Unit-1: Introduction

Internet

- Internet is the word derived from two separate word i.e., international networking, which means interlink of computer networking worldwide which is available to anybody publicly.
- Internet is such a wide network of different interlinked networks which are relating to the business, government, academic, domestic networks, therefore internet is known as the network of all other international networks.
- Internet is a world-wide global system of interconnected computer networks.
- Internet uses the standard internet protocol (TCP/IP).
- Every computer in internet is identified by a unique IP address.
- IP address is a unique set of numbers (Such as 192.168.1.1) which identifies a computer location.
- A special computer DNS (domain name server) is used to give name to the Ip address so that the user can locate a computer by name.
- Internet is accessible to every user all over the world hence called world wide web.
- As the Internet is often called the Network of Networks which is shown in Figure below.

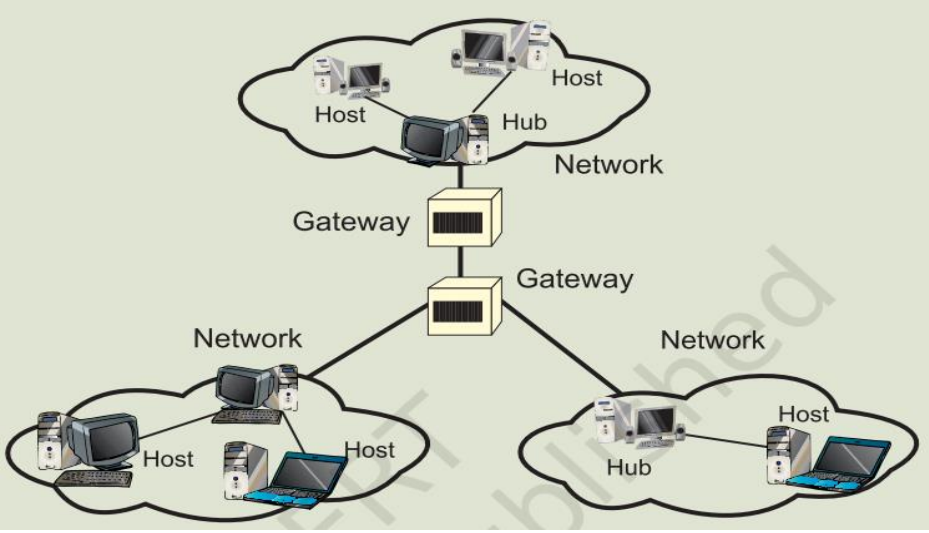


Figure: Network of network

Intranet

Intranets are private networks used by organizations to distribute communications exclusively to their workforce; and they've been used for decades by enterprises for internal communications. Intranet is the result of applying internet technologies to a company, University, College, Hospital or any organization. It is the way of technology adopted by an organization to enhance its productivity by supporting its applications using Internet Standards. It is an "Internal Internet".

Privatization of Internet services will definitely motivate the corporate sector to go for organization wide intranet in a very big way. The evolution of Intranet is dynamic, non-stop and continuum.

The main purpose of an intranet is to share information within the organization and computing resources among employees.

An intranet is described as an internal or local computer network in an organization's private network and a smaller version of the Internet, with websites that appear and function as a typical web-site but are only accessible to authorized internal. Intranet utilizes the same Web server software that provides the public access to websites over the Internet. However, the main difference is that an intranet often restricts access only to members and selected contractors within an organization.

Hence, the difference between the internet and the intranet is that, the internet is a public network that is not owned by an entity, while an intranet is privately owned and not accessible to just anyone who can get online.

Uses of Intranet:

An intranet is primarily used by organizations as a tool to:

- Share company updates
- Store files
- Connect employees
- Collaborate with teams across borders.
- Increase productivity
- Give employees a voice in the company

Function/Application of Intranet:

Share files:

Intranet platform can integrate with existing file-sharing solutions (e.g. Google Drive, Dropbox, OneDrive or others), or it have own its file-storing system.

Search across your organization:

Search is one of the most frequently-used features in an intranet. The best intranets don't just search across information in pages or posts, but also in file names.

Find people in the people directory:

When you have a large organization, finding the one person you need can be a challenge. Almost all intranet platforms have a people directory – the most useful ones are auto-generated. Here you can search for people across location, job title or even skill set, depending on your intranet solution.

Build pages:

A page builder tool is essential for static communication, but it can do much more than just inform. Automate your onboarding journey, house your brand assets, make wiki pages or set up a place to request IT assistance.

Collaborate on documents, spreadsheets and slides:

Modern intranets allow you to directly work on documents, spreadsheets, slides or other frequently-used apps in your productivity suite. Rather than just linking to a file stored in the cloud, editing it directly makes conversation and collaboration revolve around what really matters – work.

Mobile access:

Most modern intranets offer a mobile app, so you can stay connected on the go. Since work is increasingly less dependent on your device, a mobile app simply means that employees can stay engaged, wherever they are.

Promote employee engagement:

Gluing a global workforce together is no easy feat, and keeping them engaged is key. A once-a-year survey just won't cut it, you need a scalable solution to let employees voice concerns, and to listen. That's the great thing about an intranet – with strong analytics features you will be able to measure and gauge the sentiment around your

company, a specific post or an internal campaign. In other words: it truly measures employee engagement.

World-Wide Web (WWW)

World-Wide Web (WWW or W3) commonly known as the Web, is a hypertext-based information system. World Wide Web also known as a Web, is a collection of websites or web pages stored in web servers and connected to local computers through the internet. These websites contain text pages, digital images, audios, videos, etc. Users can access the content of these sites from any part of the world over the internet using their devices such as computers, laptops, cell phones, etc. The WWW, along with internet, enables the retrieval and display of text and media to your device.

Many people think that the internet and the world wide web are the same thing. While they are closely linked, they are very different systems. The internet is a huge network of computers all connected together. The world wide web ('www' or 'web' for short) is a collection of webpages found on this network of computers. Your web browser uses the internet to access the web.

The building blocks of the Web are web pages which are formatted in HTML and connected by links called "hypertext" or hyperlinks and accessed by HTTP. These links are electronic connections that link related pieces of information so that users can access the desired information quickly. Hypertext offers the advantage to select a word or phrase from text and thus to access other pages that provide additional information related to that word or phrase.

A web page is given an online address called a Uniform Resource Locator (URL). A particular collection of web pages that belong to a specific URL is called a website, e.g., www.facebook.com, www.google.com, etc. So, the World Wide Web is like a huge electronic book, whose pages are stored on multiple servers across the world.

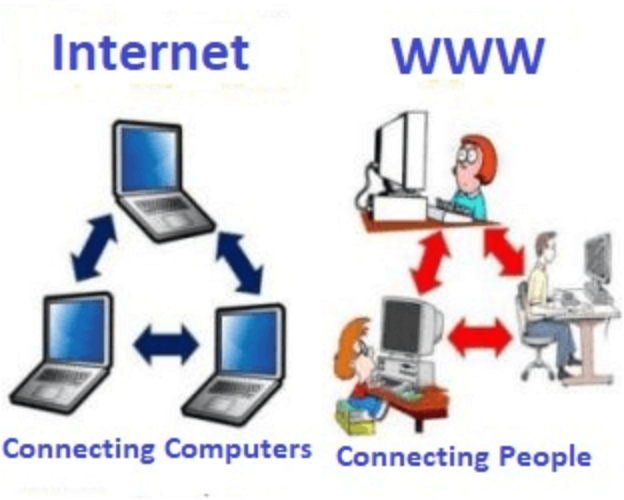
Small websites store all of their WebPages on a single server, but big websites or organizations place their WebPages on different servers in different countries so that when users of a country search their site they could get the information quickly from the nearest server.

So, the web provides a communication platform for users to retrieve and exchange information over the internet. Unlike a book, where we move from one page to another in a sequence, on World Wide Web we follow a web of hypertext links to visit a web page and from that web page to move to other web pages. You need a browser, which is installed on your computer, to access the Web.

Difference between World Wide Web and Internet:

Some people use the terms 'internet' and 'World Wide Web' interchangeably. They think they are the same thing, but it is not so. Internet is entirely different from WWW. It is a worldwide network of devices like computers, laptops, tablets, etc. It enables users to send emails to other users and chat with them online. For example, when you send an email or chatting with someone online, you are using the internet.

But, when you have opened a website like google.com for information, you are using the World Wide Web; a network of servers over the internet. You request a webpage from your computer using a browser, and the server renders that page to your browser. Your computer is called a client who runs a program (web browser), and asks the other computer (server) for the information it needs.



Web:

Key Differences Between the Web and the Internet

- **The Web (World Wide Web):**
  - A **subset** of the Internet.
  - Consists of **web pages** accessed using a **web browser** (e.g., Chrome, Safari).
  - Includes resources like text, images, videos, and hyperlinks.
  - Relies on Internet protocols like **HTTP/HTTPS** to display content.
- **The Internet:**
  - A **global network** of servers, computers, and other devices.
  - Provides the infrastructure for **communication and information sharing**.
  - Supports multiple services, including the Web, email, file transfer, and online gaming.
  - Uses protocols like **TCP/IP** to connect devices and networks.
- **Key Points:**
  - **The Web ≠ The Internet:** The Web is only one part of the Internet.
  - Many people mistakenly use these terms interchangeably.
  - The Internet enables the Web to function by providing the foundation for data exchange.

Web Page:

- A web page (or webpage) is a **document displayed in a web browser** like Firefox, Google Chrome, Safari, Opera, Microsoft Edge, or Internet Explorer.
- Typically written in **HTML (HyperText Markup Language)**.
- Accessed by entering a **URL (Uniform Resource Locator)** into a browser's address bar.
- Can contain:

  - **Text** for conveying information.
  - **Graphics**, pictures, and videos for illustration.
  - **Hyperlinks** to other web pages or files.

Uses of a Web Page

- Provides **information** to viewers on various topics.
- Illustrates content with **multimedia elements** such as pictures and videos.
- Acts as a platform to **sell products or services** to online users.
- Enables users to access other pages via **hyperlinks** or search engine results.

Relationship Between Web Pages and Websites

- **Web pages** are single documents that make up a **website**.
- Clicking a search engine link often leads to an individual web page.
- The Internet comprises **millions of web pages**, with more being added daily.

Types of Web Pages

1. **Static Web Page:**
   - Displays fixed content that doesn't change unless manually updated.
   - Example: An informational page about a company's history.
2. **Dynamic Web Page:**
   - Displays content that changes dynamically based on user interaction or data.
   - Example: An e-commerce product page showing real-time availability and prices.

Characteristics of the webpage:

- Being a part of a website, the web page contains several topics linked to the website.
- Different webpages can use the same name (title name), but they should reside in different documents having different URLs.
- Webpages take less time to be developed as compared to the website.
- The webpage is relatively easy to develop.

Static Web Page:

Static web pages are also known as flat or stationary web page. They are simple and written in the HTML language and stored in web server. Whenever server receives a request regarding a web page, it sends a response along with the requested web page to the client without performing any additional processing. It just locates that page on its hard disk and add HTTP headers, and reply back an HTTP response.

The peculiar thing in a static web page is that the content in these types of the web page does not change depending on the request. They are always the same unless the

content is changed physically on the server's hard disk. That is the reason these web pages are known as static web pages.

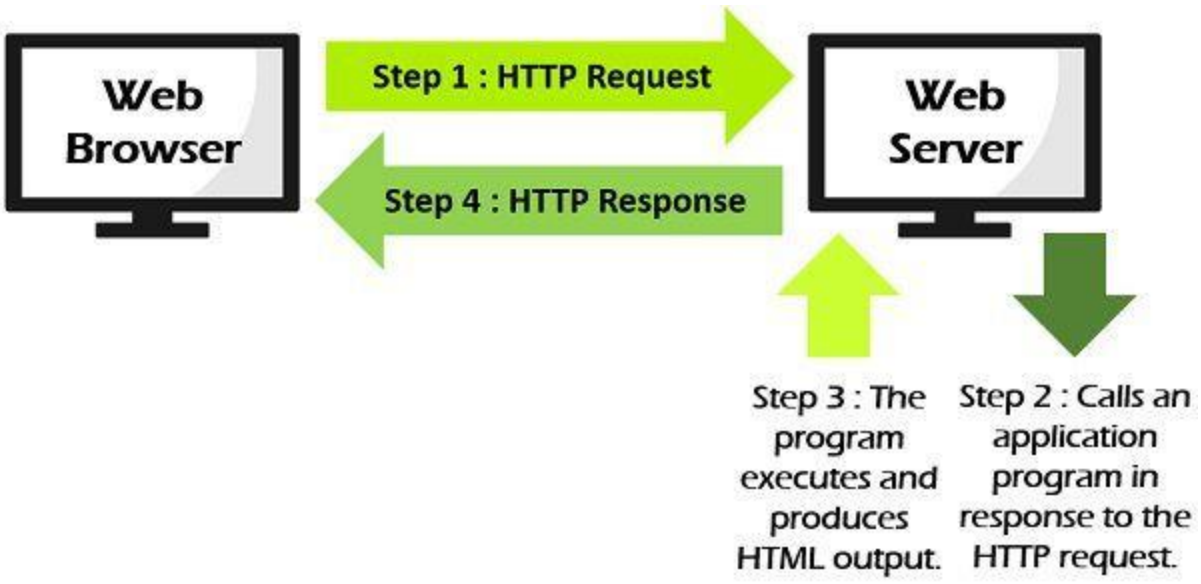
Dynamic Web Page:

Dynamic web page shows different information at different point of time. It is possible to change a portion of a web page without loading the entire web page. It has been made possible using Ajax technology.

Whenever server receives a request regarding a web page, the web server calls a program located on the hard disk which can access a database, perform transaction procedure, etc. If the application program produces HTML output, then it adds HTTP headers, and reply back an HTTP response to the web browser.

The dynamic web pages are employed where the information changes very often such as stock prices, weather information, news and sports updates etc.

Let's assume a person has to physically change the Web page every 10 seconds to show the latest update of the stock prices which is impractical to physically alter the HTML pages very often, so in this case, a dynamic web page can be used.



There are several tools used for the creation of dynamic web pages. For example, CGI (Common Gateway Interface), ASP (Active Server Pages), JSP (Java Server Pages), ASP.NET, AJAX (Asynchronous JavaScript and XML), etc.

Following are the important differences between Static Web Page and Dynamic Web Page:

Key	Static Web Page	Dynamic Web Page
Definition	Static web pages are generally simple HTML written pages which serve as response from server to browser in which all the information and data is static in nature and it does not get changed until someone changed it manually.	On other hand Dynamic webpages are the pages written in some more complex language such as ASP.NET in which data is rendered after some interpretation and capacity to produce distinctive content for different calls.
Complexity	As mentioned in above point as data in static web pages is static and do not require any interpretation before	Dynamic web pages on other hand does the interpretation process which make data dynamic in nature and due

Key	Static Web Page	Dynamic Web Page
	rendering so static web pages are simple in complexity.	to which dynamic web pages become complex in complexity as compare to static web pages.
Language used	Static web pages are generally written in simpler languages such as HTML, JavaScript, CSS, etc.	On other Dynamic web pages are written in more complex languages such as CGI, AJAX, ASP, ASP.NET, etc.
Rendered Data	For static web pages data do not changes until someone changes it manually and hence data is static in nature.	On other hand for Dynamic web page data is first interoperate at server side and due to which it does not remain same on every call and this makes data dynamic in nature.
Time	Static web pages due to static data take less time to get load.	While Dynamic web pages due to dynamic data take comparatively more time as compare to static web pages.
Database	In Static web pages generally no involvement of database for data redecoration.	On other hand in case of Dynamic web page database is used for data redecoration.

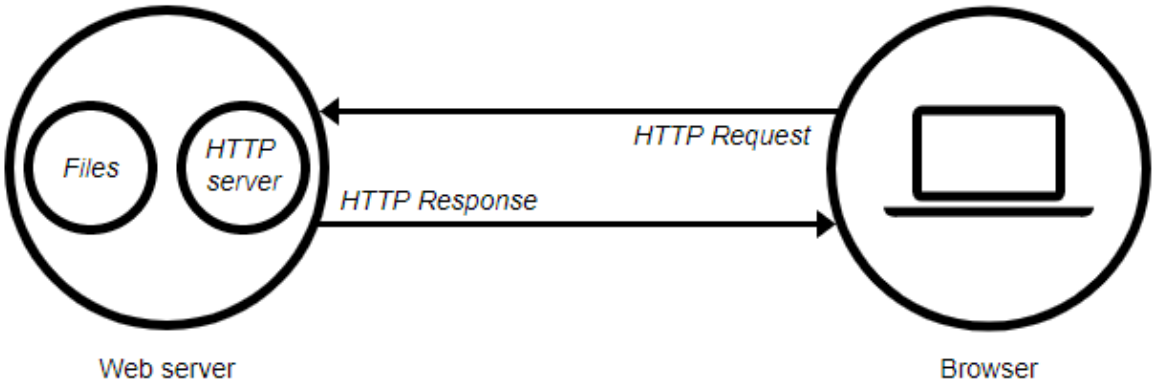
Web Server:

A web server is software and hardware that uses HTTP (Hypertext Transfer Protocol) and other protocols to respond to client requests made over the World Wide Web. The main job of a web server is to display website content through storing, processing and delivering webpages to users. Besides HTTP, web servers also support SMTP (Simple Mail Transfer Protocol) and FTP (File Transfer Protocol), used for email, file transfer and storage.

Hence, the term web server refers to the hardware or software, or both of them working together.

On the hardware side, a web server is a computer that stores web server software and a website's component files. (for example, HTML documents, images, CSS stylesheets, and JavaScript files) A web server connects to the Internet and supports physical data interchange with other devices connected to the web.

On the **software side**, a **web server includes** several parts that control how web users access hosted files. At a minimum, this is an HTTP server. An HTTP server is software that understands URLs (web addresses) and HTTP (the protocol your browser uses to view webpages). An HTTP server can be accessed through the domain names of the websites it stores, and it delivers the content of these hosted websites to the end user's device.



How do web servers work?

When a browser, like Google Chrome or Firefox, needs a file that is hosted on a web server, the browser requests that file via HTTP. When the request reaches the correct (hardware) web server, the (software) HTTP server accepts the request, finds the requested document, and sends it back to the browser, also through HTTP. If the server doesn't find the requested document, it returns a 404 response instead.

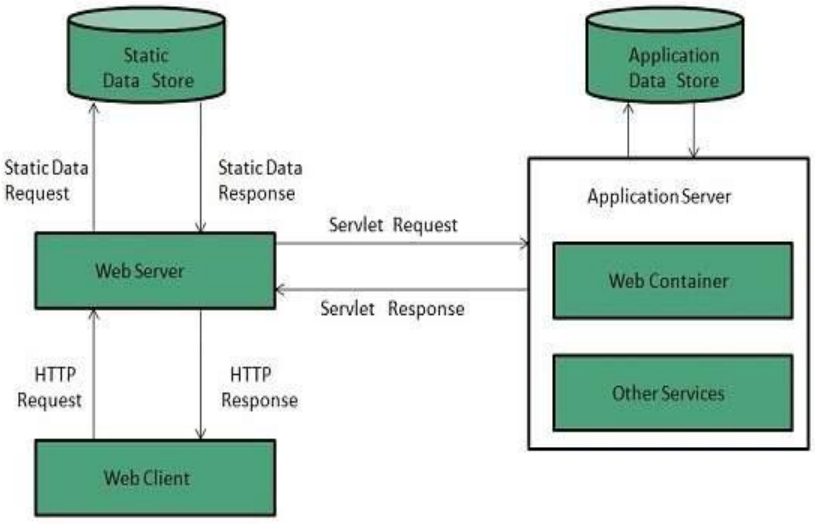
More specifically, when a browser requests a page from a web server, the process will follow a series of steps. First, a person will specify a URL in a web browser's

address bar. The web browser will then obtain the IP address of the domain name either translating the URL through DNS (Domain Name System) or by searching in its cache. This will bring the browser to a web server. The browser will then request the specific file from the web server by an HTTP request. The web server will respond, sending the browser the requested page, again, through HTTP. If the requested page does not exist or if something goes wrong, the web server will respond with an error message. The browser will then be able to display the webpage.

Dynamic vs. static web servers

A web server can be used to serve either static or dynamic content. Static refers to the content being shown as it is, while dynamic content can be updated and changed. A static web server will consist of a computer and HTTP software. It is considered static because the sever will send hosted files as it is to a browser.

Dynamic web browsers will consist of a web server and other software such as an application server and database. It is considered dynamic because the application server can be used to update any hosted files before they are sent to a browser. The web server can generate content when it is requested from the database. Though this process is more flexible, it is also more complicated which is shown in figure below.



Common web server software on the market

- **Apache HTTP Server:** Developed by Apache Software Foundation, it is a free and open-source web server for Windows, Mac OS X, Unix, Linux, Solaris and other operating systems; it needs the Apache license.
- **Microsoft Internet Information Services (IIS):** Developed by Microsoft for Microsoft platforms; it is not open sourced, but widely used.
- **Nginx**: (pronounced engine X) A popular open-source web server for administrators because of its light resource utilization and scalability. It can handle many concurrent sessions due to its event-driven architecture. Nginx also can be used as a proxy server and load balancer.
- **Lighttpd:** A free web server that comes with the FreeBSD operating system. It is seen as fast and secure, while consuming less CPU power.
- **Sun Java System Web Server:** A free web server from Sun Microsystems that can run on Windows, Linux and Unix. It is well-equipped to handle medium to large websites.

Considerations in choosing a web server include how well it works with the operating system and other servers; its ability to handle server-side programming; security characteristics; and the publishing, search engine and site-building tools that come with it. Web servers may also have different configurations and set default values. To create high performance, a web server, high throughput and low latency will help.

Web client:

A web client typically refers to the web browser in the user's machine, computer or mobile devices.

It may also refer to extensions and helper applications that enhance the browser to support special services from the site. It communicates with a web server using the Hypertext Transfer Protocol (HTTP).

A web client is a program capable of communicating with Web servers, requesting and receiving information from them, and processing it for display or other uses. A web browser is one kind of Web client.

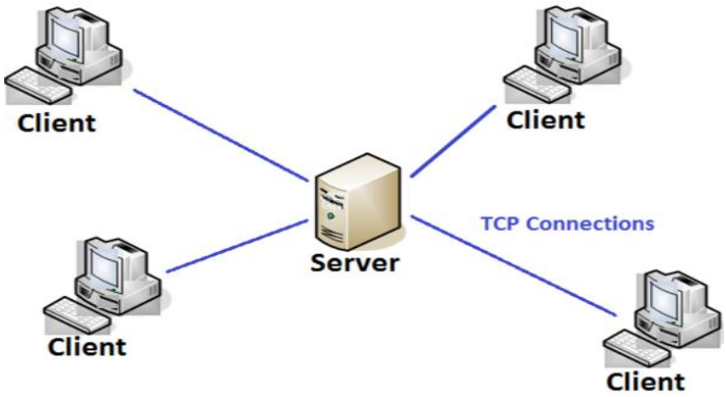
A web client has two parts: dynamic web pages and a web browser. Dynamic web pages are produced by components that run at the web level, and a web browser delivers web pages received from the server.

Client Server Architecture:

Client-server architecture is an architecture of a computer network in which many clients (remote processors) request and receive service from a centralized server (host computer). Client computers provide an interface to allow a computer user to request services of the server and to display the results the server returns. Servers wait for requests to arrive from clients and then respond to them.

Ideally, a server provides a standardized transparent interface to clients so that clients need not be aware of the specification of the system (i.e., the hardware and software) that is providing the service. Clients are often situated at workstations or on personal computers, while servers are located elsewhere on the network, usually on more powerful machines. This computing model is especially effective when clients and the server each have distinct tasks that they routinely perform.

Client/server architecture works when the client computer sends a request for resource or process to the server over the network connection, which is then processed and delivered to the client. A server computer can manage several clients simultaneously, whereas one client can be connected to several servers at a time, each providing a different set of services.



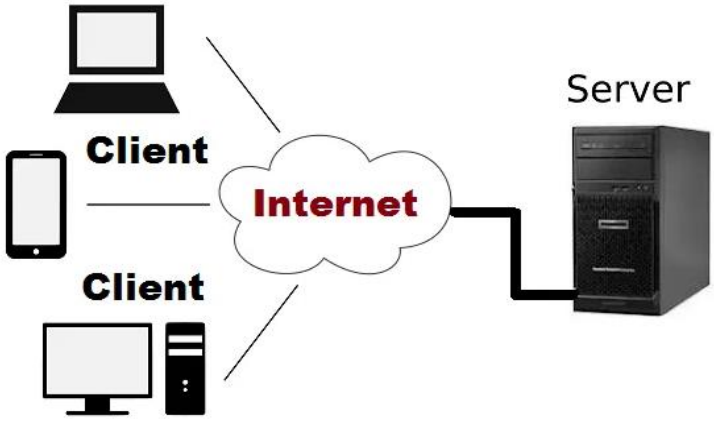


Fig: Client-server architecture

Client-Server Architecture Components:

Working of each architecture requires few components. Similarly, the client-server architecture is based on three interconnected components listed below.

- Workstations
- Servers
- Networking Devices

1. Workstations:

Workstation is the system of users that are sometimes also named as "client's computer". These workstations use various kinds of operating systems. Most of the times MS Windows are used on the workstations. Although, the OS used on the client's workstation is much cheaper than the OS considered for server machines.

2. Server:

An ultra-performer device that retains a fast-processing speed, more storage space and robust memory to deal with multiple requests approaching simultaneously from various workstations. At the same time, a server performs numerous kinds of functions, such as mail servers, database servers, file servers, and domain controllers.

3. Networking Devices:

Workstations and servers are interconnected with each by means of a specific medium. Such a medium is called a network device. Each networking device used in the client-server architecture has its operation and properties.

- For making a connection to a server with various workstation hubs are used.
- For transferring data from a device to another device, repeaters are used.
- For isolated network segmentation bridges are used.

Client-Server Architecture Working:

- A Client-server architecture network design model composed of two elements, client nodes and server nodes, relates to the word client-server. These two elements may communicate and build a network linking several users.
- Computers can connect on a network employing such technology. These networks target the paradigm of file sharing, where the device downloads data from the respective file server and the program uses the received data to operate locally.
- In a client-server architecture, clients are considered as users or consumers, whereas the server performs like a producer. A client often demands high-end computing services from the server to fulfil the requirements.
- Servers facilitate various services to the client that include access for specific applications, storage of data, sharing of files, utilizing the computing power by means of the server.
- A user at its end sends a request to the server using a network that is processed at various stages and then data is delivered to the user according to the submitted request.
- Moreover, servers can entertain multiple consumers simultaneously. Meanwhile, a client has to remain connected with the various servers in a single time as each server is providing services according to its domain.

Types of client-Server Architecture:

There are different types of client server architecture as given below.

- 1-Tier Architecture
- 2-Tier Architecture

- 3-Tier Architecture
- N-Tier Architecture

Before going to explain the types of architecture firstly you need to understand the different layers which are the foundation of software architecture.

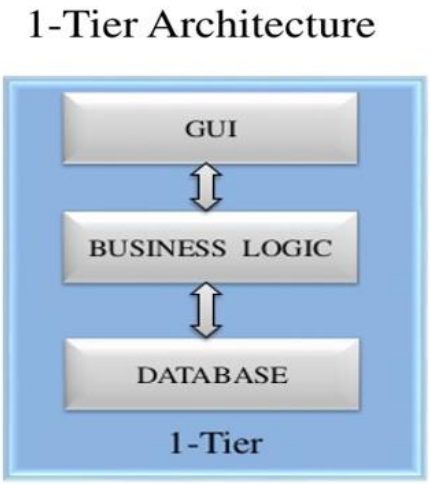
There are four types of layers:

- **Presentation Layer:** This layer is responsible to display the user interface and manage user interaction.
- **Application Layer:** Application layer (also known as Business Layer) has all the business logic, rules and policies. The application layer is a bridge between the presentation layer and the data layer.
- **Data Layer:** This layer is responsible for storing and transfer of the data.
- **Service Layer:** This layer is responsible to define and implement the service interface and the data contracts. Service layer communicates with the application layer

1- Tier Architecture:

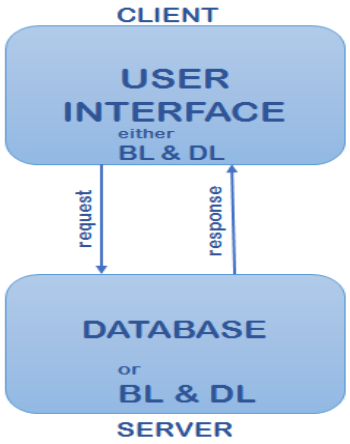
In this type of client server environment, the user interface, business logic & data logic are present in same system. This kind of client server service is cheapest but it is difficult to handle because of data inconsistency that allows repetition of work. Below are the few layers that hold 1 tier architecture.

- Presentation layer
- Business layer
- Data access layer



2-tier architecture:

In this type of client server environment user interface is stored at client machine and database are stored on server. Database logic & business logic are stored at either client or server but it must be unchanged.



If Business Logic & Data Logic are stored at client side, it is called fat client thin server architecture. If Business Logic & Data Logic are stored on server, it is called thin client fat server architecture. This kind of architecture are affordable and comparatively better.

2-tier architecture is useful where a client talks directly to a server. There is no intervening server. It is typically used in small environments. Here, the user interface is placed at user's desktop environment and the DBMS services are usually placed

in a server. Information processing is split between the user system interface environment and the database management server environment.

3-tier architecture:

In this kind of client server environment an additional middle-ware is used, that means client request goes to server through that middle layer and the response of server is firstly accepted by middle-ware and then to client.

This architecture overcomes all the drawbacks of 2-tier architecture and gives best performance. It is costly and easy to handle. The middle-ware stores all the business logic and data access logic.

If there are multiple Business Logic & Data Logic, it is called n-tier architecture. The purpose of middle-ware is to database staging, queuing, application execution, scheduling etc. middle-ware can be file server, message server, application server, transaction processing monitors etc. It improves flexibility and gives best performance.

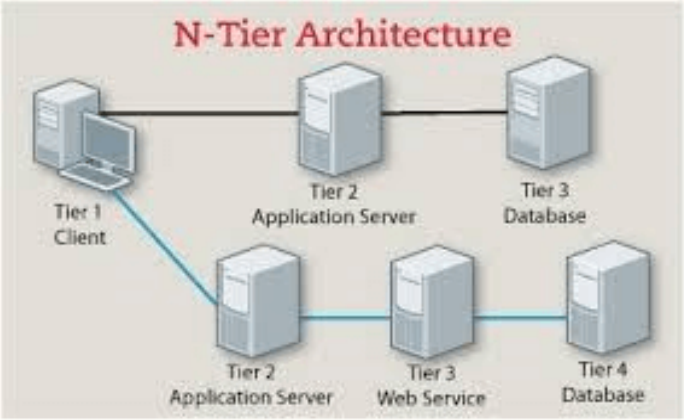
The middle-ware are implemented in different ways such as transaction processing monitors, message servers or application servers. The middle-ware perform several functions like queuing, application execution, database staging. It also adds scheduling and prioritization for work in progress.

3-tier architecture is used to improve performance for large number of users. It also improves flexibility when compared to the two-tier approach. but the development environment is more difficult to use than the development of two-tier applications.

N-tier Architecture:

N-tier architecture is also called a Distributed Architecture or Multi-tier Architecture. It is similar to three-tier architecture but the number of the application

server is increased and represented in individual tiers in order to distribute the business logic so that the logic can be distributed.



N-tier architecture, in which presentation, application processing and data management functions are physically and logically separated. That means that the different functions are hosted on several machines or clusters, ensuring that services are provided without resources being shared and, as such, these services are delivered at top capacity.

Client-Server Architecture Characteristics:

The architecture operates through a request and response mechanism. A request is submitted to the server by the client, and the server provides data in response to requested details.

- A standard contact protocol is adopted by the architecture so that devices can communicate with one another easily. In the application layer, all the data transferring protocols are accessible.
- Only a finite amount of client queries may be accommodated by a server at a particular time. In order to reply to all queries, it employs a method targeting priority.
- As the functionality of both client and server are different; therefore, different hardware and software resources are required for both kinds of machines.

Examples of Client Server Architecture

There are four examples of Client Server Architecture. Below explain each one –

- **Web Servers:** Web server, likes as high-performance computer system that can host multiples websites. On this server, to install different types of web server software's like as Apache or Microsoft IIS, which delivers access to hosted several websites on the internet, and these servers are linked with internet through higher speed connection that delivers ultra-data transmission rates.
- **Mail Servers:** Email servers helps to send and receive all emails. Some software's are run on the mail server which allow to administrator to create and handle all email accounts for any domain that is hosted on the server. Mail servers use some protocols for sending and receiving emails such as SMTP, IMAP, and POP3. SMTP protocol helps to fire messages and manages all outgoing email requests. IMAP and POP3 help to receive all messages and handle all incoming mails.
- **File Servers:** File server is dedicated systems that allow users to access for all files. It works like as centralized file storage location, and it can be accessed by several terminal systems.
- **DNS –** DNS stands for "Domain Name Server ", and it has huge database of different types of public IP addresses, and they link with their hostnames

Above mentioned, all types of server help to deliver all resources (like as files, directories, shared devices such as applications and printers) to client terminal like as PCs, smart phones, PDAs, laptops, tablets etc.

Client-Server Architecture Advantages:

- There is a centralized network that has full leverage to control the processes and activities in a client-server architecture.
- The central area of the architecture is used for the storage of data.
- The devices used in architecture can be controlled centrally.
- Network protection, data backup and all other concerned elements are tackled centrally.
- Users have the authority to access all the files at any time residing on the central storage.

- There exists no restriction regarding geography to access the information. One can access any information from any place.
- The architecture can be expanded as the growth requires the scaling of users or any other parameters.
- Clients have the facility to share any resources at various platforms and places.
- The client-server design is implemented on a distributed model basis, ensuring that the server is replaced, restored, upgraded and moved without influencing the client.
- The architecture provides an easy user interface, file finding procedure and management system for keeping all the files in an organized pattern.

Client-Server Architecture Disadvantages:

- The users will suffer if the primary server goes dump.
- The architecture requires a particular OS related to networking.
- Configuration of Hardware components and software tools utilize plenty of costs.
- There is a need for technical as well as skilled staff for maintenance of the network specifically for server machines.
- Numerous requests simultaneously can cause "traffic congestion problems".
- The architecture can mostly be availed in large organizations and enterprises as the cost is unbearable for typical clients.

HTTP: HTTP Request and Response;

What is HTTP?

HyperText Transfer Protocol (HTTP) is used mainly to access data on the World Wide Web. HTTP is a Server and Client communication Protocol, which is primarily set of rules for formatting and transferring webpage data (text, images, video and Multimedia files) over the World Wide Web. It is the Protocol used to create communication between Web Servers and Web Users. HTTP is an application layer Protocol that works on the top of the TCP/IP suite of Protocols.

HTTP protocol uses server and client model. It acts as a request-response protocol. For Example, A client who uses a web browser and a server is a Web host that hosts the website. Whenever a client transmits a request to the Website server, HTTP protocol proceeds that request and creates a connection between client and server through TCP. After that, HTTP sends a request to the server, which picks up the requested data, and HTTP sends the response back to the client.

The idea of the HTTP protocol is very simple. A client sends a request, which looks like mail, to the server. The server sends the response, which looks like a mail reply, to the client. The request and response messages carry data in the form of a letter with a MIME (Multipurpose Internet Mail Extensions)-like format. The commands from the client to the server embedded in a letter-like request message. The contents of the requested file or other information embedded in a letter-like response message.

HTTP request

- An Http request message consists of a request line, headers and sometimes a body.
- An HTTP request is ways that web browsers ask for information to load website pages. HTTP request contains HTTP version type, a URL, HTTP request headers and HTTP body.
- **HTTP request headers:** HTTP request headers include text information saved in key-value pairs, and these contained in every HTTP request.
- **URL:** A client that wants to access a document needs an address. To facilitate the access of documents distributed throughout the world, HTTP protocol uses the concept of locations. The World Wide Web uses a locator called a URL to identify and intertribal data.
- URL called as (Uniform Resource Locator). A URL is an internet address of any website in common format https://ecomputernotes.com. A URL has three parts: Method: //Host/Path, which used for accessing any file, document or website.

HTTP response

- An HTTP response means when the web client gets the answer back from the webserver. It contains the information that asked for in the HTTP request.
- The HTTP response includes an HTTP status code, HTTP response headers, and an HTTP body.

HTTP status codes:

When a client sends a request to the server, and if the HTTP request is not correct, the error comes. These errors shown in numeric codes. These also called HTTP status codes.

The server gives these status codes to identify the problem. Suppose a client does not know the difference between codes how a client can solve the problem, so the client has to know about every code.

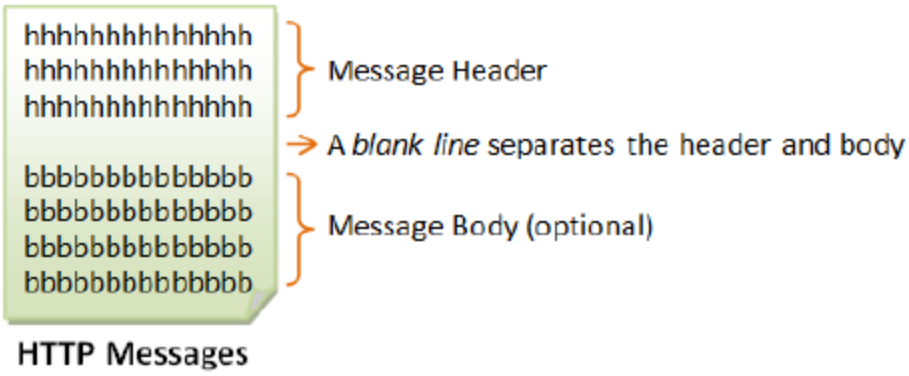
Some standard codes are shown below.

- 401- "Unauthorized."
- 400- "Bad request."
- 404- "File not found."

HTTP Request and Response Messages

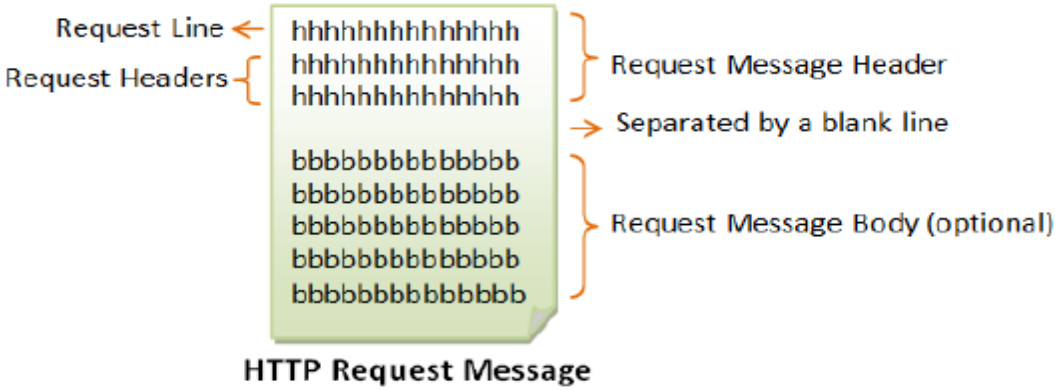
HTTP client and server communicate by sending text messages. The client sends a request message to the server. The server, in turn, returns a response message.

An HTTP message consists of a message header and an optional message body, separated by a blank line, as illustrated below:



HTTP Request Message

The format of an HTTP request message is as follow:



Request Line:

The first line of the header is called the request line, followed by optional request headers. The request line has the following syntax:

request-method-name request-URI HTTP-version

- **request-method-name:** HTTP protocol defines a set of request methods, e.g., GET, POST, HEAD, and OPTIONS. The client can use one of these methods to send a request to the server.
- **request-URI:** specifies the resource requested.
- **HTTP-version:** Two versions are currently in use: HTTP/1.0 and HTTP/1.1.

Examples of request line are:

- GET /test.html HTTP/1.1

- HEAD /query.html HTTP/1.0
- POST /index.html HTTP/1.1

Request Headers:

The request headers are in the form of name:value pairs. Multiple values, separated by commas, can be specified.

Syntax:

request-header-name: request-header-value1, request-header-value2, ...

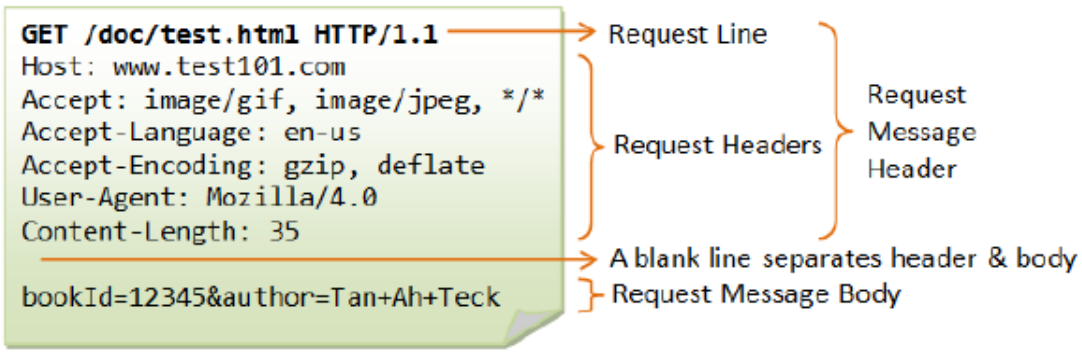
Examples of request headers are:

Host: www.xyz.com

- Connection: Keep-Alive
- Accept: image/gif, image/jpeg, */*
- Accept-Language: us-en, fr, cn

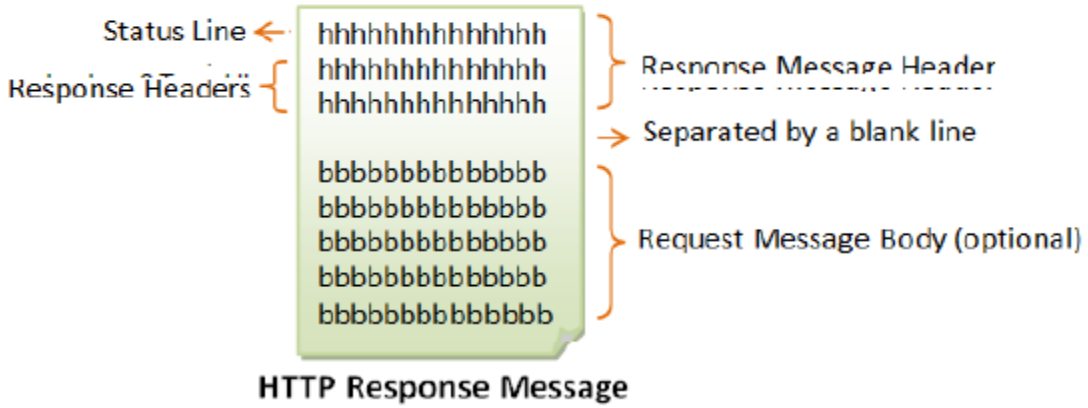
Example:

The following shows a sample HTTP request message:



HTTP Response Message

The format of the HTTP response message is as follows:



Status Line:

The first line is called the status line, followed by optional response header(s).

The status line has the following syntax:

HTTP-version status-code reason-phrase

- **HTTP-version:** The HTTP version used in this session. Either HTTP/1.0 and HTTP/1.1.
- **status-code:** a 3-digit number generated by the server to reflect the outcome of the request.
- **reason-phrase:** gives a short explanation to the status code. Common status code and reason phrase are "200 OK", "404 Not Found", "403 Forbidden", "500 Internal Server Error".

Examples of status line are:

- HTTP/1.1 200 OK
- HTTP/1.0 404 Not Found
- HTTP/1.1 403 Forbidden

Response Headers:

The response headers are in the form name:value pairs:

response-header-name: response-header-value1, response-header-value2, ...

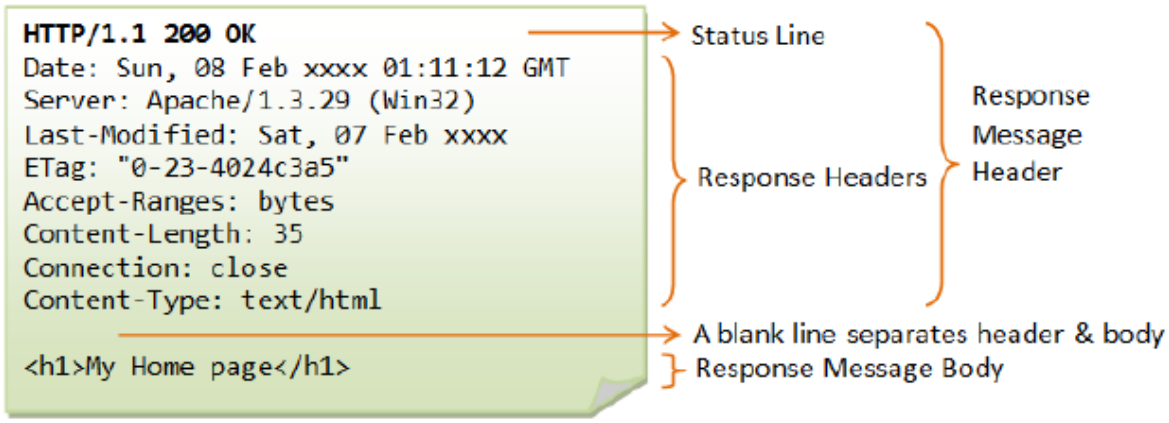
Examples of response headers are:

- Content-Type: text/html
- Content-Length: 35
- Connection: Keep-Alive
- Keep-Alive: timeout=15, max=100

The response message body contains the resource data requested.

Example

The following shows a sample response message:



HTTP Request Methods:

HTTP protocol defines a set of request methods. A client can use one of these request methods to send a request message to an HTTP server. The methods are:

- **GET:** A client can use the GET request to get a web resource from the server.
- **HEAD:** A client can use the HEAD request to get the header that a GET request would have obtained. Since the header contains the last-modified date of the data, this can be used to check against the local cache copy.
- **POST:** Used to post data up to the web server.
- **PUT:** Ask the server to store the data.
- **DELETE:** Ask the server to delete the data.
- **TRACE:** Ask the server to return a diagnostic trace of the actions it takes.
- **OPTIONS:** Ask the server to return the list of request methods it supports.

- **CONNECT:** Used to tell a proxy to make a connection to another host and simply reply the content, without attempting to parse or cache it. This is often used to make SSL connection through the proxy.

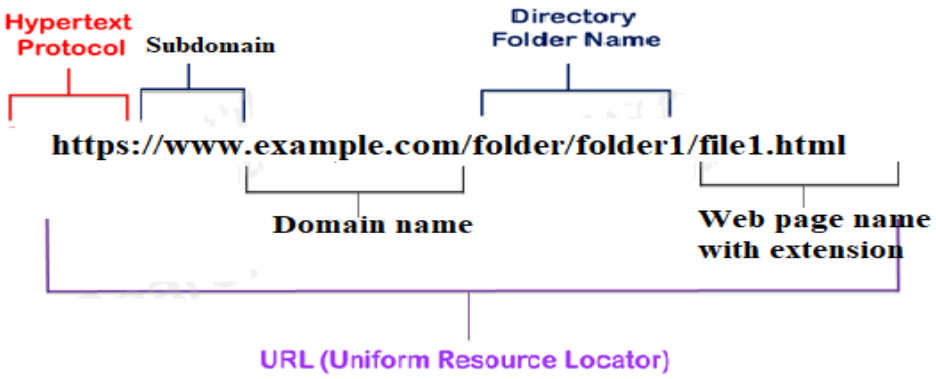
Features of HTTP:

- **Connectionless protocol:** HTTP is a connectionless protocol. HTTP client initiates a request and waits for a response from the server. When the server receives the request, the server processes the request and sends back the response to the HTTP client after which the client disconnects the connection. The connection between client and server exists only during the current request and response time only.
- **Media independent:** HTTP protocol is a media independent as data can be sent as long as both the client and server know how to handle the data content. It is required for both the client and server to specify the content type in MIME (Multipurpose Internet Mail Extensions)-type header.
- **Stateless:** HTTP is a stateless protocol as both the client and server know each other only during the current request. Due to this nature of the protocol, both the client and server do not retain the information between various requests of the web pages.

Uniform Resource Locator (URL)

A Uniform Resource Locator (URL) also known as a Universal Resource Locator, is the address of a resource on the Internet and the protocol used to access it. A URL is nothing more than the address of a given unique resource on the Web. In theory, each valid URL points to a unique resource. Such resources can be an HTML page, a CSS document, an image, etc.

A URL is composed of different parts, some mandatory and others optional. The most important parts are given below



http:// or https://(also called scheme):

The "http" stands for Hypertext Transfer Protocol. It let the browser to know which protocol it is going to use to access the information specified in the domain. An "https" protocol is short for "Hypertext Transfer Protocol Secure" and indicates that information transmitted over HTTP is encrypted and secure. After the http or https is the colon (:) and two forward slashes (//) that separate the protocol from the remainder of the URL.

www (host address or domain name):

Next, "www" stands for World Wide Web and is used to distinguish the content. This portion of the URL is not required and many times can be left out.

For example, typing "http://example.com" would still get you to the example website. This portion of the address can also be substituted for an important subpage known as a subdomain.

example.com:

Next, "cxample.com" is the domain name for the website. The last portion of the domain is known as the domain suffix, or TLD (top-level domain). It is used to identify the type or location of the website. For example, ".com" is short for commercial, ".org" is short for an organization, and ".co.uk" is the United Kingdom. There are several domain suffixes available. To get a domain, you would register the name through a domain registrar.

/folder/folder1/

Next, "folder" and "folder1" are the directories where the web page is on the server. In this example, the web page is two directories deep. To find the file on the server, it would be in the /public_html//folder/folder1 directory. With most servers, the public_html directory is the default directory containing the HTML files.

File1.htm

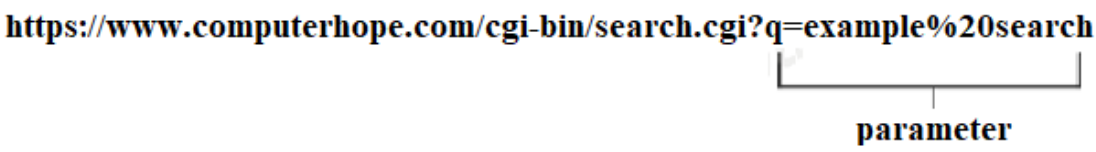
Finally, File1.htm is the actual web page on the domain you're viewing. The trailing .htm is the file extension of the web page that indicates the file is an HTML file. Other common file extensions on the Internet include .html, .php, .asp, .cgi, .xml, .jpg, and .gif. Each of these file extensions performs a different function, like all the different types of files on your computer.

URLs and parameters:

Some URL include string of character after the path being with a question mark called a parameter string.

When a URL points to a script that performs additional functions, additional information (parameters) is added to the end of the URL. For example, a search engine URL pointing to a search results page includes a parameter with the search query words.

Below is an example URL that points to the Computer Hope search page, with the search query parameter of "example search".



In this URL, the script file being pointed to is search.cgi in the cgi-bin directory. Because this file ends with .cgi, it is assumed to be a Perl script. After the script file name is a ? (question mark). The question mark in a URL separates the URL from all the parameters or variables sent to the script.

In the example above, the parameter is q=example%20search. The "q" is a variable name, and the "example%20search" is the value for that variable. Because no spaces are allowed in a URL, the space is encoded as %20. In many scripts, a + (plus) is also used to represent a space.

In our example, because there is a variable the script would use it as it is executed. Scripts are also not limited to only one variable. If the script needs multiple variables, each variable can be separated with a & (ampersand), as shown in the example below.

https://www.computerhope.com/cgi-bin/search.cgi?q=example%20search&example=test

In the above example, there are two different variables. The "q" variable equals "example search" and the "example" variable equals "test". If the script was looking for an example variable, it could be processed and perform an additional feature.

Client-Side and Server-side Scripting:

A script is generally a series of program or instruction, which has to be executed on other program or application. As we know that the web works in a client-server environment.

The scripts can be written in two forms, at the server end or at the client end. The main difference between server-side scripting and client-side scripting is that the server-side scripting involves server for its processing. On the other hand, client-side scripting requires browsers to run the scripts on the client machine but does not interact with the server while processing the client-side scripts.

Client-side scripts:

A **client-side script** is a program that is processed within the client browser. These kinds of scripts are small programs which are downloaded, compiled and run by the browser. JavaScript is an important client-side scripting language and widely used in dynamic websites.

The script can be embedded within the HTML or stored in an external file. External scripts are sent to the client from the server when they are requested. Scripts can also be executed as a result of the user doing something like pressing a page button.

The client-side scripting can be used to examine the user's form for the errors before submitting it and for changing the content according to the user input. Client-side scripts can often be looked at if the user chooses to view the source code of the page.

Example of Client-side scripting languages are:

- **HTML:** It is the fundamental building blocks of web programming which provides the frame to the website. It describes the arrangement of the content.
- **CSS:** CSS provides the way to design the graphic elements which help in making the appearance of the web application more attractive.
- **JavaScript:** It is also a client-side scripting language which essentially devised for the specific purpose, but currently there are various JavaScript frameworks used as server-side scripting technology.

Server-side scripts:

Any scripting or programming that can run on the web server is known as server-side scripting. A server-side script is processed on the web server when the user requests information. These kinds of scripts can run before a web page is loaded. They are needed for anything that requires dynamic data, such as storing user login details.

Some common server-side languages include PHP, Python, Ruby and Java. These execute like programming languages on the server. When a server-side script is processed, the request is sent to the server and the result is sent back to the client. This is useful for websites which store large amounts of data, such as search engines

or social networks - it would be very slow for the client browser to download all the data.

The server-side involves three parts: **server, database, API's and back-end web software** developed by the server-side scripting language. The operations like customization of a website, dynamic change in the website content, response generation to the user's queries, accessing the database, and so on are performed at the server end.

Example of Server-side scripting languages are:

- **PHP:** It is the most prevalent server-side language used on the web which was designed to extract and manipulate information in the database. The language is used in association with SQL language for the Database. It is used in Facebook, WordPress and Wikipedia.
- **Python:** The language is fast and contains shorter code. It is good for beginners as it concentrates on the readability and simplicity of the code. Python functions well in the object-oriented environment and used in famous sites like YouTube, Google, etc.
- **Ruby:** It contains complex logic which packages the back-end with database utility which can also be provided by PHP and SQL.

Web 1.0:

Web 1.0 is the term used to refer to the first stage of development on the World Wide Web that was characterized by simple static websites. Web 1.0 was first implementation of the web and it lasted from 1989 to 2005. It was defined as web of information connections.

According to the innovator of World Wide Web, Tim Berners-Lee considers the Web as "read-only" Web. It provides very little interaction where consumer can exchange the information together but it was not possible to interact with the website. The role of the web was very passive in nature.

Web 1.0 was referred as the first generation of World Wide Web which was basically defined as "It is an information space in which the items of interest referred to as

resources are identified by global identifier called as Uniform Resources Identifiers (URIs)".

First generation Web was era static pages and content delivery purpose only. In other words, the early web allowed us to search for information and read it. There was very little in the way of user interaction or content contribution.

Characteristics of Web 1.0:

Web 1.0 Technologies includes core web protocols: HTML, HTTP and URI. The major characteristics of Web 1.0 are as follow:

- **Static pages:** Pages didn't offer interactive features that changed based on website visitor behavior. At that point websites were largely informational.
- **Website content stored in files:** Virtually every modern website makes use of a database to store the majority of website content. During Web 1.0 this was not the case and most website content was stored directly in the website files, not in a separate database.
- **Combination of content and layout:** Good web design practice today dictates the separation of webpage markup and styling. Virtually every modern website makes use of external style sheets to determine the look and layout of webpages. During Web 1.0 most styling was built into the page markup itself, often by misusing HTML elements such as tables.
- **Proprietary HTML tags:** During Web 1.0 browsers attempted to stand out by offering support for proprietary tags, creating significant incompatibility problems between websites that used these tags and site visitors using unsupported browsers.
- **Guestbooks:** Website visitor comments were usually added to a Guestbook page rather than attached directly to content pages.
- **E-mailing of forms:** Web hosting servers during the Web 1.0 phase rarely offered support for server-side scripting, which is required to use the web server to submit a form. As a result, during Web 1.0, when the Submit button was clicked on most forms the website visitor's e-mail client would launch, and the visitor would have to e-mail their form to an e-mail address provided by the website.

Web 2.0

Web 2.0 is a term that was introduced in 2004 and refers to the second generation of the World Wide Web. The term "2.0" comes from the software industry, where new versions of software programs are labelled with an incremental version number. Like software, the new generation of the Web includes new features and functionality that was not available in the past. However, Web 2.0 does not refer to a specific version of the Web, but rather a series of technological improvements.

"Web 2.0" originated at a brainstorming session with the O'Reilly Network and MediaLive International. Web 2.0 is basically a series of concepts and ideas that redefine the web as a platform where the individual users control their own data. It is the use of new technology to enhance the user's online experience.

Web 2.0 is the term used to describe a variety of web sites and applications that allow anyone to create and share online information or material they have created. A key element of the technology is that it allows people to create, share, collaborate & communicate. Web 2.0 differs from other types of websites as it does not require any web design or publishing skills to participate, making it easy for people to create and publish or communicate their work to the world.

The nature of this technology makes it an easy and popular way to communicate information to either a select group of people or to a much wider audience. The University can make use of these tools to communicate with students, staff and the wider academic community. It can also be an effective way to communicate and interact with students and research colleagues.

There are number of different types of web 2.0 applications including wikis, blogs, social networking, folksonomies, podcasting & content hosting services. Many of the most popular websites are Web 2.0 sites such as Wikipedia, YouTube, Facebook, MySpace, Flickr etc.

The key characteristics of Web 2.0 are:

- Web-based applications can be accessed from anywhere
- Simple applications solve specific problems
- Value lies in content, not the software used to display content
- Data can be readily shared

- Employees and customers can access and use tools on their own
- Social tools encourage people to create, collaborate, edit, categorize, exchange, and promote information
- Network effects are encouraged; the more people who contribute, the better the content gets

Difference between Web 1.0 and Web 2.0

	Web 1.0	Web 2.0
Communication	Broadcast	Interactive
Information	Static / Read-only	Dynamic
Focus	Organization	Community
Personal	Home Pages	Blogs / Wikis
Content	Ownership	Sharing
Interaction	Web Forms	Web Applications
Search	Directories	Keywords / Tags
Metrics	Page Views	Cost Per Click
Advertising	Banners	Interactive
Research	Britannica Online	Wikipedia
Technologies	HTML / FTP	Flash / Java / XML

Web Hosting, Web Application Deployment, and Using Version Control Tools

Web Hosting

Web hosting is a service that allows individuals or organizations to publish their websites or web applications on the internet. Hosting providers offer the technology and infrastructure required to make a website accessible online.

Key Components of Web Hosting

1. **Server**: The physical or virtual machine that stores website files.
2. **Domain Name**: The address used to access the website (e.g., www.example.com).
3. **Bandwidth**: The amount of data transfer allowed between the website and users.
4. **Storage**: Space provided to store files, databases, and other resources.

Types of Web Hosting

1. **Shared Hosting**: Multiple websites share the same server resources.
    - Cost-effective but limited in performance.
2. **VPS (Virtual Private Server) Hosting**: A server is divided into virtualized environments for individual users.
    - Offers better control and performance than shared hosting.
3. **Dedicated Hosting**: A single server is dedicated to one website or application.
    - High performance and control but expensive.
4. **Cloud Hosting**: Uses a network of servers to host websites, providing scalability and reliability.
5. **Managed Hosting**: Hosting provider manages the technical aspects (e.g., updates, security).

Web Hosting Features to Consider

- **Uptime Guarantee**: Minimum expected availability of the website.
- **Support**: Availability of customer service for technical issues.
- **Security**: SSL certificates, firewalls, and malware protection.
- **Scalability**: Ability to handle increased traffic.

Web Application Deployment

Web application deployment involves making an application accessible to users by deploying it to a hosting environment.

Steps for Web Application Deployment

1. **Preparation**:
   - Finalize the application code.
   - Configure environment variables and dependencies.
2. **Selecting Hosting Platform**:
   - Options include AWS, Microsoft Azure, Google Cloud, DigitalOcean, or a custom server.
3. **Deployment Methods**:
   - **Manual Deployment**: Uploading files to a server using FTP/SFTP.
   - **Automated Deployment**: Using CI/CD pipelines to automate the process.
4. **Configuring the Server**:
   - Set up web server software (e.g., Apache, Nginx).
   - Install runtime environments (e.g., Node.js, Python, PHP).
5. **Database Integration**:
   - Set up the database and configure connections.
6. **Testing**:
   - Conduct functional and performance tests in the live environment.
7. **Going Live**:
   - Point the domain to the hosting server.
   - Monitor performance and logs for issues.

Common Deployment Tools and Services

- **Heroku**: Simplified deployment with pre-configured environments.
- **Docker**: Containerization for consistent deployment across environments.
- **CI/CD Tools**: Jenkins, GitHub Actions, or GitLab CI/CD for automated workflows.

Using Version Control Tools

Version control tools manage changes to source code, enabling collaboration and tracking of modifications over time.

Popular Version Control Tools

1. **Git**:
    - Distributed version control system.
    - Common platforms: GitHub, GitLab, Bitbucket.
2. **Subversion (SVN)**:
    - Centralized version control system.
3. **Mercurial**:
    - Similar to Git but less commonly used.

Key Features of Version Control Tools

1. **Branching and Merging**: Work on different features simultaneously without affecting the main codebase.
2. **History Tracking**: Maintain a record of all changes with timestamps and authors.
3. **Collaboration**: Multiple developers can work on the same project simultaneously.
4. **Conflict Resolution**: Manage and resolve code conflicts during merges.

Basic Git Commands

- git init: Initialize a new repository.
- git clone: Clone a repository to your local machine.
- git add: Stage changes for commit.
- git commit: Save changes to the repository.
- git push: Upload local changes to a remote repository.
- git pull: Download changes from a remote repository.

End of Unit-1