>


</html>
```

Now see what you have done, and after I invite you to see what is in the next section.

HTML - Forms

HTML Forms are required, when you want to collect some data from the site visitor. For example, during user registration you would like to collect information such as name, email address, credit card, etc.

A form will take input from the site visitor and then will post it to a back-end application such as CGI, ASP Script or PHP script etc. The back-end application will perform required processing on the passed data based on defined business logic inside the application.

There are various form elements available like text fields, textarea fields, drop-down menus, radio buttons, checkboxes, etc.

The HTML **<form>** tag is used to create an HTML form and it has following syntax −

```
<form action = "Script URL" method = "GET|POST">
   form elements like input, textarea etc.
</form>
```

Form Attributes

Apart from common attributes, following is a list of the most frequently used form attributes −

Sr.No	Attribute & Description
1	**action** Backend script ready to process your passed data.
2	**method** Method to be used to upload data. The most frequently used are GET and POST methods.

3	**target**
	Specify the target window or frame where the result of the script will be displayed. It takes values like _blank, _self, _parent etc.
4	**enctype**
	You can use the enctype attribute to specify how the browser encodes the data before it sends it to the server. Possible values are −
	application/x-www-form-urlencoded − This is the standard method most forms use in simple scenarios.
	mutlipart/form-data − This is used when you want to upload binary data in the form of files like image, word file etc.

HTML Form Controls

There are different types of form controls that you can use to collect data using HTML form −

- Text Input Controls
- Checkboxes Controls
- Radio Box Controls
- Select Box Controls
- File Select boxes
- Hidden Controls
- Clickable Buttons
- Submit and Reset Button

Text Input Controls

There are three types of text input used on forms −

- **Single-line text input controls** − This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML **<input>** tag.

- **Password input controls** − This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTMl <input> tag.

- **Multi-line text input controls** − This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML **<textarea>** tag.

Single-line text input controls

This control is used for items that require only one line of user input, such as search boxes or names. They are created using HTML <input> tag.

Example

Here is a basic example of a single-line text input used to take first name and last name –

```
<!DOCTYPE html>

<html>


  <head>

    <title>Text Input Control</title>

  </head>


  <body>

    <form >

      First name: <input type = "text" name = "first_name" />

      <br>

      Last name: <input type = "text" name = "last_name" />

    </form>

  </body>

</html>
```

This will produce the following result –

Attributes

Following is the list of attributes for <input> tag for creating text field.

Sr.No	Attribute & Description
1	**type** Indicates the type of input control and for text input control it will be set to **text**.
2	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value** This can be used to provide an initial value inside the control.
4	**size** Allows to specify the width of the text-input control in terms of characters.
5	**maxlength** Allows to specify the maximum number of characters a user can enter into the text box.

Password input controls

This is also a single-line text input but it masks the character as soon as a user enters it. They are also created using HTML <input>tag but type attribute is set to **password**.

Example

Here is a basic example of a single-line password input used to take user password –

```
<!DOCTYPE html>
```

```
<html>

  <head>
    <title>Password Input Control</title>
  </head>

  <body>
    <form >
      User ID : <input type = "text" name = "user_id" />
      <br>
      Password: <input type = "password" name = "password" />
    </form>
  </body>

</html>
```

This will produce the following result –

Attributes

Following is the list of attributes for <input> tag for creating password field.

Sr.No	Attribute & Description
1	**type**

		Indicates the type of input control and for password input control it will be set to **password**.
2	**name**	Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value**	This can be used to provide an initial value inside the control.
4	**size**	Allows to specify the width of the text-input control in terms of characters.
5	**maxlength**	Allows to specify the maximum number of characters a user can enter into the text box.

Multiple-Line Text Input Controls

This is used when the user is required to give details that may be longer than a single sentence. Multi-line input controls are created using HTML <textarea> tag.

Example

Here is a basic example of a multi-line text input used to take item description −

```
<!DOCTYPE html>

<html>
```

```
<head>

  <title>Multiple-Line Input Control</title>

</head>


<body>

  <form>

    Description : <br />

    <textarea rows = "5" cols = "50" name = "description">

      Enter description here...

    </textarea>

  </form>

</body>

</html>
```

This will produce the following result –

Attributes

Following is the list of attributes for <textarea> tag.

Sr.No	Attribute & Description

1	**name**
	Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**rows**
	Indicates the number of rows of text area box.
3	**cols**
	Indicates the number of columns of text area box

Checkbox Control

Checkboxes are used when more than one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **checkbox.**

Example

Here is an example HTML code for a form with two checkboxes −

```
<!DOCTYPE html>
```

```
<html>

  <head>

    <title>Checkbox Control</title>

  </head>


  <body>

    <form>

      <input type = "checkbox" name = "maths" value = "on"> Maths

      <input type = "checkbox" name = "physics" value = "on"> Physics

    </form>

  </body>


</html>
```

This will produce the following result –

Attributes

Following is the list of attributes for <checkbox> tag.

Sr.No	Attribute & Description

1	**type**
	Indicates the type of input control and for checkbox input control it will be set to **checkbox.**.
2	**name**
	Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value**
	The value that will be used if the checkbox is selected.
4	**checked**
	Set to *checked* if you want to select it by default.

Radio Button Control

Radio buttons are used when out of many options, just one option is required to be selected. They are also created using HTML <input> tag but type attribute is set to **radio**.

Example

Here is example HTML code for a form with two radio buttons −

```
<!DOCTYPE html>
```

```
<html>

  <head>
    <title>Radio Box Control</title>
  </head>

  <body>
    <form>
      <input type = "radio" name = "subject" value = "maths"> Maths
      <input type = "radio" name = "subject" value = "physics"> Physics
    </form>
  </body>

</html>
```

This will produce the following result –

Attributes

Following is the list of attributes for radio button.

Sr.No	Attribute & Description

1	**type**
	Indicates the type of input control and for checkbox input control it will be set to radio.
2	**name**
	Used to give a name to the control which is sent to the server to be recognized and get the value.
3	**value**
	The value that will be used if the radio box is selected.
4	**checked**
	Set to *checked* if you want to select it by default.

Select Box Control

A select box, also called drop down box which provides option to list down various options in the form of drop down list, from where a user can select one or more options.

Example

Here is example HTML code for a form with one drop down box

```
<!DOCTYPE html>

<html>


  <head>

    <title>Select Box Control</title>

  </head>


  <body>

    <form>

      <select name = "dropdown">

        <option value = "Maths" selected>Maths</option>

        <option value = "Physics">Physics</option>

      </select>

    </form>

  </body>

</html>
```

This will produce the following result –

Attributes

Following is the list of important attributes of <select> tag −

Sr.No	Attribute & Description
1	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**size** This can be used to present a scrolling list box.
3	**multiple** If set to "multiple" then allows a user to select multiple items from the menu.

Following is the list of important attributes of <option> tag −

Sr.No	Attribute & Description
1	**value** The value that will be used if an option in the select box box is selected.
2	**selected** Specifies that this option should be the initially selected value when the page loads.
3	**label** An alternative way of labeling options

File Upload Box

If you want to allow a user to upload a file to your web site, you will need to use a file upload box, also known as a file select box. This is also created using the <input> element but type attribute is set to file.

Example

Here is example HTML code for a form with one file upload box −

```
<!DOCTYPE html>

<html>


  <head>

    <title>File Upload Box</title>

  </head>


  <body>

    <form>

      <input type = "file" name = "fileupload" accept = "image/*" />

    </form>

  </body>


</html>
```

This will produce the following result −

Attributes

Following is the list of important attributes of file upload box −

Sr.No	Attribute & Description
1	**name** Used to give a name to the control which is sent to the server to be recognized and get the value.
2	**accept** Specifies the types of files that the server accepts.

Button Controls

There are various ways in HTML to create clickable buttons. You can also create a clickable button using <input>tag by setting its type attribute to **button**. The type attribute can take the following values −

Sr.No	Type & Description
1	**submit** This creates a button that automatically submits a form.
2	**reset** This creates a button that automatically resets form controls to their initial values.
3	**button** This creates a button that is used to trigger a client-side script when the user clicks that button.
4	**image** This creates a clickable button but we can use an image as background of the button.

Example

Here is example HTML code for a form with three types of buttons −

```
<!DOCTYPE html>

<html>


  <head>

    <title>File Upload Box</title>

  </head>



  <body>

    <form>

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

      <input type = "reset" name = "reset"  value = "Reset" />

      <input type = "button" name = "ok" value = "OK" />

      <input type = "image" name = "imagebutton" src = "/html/images/logo.png" />

    </form>

  </body>

</html>
```

This will produce the following result −

Hidden Form Controls

Hidden form controls are used to hide data inside the page which later on can be pushed to the server. This control hides inside the code and does not appear on the actual page. For example, following hidden form is being used to keep current page number. When a user will click next

page then the value of hidden control will be sent to the web server and there it will decide which page will be displayed next based on the passed current page.

Example

Here is example HTML code to show the usage of hidden control −

```
<!DOCTYPE html>

<html>


  <head>

    <title>File Upload Box</title>

  </head>


  <body>

    <form>

      <p>This is page 10</p>

      <input type = "hidden" name = "pagename" value = "10" />

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

      <input type = "reset" name = "reset"  value = "Reset" />

    </form>

  </body>


</html>
```

This will produce the following result −

Java script

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

JavaScript language elements (JavaScript)

The JavaScript™ language elements are based on the ECMAScript Language Specification Standard ECMA-262 (see http://www.ecma-international.org/publications/standards/Ecma-262.htm). The JavaScript described here applies to the server-side interpreter. Client-side scripts are interpreted by the browser.

The following table lists the language elements:

Language element	Description
Statements (JavaScript)	Keywords and syntax that form the framework of a script.
Operators (JavaScript)	Symbols and keywords that assign, compare, and combine values.
Top-level functions (JavaScript)	Functions not associated with any object.
@Functions (JavaScript)	Functions that emulate IBM® Notes @functions.
Implied variable typing (JavaScript)	Ability to associate a variable with a data type to document your code and direct content assist.
Embedding Java in JavaScript	Ability to embed Java code.

Objects overview

Objects in JavaScript, just as in many other programming languages, can be compared to objects in real life. The concept of objects in JavaScript can be understood with real life, tangible objects.

In JavaScript, an object is a standalone entity, with properties and type. Compare it with a cup, for example. A cup is an object, with properties. A cup has a color, a design, weight, a material it is

made of, etc. The same way, JavaScript objects can have properties, which define their characteristics.

In JavaScript, almost "everything" is an object.

- Booleans can be objects (if defined with the **new** keyword)
- Numbers can be objects (if defined with the **new** keyword)
- Strings can be objects (if defined with the **new** keyword)
- Dates are always objects
- Maths are always objects
- Regular expressions are always objects
- Arrays are always objects
- Functions are always objects
- Objects are always objects

All JavaScript values, except primitives, are objects.

JavaScript - The Arrays Object

The **Array** object lets you store multiple values in a single variable. It stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type.

Array Properties

Here is a list of the properties of the Array object along with their description.

Sr.No	Property & Description
1	**constructor** Returns a reference to the array function that created the object.
2	**index** The property represents the zero-based index of the match in the string
3	**input**

	This property is only present in arrays created by regular expression matches.	
4	**length** Reflects the number of elements in an array.	
5	**prototype** The prototype property allows you to add properties and methods to an object.	

In the following sections, we will have a few examples to illustrate the usage of Array properties.

Array Methods

Here is a list of the methods of the Array object along with their description.

Sr.No	Method & Description
1	**concat()** Returns a new array comprised of this array joined with other array(s) and/or value(s).
2	**every()** Returns true if every element in this array satisfies the provided testing function.
3	**filter()** Creates a new array with all of the elements of this array for which the provided filtering function returns true.
4	**forEach()**

		Calls a function for each element in the array.
5	**indexOf()**	Returns the first (least) index of an element within the array equal to the specified value, or -1 if none is found.
6	**join()**	Joins all elements of an array into a string.
7	**lastIndexOf()**	Returns the last (greatest) index of an element within the array equal to the specified value, or -1 if none is found.
8	**map()**	Creates a new array with the results of calling a provided function on every element in this array.
9	**pop()**	Removes the last element from an array and returns that element.
10	**push()**	Adds one or more elements to the end of an array and returns the new length of the array.
11	**reduce()**	Apply a function simultaneously against two values of the array (from left-to-right) as to reduce it to a single value.
12	**reduceRight()**	

	Apply a function simultaneously against two values of the array (from right-to-left) as to reduce it to a single value.
13	**reverse()** Reverses the order of the elements of an array -- the first becomes the last, and the last becomes the first.
14	**shift()** Removes the first element from an array and returns that element.
15	**slice()** Extracts a section of an array and returns a new array.
16	**some()** Returns true if at least one element in this array satisfies the provided testing function.
17	**toSource()** Represents the source code of an object
18	**sort()** Sorts the elements of an array
19	**splice()** Adds and/or removes elements from an array.
20	**toString()** Returns a string representing the array and its elements.

21	**unshift()**
	Adds one or more elements to the front of an array and returns the new length of the array.

In the following sections, we will have a few examples to demonstrate the usage of Array methods.

Dynamic HTML

Dynamic HTML, or **DHTML**, is an umbrella term for a collection of technologies used together to create interactive and animated websites[1] by using a combination of a static markup language (such as HTML), a client-side scripting language (such as JavaScript), a presentation definition language (such as CSS), and the Document Object Model (DOM).[2] The application of DHTML was introduced by Microsoft with the release of Internet Explorer 4 in 1997.

DHTML allows scripting languages to change variables in a web page's definition language, which in turn affects the look and function of otherwise "static" HTML page content, *after* the page has been fully loaded and during the viewing process. Thus the dynamic characteristic of DHTML is the way it functions while a page is viewed, not in its ability to generate a unique page with each page load.

By contrast, a dynamic web page is a broader concept, covering any web page generated differently for each user, load occurrence, or specific variable values. This includes pages created by client-side scripting, and ones created by server-side scripting (such as PHP, Perl, JSP or ASP.NET) where the web server generates content before sending it to the client.

DHTML is differentiated from Ajax by the fact that a DHTML page is still request/reload-based. With DHTML, there may not be any interaction between the client and server after the page is loaded; all processing happens in JavaScript on the client side. By contrast, an Ajax page uses features of DHTML to initiate a request (or 'subrequest') to the server to perform additional actions. For example, if there are multiple tabs on a page, pure DHTML approach would load the contents of all tabs and then dynamically display only the one that is active, while AJAX could load each tab only when it is really needed.

Dynamic HyperText Markup Language (DHTML) **mean?**

Dynamic HyerText Markup Language (DHTML) is a combination of Web development technologies used to create dynamically changing websites. Web pages may include animation,

dynamic menus and text effects. The technologies used include a combination of HTML, JavaScript or VB Script,

CSS and the document object model (DOM).

Designed to enhance a Web user's experience, DHTML includes the following features:

- Dynamic content, which allows the user to dynamically change Web page content
- Dynamic positioning of Web page elements
- Dynamic style, which allows the user to change the Web page's color, font, size or content

Dynamic HyperText Markup Language (DHTML)

While DHTML enhances the website user's experience, the technology may also be frustrating for users when it is used incorrectly. For example, a website menu with flashy DHTML animations can easily confuse user navigation. Another DHTML issue occurs when Web developers attempt to create cross-browser DHTML, which is very difficult.

For Web developers, DHTML poses the following problems:

- It can be difficult to develop and debug because of lack of Web browser and technological support.
- DHTML scripts may not work correctly in various Web browsers.
- The Web page layout may not display correctly when it is developed to display in different screen size combinations and in different browsers.

As a result of these problems, Web developers must determine whether DHTML enhances the user experience in any given context. Most Web developers abandon complex DHTML and use simple cross-browser routines to improve user experience, as opposed to integrating excessive DHTML visual effects.

Document Object Model (*DOM*)

The **Document Object Model (*DOM*)** connects web pages to scripts or programming languages. Usually that means JavaScript, but modelling HTML, SVG, or XML documents as objects is not part of the JavaScript language. The DOM model represents a document with a logical tree. Each

branch of the tree ends in a node, and each node contains objects. DOM methods allow programmatic access to the tree; with them you can change the document's structure, style or content. Nodes can have event handlers attached to them. Once an event is triggered, the event handlers get executed.