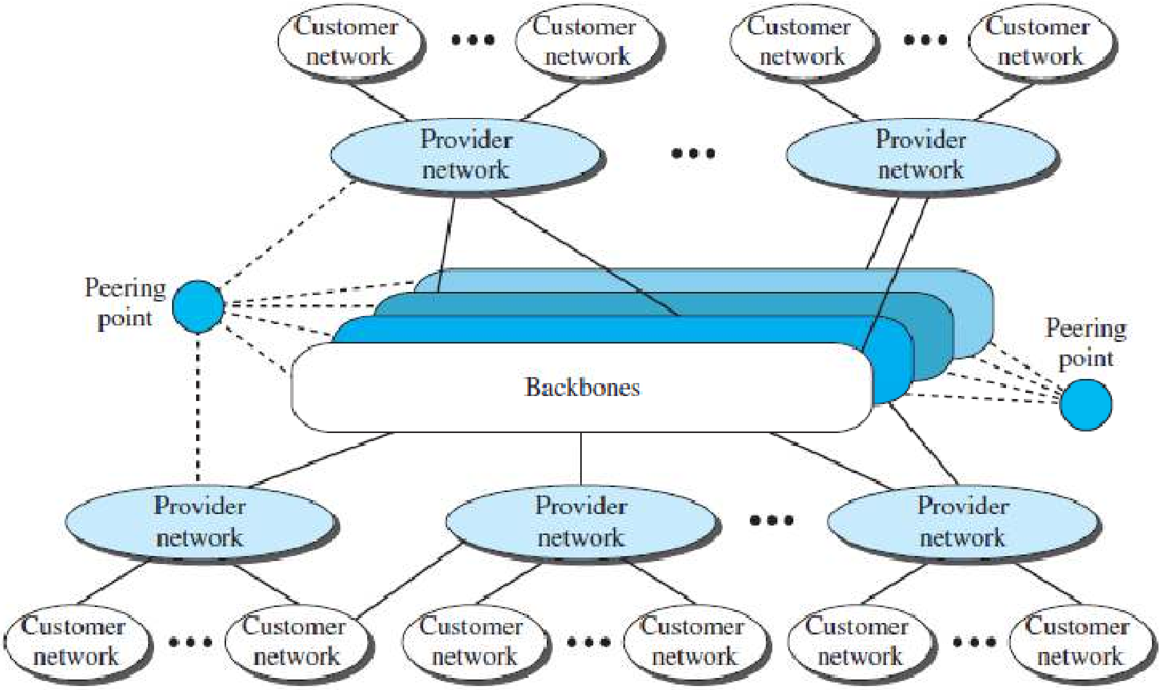
IWT Module-1(BKP)

Internet:

internet: internet (note the lowercase *i*) is two or more networks that can communicate with each other
Internet: The Internet is a global system of interconnected computer networks that use the standard Internet Protocol Suite (TCP/IP) to serve billions of users worldwide



- At the top level, the *backbones* are large networks owned by some communication companies such as Sprint, Verizon (MCI), AT&T, and NTT. The backbone networks are connected through some complex switching systems, called *peering points*.
- At the second level, there are smaller networks, called *provider networks*, that use the services of the backbones for a fee. The provider networks are connected to backbones and sometimes to other provider networks.
- The *customer networks* are networks at the edge of the Internet that actually use the services provided by the Internet. They pay fees to provider networks for receiving services.
- Backbones and provider networks are also called **Internet Service Providers (ISPs).**
- **T**he backbones are often referred to as *international ISPs; T*the provider networks are often referred to as *national* or *regional ISPs*.
- The user, needs to be physically connected to an ISP. The physical connection is normally done through a point-to-point WAN

Brief History of Internet

- Internet has evolved from a private network to a global one in less than 40 years.
- Communication networks, such as telegraph and telephone networks, were suitable for constant-rate communication at that time i.e after a connection was made between two users, the encoded message (telegraphy) or voice (telephony) could be exchanged.
- But a computer network, should be able to handle *bursty* data, i.e data received at variable rates at different times.
- The Department of Depart of Defense of USA(DOD) was interested in finding a way to connect computers so that the researchers they funded could share their findings, thereby reducing costs and eliminating duplication of effort.

- In 1967, at an Association for Computing Machinery (ACM) meeting, ARPA presented its ideas for the **Advanced Research Projects Agency Network (ARPANET),** a small network of connected computers. The idea was that each host computer would be attached to a specialized computer, called an *interface message processor* (IMP). The IMPs, in turn, would be connected to each other. Each IMP had to be able to communicate with other IMPs as well as with its own attached host.

- By 1969, ARPANET was a reality. Four nodes, at the University of California at Los Angeles (UCLA), the University of California at Santa Barbara (UCSB), Stanford Research Institute (SRI), and the University of Utah, were connected via the IMPs to form a network. Software called the *Network Control Protocol* (NCP) provided communication between the hosts.
- In 1972, Vint Cerf and Bob Kahn, both of whom were part of the core ARPANET group, collaborated on what they called the *Internetting Project.* Cerf and Kahn devised the idea of a device called a *gateway* to serve as the intermediary hardware to transfer data from one network to another.
- Cerf and Kahn's landmark 1973 paper outlined the protocols to achieve end-to-end delivery of data. This was a new version of NCP. This paper on transmission control protocol (TCP) included concepts such as encapsulation, the datagram, and the functions of a gateway. A radical idea was the transfer of responsibility for error correction from the IMP to the host machine.
- Around this time, responsibility for the ARPANET was handed over to the Defense Communication Agency (DCA).
- In October 1977, an internet consisting of three different networks (ARPANET,packet radio, and packet satellite) was successfully demonstrated.
- Shortly thereafter, authorities made a decision to split TCP into two protocols: **Transmission Control Protocol (TCP)** and **Internet Protocol (IP).** IP would handle datagram routing while TCP would be responsible for higher level functions such as segmentation, reassembly, and error detection. The new combination became known as TCP/IP.
- In 1981, under a Defence Department contract, UC Berkeley modified the UNIX operating system to include TCP/IP.
- In 1983, authorities abolished the original ARPANET protocols, and TCP/IP became the official protocol for the ARPANET
- In 1983, ARPANET split into two networks: **Military Network (MILNET)** for military users and ARPANET for nonmilitary users.
- Another milestone in Internet history was the creation of CSNET in 1981. **Computer Science Network (CSNET)** was a network sponsored by the National Science Foundation (NSF). The network was conceived by universities that were ineligible to join ARPANET due to an absence of ties to the Department of Defense. CSNET was a less expensive network; there were no redundant links and the transmission rate was slower.
- By the mid-1980s, most U.S. universities with computer science departments were part of CSNET. Other institutions and companies were also forming their own networks and using TCP/IP to interconnect.
- The term *Internet,* originally associated with government-funded connected networks, now referred to the connected networks using TCP/IP protocols.
- NSF in 1986 sponsored the **National Science Foundation Network (NSFNET),** a backbone that connected five supercomputer centers located throughout the United States. Community networks were allowed access to this backbone, In 1995, NSFNET reverted back to its original concept of a research network.
- In 1991, the U.S. government decided that NSFNET was not capable of supporting the rapidly increasing Internet traffic. Three companies, IBM, Merit, and Verizon, filled the void by forming a nonprofit organization called Advanced Network & Services (ANS) to build a new, high-speed Internet backbone called **Advanced Network Services Network (ANSNET).**
- The Internet today is a set of pier networks that provide services to the whole world.

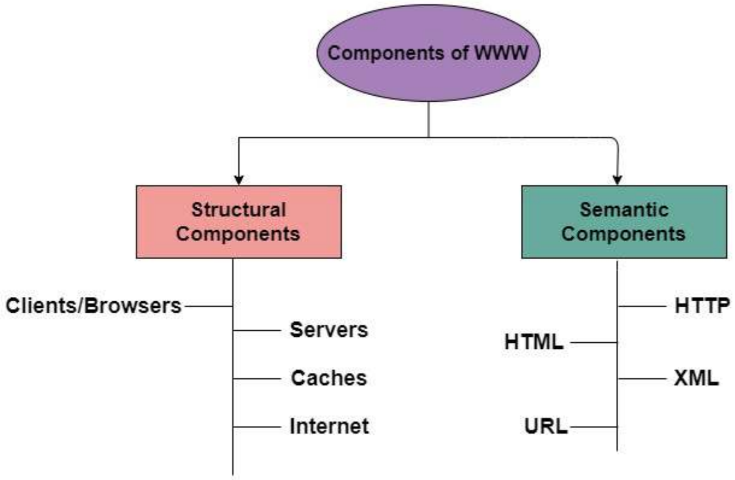
World Wide Web

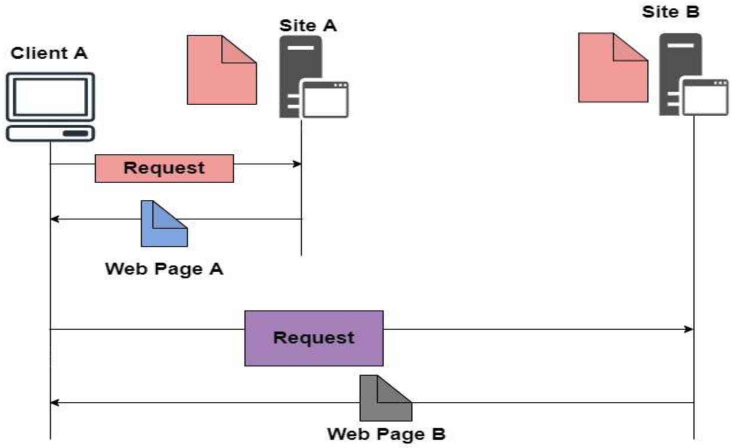
- Internet and the Web are not the same thing. The *Internet* is a collection of computers and other devices connected by equipment that allows them to communicate with each other
- The idea of the Web was first proposed by Tim Berners-Lee in 1989 at *CERN.* The commercial Web started in the early 1990s.
- The world wide web or simply Web is a repository of information in which the electronic documents, called ***webpages,*** are distributed all over the world and related documents are linked together. The linking of web pages was achieved using a concept called ***hypertext(***text with embedded links to text)**.** ***i.e*** web is a system of interlinked hypertext documents accessed via the Internet
- It is basically a way of exchanging information between computers on the Internet.
- The WWW today is a distributed client-server service, in which a client using a browser can access a service using a server. However, the service provided is distributed over many locations called *sites.* Each site holds one or more web pages.
- Each web page, however, can contain some links to other web pages in the same or other sites. In other words, a web page can be simple or composite. Each web page is a file with a name and address.

Components of WWW

The Components of WWW mainly falls into two categories:

1. Structural Components
2. Semantic Components

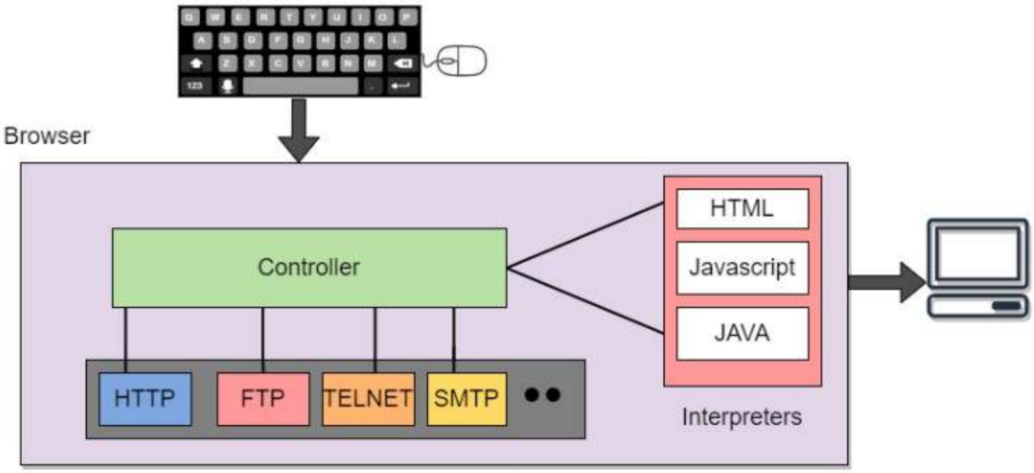




In the above case, the client sends some information that belongs to **site A**. It generally sends a request through its browser (It is a program that is used to fetch the documents on the web) and also the request generally contains other information like the address of the site, web page(URL).
The server at **site A** finds the document then sends it to the client. after that when the user or say the client finds the reference to another document that includes the web page at **site B**.
The reference generally contains the URL of site B. And the client is interested to take a look at this document too. Then after the client sends the request to the new site and then the new page is retrieved.

Client/ Web Browser

- The Client/Web browser is basically a program that is used to communicate with the web server on the Internet. i.e A web browser is the program you use to view pages and navigate the World Wide Web.
- Each browser mainly comprises of three components and these are:
  - Controller
  - Interpreter
  - Client Protocols
- Controller mainly receives the input from the input device, after that it uses the client programs in order to access the documents.
- After accessing the document, the controller makes use of an interpreter in order to display the document on the screen.
- An interpreter can be Java, HTML, javascript mainly depending upon the type of the document.
- The Client protocol can be FTP, HTTP, TELNET.



Web Server

- Web servers are programs that provide documents to requesting browsers. Servers are slave programs: They act only when requests are made to them by browsers running on other computers on the Internet.
- i.e To publish pages on the Web, you need a web server. A web server is the program that runs on a computer and is responsible for replying to web browser requests for files.
- Whenever the request of the client arrives then the corresponding document is sent to the client. The connection between the client and the server is TCP. It can become more efficient through multithreading or multiprocessing. Because in this case, the server can answer more than one request at a time.

- Web browsers initiate network communications with servers by sending them URLs . A URL can specify one of two different things: the address of a data file stored on the server that is to be sent to the client, or a program stored on the server that the client wants executed and the output of the program returned to the client.
- All the communications between a Web client and a Web server use the standard Web protocol, HTTP.
- A Web client, or browser, opens a network connection to a Web server, sends information requests and possibly data to the server, receives information from the server, and closes the connection. Of course, other machines exist between browsers and servers on the network—specifically, network routers and domain name servers
- The primary task of a Web server is to monitor a communications port on its host machine, accept HTTP commands through that port, and perform the operations specified by those commands

URL

- URL is an abbreviation of **the Uniform resource locator.** It is basically a standard used for specifying any kind of Information on the Internet.
- In order to access any page the client generally needs an address.
- To facilitate the access of the documents throughout the world HTTP generally makes use of Locators.
- Uniform (or universal) Resource Identifiers **(URIs)** are used to identify resources (often documents) on the Internet. URIs are used for two different purposes,
  i. to name a resource, in which case they are often called URIs, even though they could be more accurately called Uniform Resource Names (URNs).
  ii. The more commonly used form of URIs is to provide a path to, or location of, a resource, in which case they are called Uniform Resource Locators (URLs).
- The general forms of URIs and URLs are similar, and URIs are often confused with URLs.
- All URLs have the same general format:

scheme:object-address.

- Common schemes include http, ftp, gopher, telnet, file, mailto, and news. Different schemes use object addresses that have different forms
- HTTP protocol supports the Web. This protocol is used to request and send Hypertext Markup Language (HTML) documents. In the case of HTTP, the form of the object address of a URL is as follows:

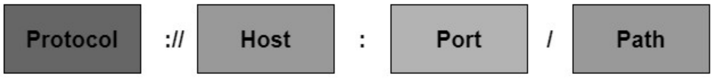
//fully-qualified-domain-name/path-to-document

- Another scheme of interest to us is file. The file protocol means that the document resides on the machine running the browser. This approach is useful for testing documents to be made available on the Web without making them visible to any other browser. When file is the protocol, the fully qualified domain name is omitted, making the form of such URLs as follows:

file: ***//path-to-document***

URL gen format:

- **URL mainly defines the four things**



- **Protocol:** It is a client/server program that is mainly used to retrieve the document. A commonly used protocol is HTTP.

- **Host-Computer:** It is the computer on which the information is located. It is not mandatory because it is the name given to any computer that hosts the web page.
- **Port:** The URL can optionally contain the port number of the server. If the port number is included then it is generally inserted in between the host and path and is generally separated from the host by the colon.
- **Path** It indicates the pathname of the file where the information is located.
  Ex

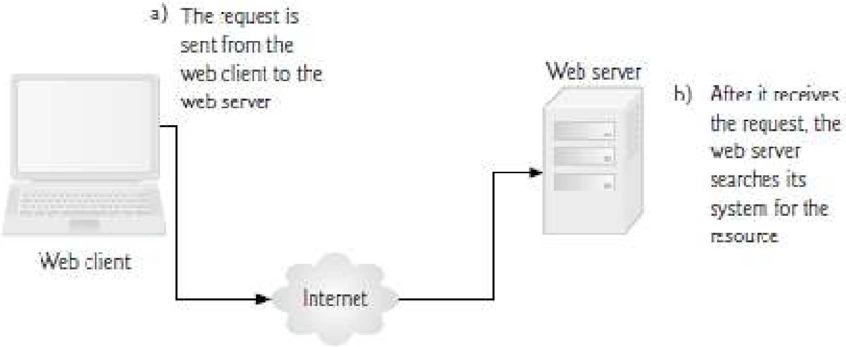
http://www.ABC.com/books/downloads.html

- http:// indicates that the HyperText Transfer Protocol (HTTP) should be used to obtain the resource
- Next in the URL is the server's fully qualified hostname (for example, www.ABC.com) —the name of the web-server computer on which the resource resides. This computer is referred to as the **host.**
- The hostname www.ABC.com is translated into an **IP (Internet Protocol) address**—a numerical value that uniquely identifies the server on the Internet. An Internet **Domain Name System (DNS) server** maintains a database of hostnames and their corresponding IP addresses and performs the translations automatically.
- The remainder of the URL (/books/downloads.html) specifies the resource's location (/books) and name (downloads.html) on the web server. The location could represent an actual directory on the web server's file system. For *security* reasons, however, the location is typically a *virtual directory*. The web server translates the virtual directory into a real location on the server, thus hiding the resource's true location

URI	**URL**
URI is an acronym for Uniform Resource Identifier.	URL is an acronym for Uniform Resource Locator.
URI contains two subsets, URN, which tell the name, and URL, which tells the location.	URL is the subset of URI, which tells the only location of the resource.
All URIs cannot be URLs, as they can tell either name or location.	All URLs are URIs, as every URL can only contain the location.
A URI aims to identify a resource and differentiate it from other resources by using the name of the resource or location of the resource.	A URL aims to find the location or address of a resource on the web.
An example of a URI can be ISBN 0-486-35557-4.	An example of an URL is https://www.javatpoint.com.
It is commonly used in XML and tag library files such as JSTL and XSTL to identify the resources and binaries.	It is mainly used to search the webpages on the internet.
The URI scheme can be protocol, designation, specification, or anything.	The scheme of URL is usually a protocol such as HTTP, HTTPS, FTP, etc.

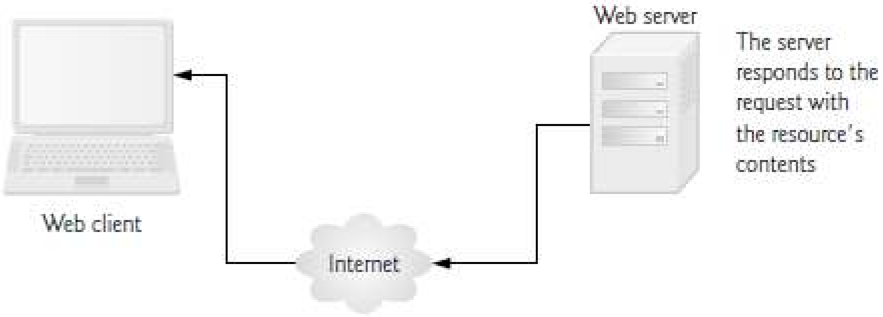
Making a Request and Receiving a Response

- When given a web page URL, a web browser uses HTTP to request the web page found at that address.


a) The request is sent from the web client to the web server
Web server
b) After it receives the request, the web server searches its system for the resource
Web client
Internet

- the web browser sends an HTTP request to the server. The request (in its simplest form) is
  GET /books/downloads.html HTTP/1.1

- **GET** is an **HTTP method** indicating that the client wishes to obtain a resource from the server.
- The remainder of the request provides the path name of the resource and the protocol's name and version number (HTTP/1.1). The client's request also contains some required and optional headers. Any server that understands HTTP (version 1.1) can translate this request and respond appropriately


Web server
The server responds to the request with the resource's contents
Web client
Internet

- The server first sends a line of text that indicates the HTTP version, followed by a numeric code and a phrase describing the status of the transaction.
  For example, HTTP/1.1 200 OK indicates success,
  whereas HTTP/1.1 404 Not found informs the client that the web server could not locate the requested resource.
- Next, the server sends one or more **HTTP headers**, which provide additional information about the data that will be sent. In this case, the server is sending an HTML5 text document, so one HTTP header for this example would read: Content-type: text/html
- The information provided in this header specifies the **Multipurpose Internet Mail Extensions (MIME) type** of the content that the server is transmitting to the browser.
- The MIME standard specifies data formats, which programs can use to interpret data correctly. For example, the MIME type text/plain indicates that the sent information is text that can be displayed directly.
  Similarly, the MIME type image/jpeg indicates that the content is a JPEG image. When the browser receives this MIME type, it attempts to display the image.
- The header or set of headers is followed by a blank line, which indicates to the client browser that the server is finished sending HTTP headers.

- Finally, the server sends the contents of the requested document (downloads.html). The client-side browser then renders (or displays) the document, which may involve additional HTTP requests to obtain associated CSS and images.

Simply speaking When I typed lets say http://www.example.org in the Location bar and pressed the Enter key after typing this address, the browser created a message conforming to the HTTP protocol, used DNS to obtain an IP address for www.example.org, created a TCP connection with the machine at the IP address obtained, sent the HTTP message over this TCP connection, and received back a message containing the information and displayed it in the *client area* of the browser

HTTP REQUEST AND RESPONSE

- Hyper Text Transfer protocol (HTTP) is a protocol that utilizes TCP to transfer information between computers connected on the web – this includes web servers and clients as well
- All Web communications transactions use the same protocol: the Hypertext Transfer Protocol (HTTP).
- HTTP consists of two phases:
    1. request and
    2. response.
- Each HTTP communication (request or response) between a browser and a Web server consists of two parts: a header and a body. The header contains information about the communication; the body contains the data of the communication if there is any.
- **The Request Phase**

  The general form of an HTTP request is as follows:

    1. HTTP method Domain part of the URL HTTP version
    2. Header fields
    3. Blank line
    4. Message body.
- The following is an example of the first line of an HTTP request:

  GET /storefront.html HTTP/1.1
- Commonly used HTTP request methods

1.	GET	Returns the contents of a specified document
2.	HEAD	Returns the header information for a specified document
3.	POST	Executes a specified document, using the enclosed data
4.	PUT	Replaces a specified document with the enclosed data
5.	DELETE	Deletes a specified document

Get vs Post

- ✓ The two most common **HTTP request types** (also known as **request methods**) are get and post.
- ✓ A get request typically gets (or retrieves) information from a server, such as an HTML document, an image or search results based on a user-submitted search term.
- ✓ A post request typically posts (or sends) data to a server. Common uses of post requests are to send form data or documents to a server.
- ✓ An HTTP request often posts data to a **server-side form handler** that processes the data. For example, when a user performs a search or participates in a web-based survey, the web server receives the information specified in the HTML form as part of the request.
- ✓ Get requests and post requests can both be used to send data to a web server, but each request type sends the information differently
- ✓ get request appends data to the URL, e.g., www.google.com/search?q=gita. In this case search is the name of Google's server-side form handler, q is the name of a variable in Google's search form and gita is the search term. The ? in the preceding URL separates the **query string** from the rest of the URL in a request.
- ✓ A *name/value* pair is passed to the server with the *name* and the *value* separated by an equals sign (=). If more than one *name/ value* pair is submitted, each pair is separated by an ampersand (&).
- ✓ The server uses data passed in a query string to retrieve an appropriate resource from the server. The server then sends a response to the client.
- ✓ A get request may be initiated by submitting an HTML form whose method attribute is set to "get", or by typing the URL (possibly containing a query string) directly into the browser's address bar.

- ✓ A **post** request sends form data as part of the HTTP message, not as part of the URL. A get request typically limits the query string (i.e., everything to the right of the ?) to a specific number of characters, so it's often necessary to send large amounts of information using the post method. The post method is also sometimes preferred because it hides the submitted data from the user by embedding it in an HTTP message.

- ✓ If a form submits several hidden input values along with user-submitted data, the post method might generate a URL like www.searchengine.com/search. The form data still reaches the server and is processed in a similar fashion to a get request, but the user does not see the exact information sent

- ➢ Following the first line of an HTTP communication is any number of header fields, most of which are optional. The format of a header field is the field name followed by a colon and the value of the field. There are four categories of header fields:
  - 1. *General:* For general information, such as the date
  - 2. *Request:* Included in request headers
  - 3. *Response:* For response headers
  - 4. *Entity:* Used in both request and response headers
- ➢ One common request field is the Accept field, which specifies a preference of the browser for the MIME type of the requested document. More than one Accept field can be specified if the browser is willing to accept documents in more than one format. For example, we might have any of the following:
  - Accept: text/plain
  - Accept: text/html
  - Accept: image/gif
- ➢ The header of a request must be followed by a blank line, which is used to separate the header from the body of the request. Requests that use the GET, HEAD, and DELETE methods do not have bodies. In these cases, the blank line signals the end of the request.
- ➢ A browser is not necessary to communicate with a Web server; telnet can be used instead e.g
  - telnet blanca.uccs.edu http

  This command creates a connection to the http port on the blanca.uccs .edu server.
- ➢ **The Response Phase**

  The general form of an HTTP response is as follows:
  - 1. Status line
  - 2. Response header fields
  - 3. Blank line
  - 4. Response body
- ➢ The status line includes the HTTP version used, a three-digit status code for the response, and a short textual explanation of the status code. For example, most responses begin with the following
  - HTTP/1.1 200 OK
  - HTTP/1.1 404 Not found

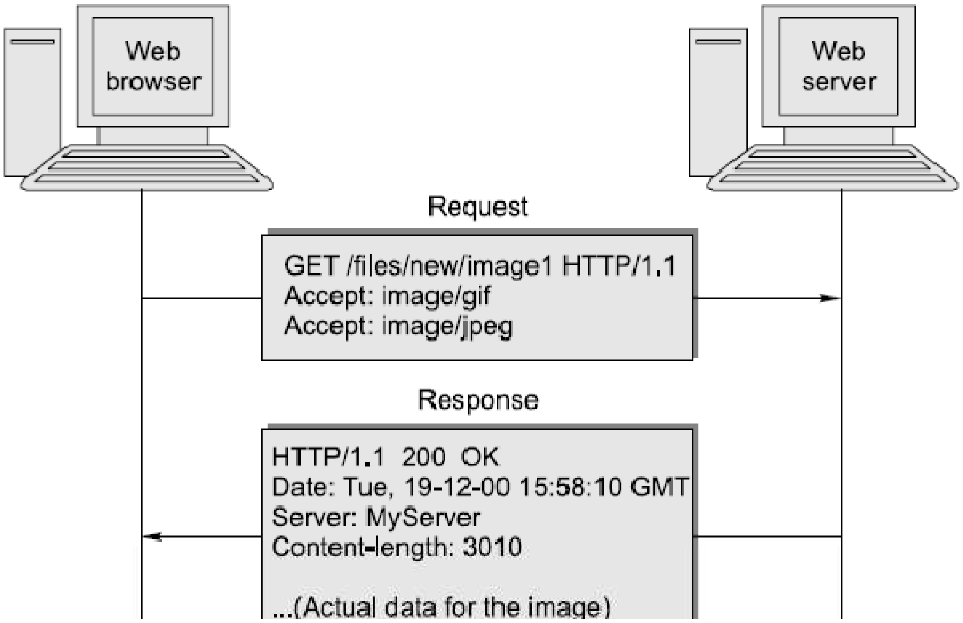
  The status codes begin with 1, 2, 3, 4, or 5.

First Digit	Category
1	Informational
2	Success
3	Redirection
4	Client error
5	Server error

- After the status line, the server sends a response header, which can contain several lines of information about the response, each in the form of a field. The only essential field of the header is Content-type.
- The following is the response header for the request

```
HTTP/1.1 200 OK
Date: Sat, 25 July 2009 22:15:11 GMT
Server: Apache/2.2.3 (CentOS)
Last-modified: Tues, 18 May 2004 16:38:38 GMT
ETag: "1b48098-16c-3dab592dc9f80"
Accept-ranges: bytes
Content-length: 364
Connection: close
Content-type: text/html, charset=UTF-8
```

- The response header must be followed by a blank line, as is the case for request headers. The response data follows the blank line. In the preceding example, the response body would be the HTML file, respond.html.



```
GET /files/new/image1 HTTP/1.1
Accept: image/gif
Accept: image/jpeg
```

```
HTTP/1.1  200  OK
Date: Tue, 19-12-00 15:58:10 GMT
Server: MyServer
Content-length: 3010

...(Actual data for the image)
```

- The browser sends a request with the GET command, as discussed.
- It also sends two more parameters by using two Accept commands. These parameters specify that the browser is capable of handling images in the GIF and JPEG format. Therefore, the server should send the image file only if it is in one of these formats.
- In response, the server sends a return code of 200 (OK).
- It also sends the information about the date and time when this response was sent back to the browser. The server's name is the same as the domain name.
- Finally, the server indicates that it is sending 3010 bytes of data (i.e., the image file is made up of bits equivalent to 3010 bytes).
- This is followed by the actual data of the image file

World Wide Web Consortium (W3C)

In October 1994, Tim Berners-Lee founded an organization—the **World Wide Web Consortium** (**W3C**)—devoted to developing nonproprietary, interoperable technologies for the World Wide Web. W3C's primary goals is to make the web universally *accessible*—regardless of disability, language or culture. Web technologies standardized by the

W3C are called **Recommendations**.

Web 1.0

- It is Basically the first version of the Web consisted of a few people creating web pages and content for a large group of readers, allowing them to access facts, information, and content from the sources.
- it was designed to help people better find information.
- This web version dealt was dedicated to users searching for data. This web version is sometimes called "the read-only Web" because it lacks the necessary forms, visuals, controls, and interactivity we enjoy on today's Internet.
- *Static pages.*
- *Content is served from the server's file system.*
- *Pages built using Server Side Includes or* **Common Gateway Interface** *(CGI).*
- *Frames and Tables are used to position and align the elements on a page.*

Features of the Web 1.0

- *Easy to connect static pages with the system via hyperlinks*
- *Supports elements like frames and tables with HTML 3.2*
- *Also has graphics and a GIF button*
- *Less interaction between the user and the server*
- *You can send* **HTML** *forms via mail*
- *Provides only a one-way publishing medium*

Web 2.0

- The term **Web 2.0** was coined by **Dale Dougherty**
- Web 2.0 is term that was introduced in 2004 and refers to the second generation of the World Wide Web.
- Web 2.0 companies use the web as a platform to create collaborative, community-based sites (e.g., social networking sites, blogs, wikis).
- Some examples of features considered to be part of Web 2.0 are Blogs , Wikis , **Social Networking , Web Application , User Participation , long tail, Rich User Experience etc**
- Websites have become much more dynamic and interconnected, producing "online communities" and making it even easier to share information on the Web.

Web 1.0 versus Web 2.0

- **Web 1.0** (the state of the web through the 1990s and early 2000s) was focused on a relatively small number of companies and advertisers producing content for users to access (some people called it the "brochure web").
- Web 2.0 *involves* the users—not only do they often create content, but they help organize it, share it, remix it, critique it, update it, etc.
- One way to look at Web 1.0 is as a *lecture*, a small number of professors informing a large audience of students.
- In comparison, Web 2.0 is a *conversation*, with everyone having the opportunity to speak and share views. Companies that understand Web 2.0 realize that their products and services are conversations as well.
- Web 2.0 embraces an **architecture of participation**—a design that encourages user interaction and community contributions. The user, are the most important aspect of Web 2.0

- For websites like Facebook®, Twitter™, YouTube, eBay® and Wikipedia ®, users create the content, while the companies provide the platforms on which to enter, manipulate and share the information.
- The architecture of participation has influenced software development as well
- Using **collective intelligence**—the concept that a large diverse group of people will create smart ideas—communities collaborate to develop software that many people believe is better and more robust than proprietary software.
- Rich Internet Applications (RIAs) are being developed using technologies such as Ajax.
- Search engines, including Google™, Microsoft Bing™, and many more, have become essential to sifting through the massive amount of content on the web.
- Social bookmarking sites allow users to share their favorite sites with others
- In the future, computers will learn to understand the meaning of the data on the web—the beginnings of the **Semantic Web** are already appearing

Web 3.0

- Web 3.0 is the "read, write, execute Web."
- It's a semantic web, where the web technology evolves into a tool that lets users create, share, and connect content via search and analysis. It is based on comprehension of words instead of numbers and keywords.
- It incorporates Artificial Intelligence and Machine Learning. If these concepts are combined with Natural Language Processing (NLP), the result is a computer that uses Web 3.0 to become smarter and more responsive to user needs.
- It presents the connectivity of multiple devices and applications through the Internet of Things (IoT). Semantic metadata makes this process possible, allowing all available information to be effectively leveraged. In addition, people can connect to the Internet anytime, anywhere, without needing a computer or smart device.
- It offers users the freedom to interact publicly or privately without having an intermediary expose them to risks, therefore offering people “trustless” data.
- It uses 3-D graphics. In fact, we already see this in computer games, virtual tours, and e-commerce.
- It facilitates participation without needing authorization from a governing body. It’s permissionless.
- It can be used for:
  - Metaverses: A 3D-rendered, boundless, virtual world
  - Blockchain games: They allow users to have actual ownership of in-game resources, following the principles of NFTs
  - Privacy and digital infrastructure: This use includes zero-knowledge proofs and more secure personal information
  - Decentralized finance. This use includes payment Blockchains, peer-to-peer digital financial transactions, smart contracts, and cryptocurrency
  - Decentralized autonomous organizations. Community members own online communities
- Web 3.0 ultimately lets users interact, exchange information, and securely conduct financial transactions without a centralized authority or coordinator. As a result, each user becomes a content owner instead of just a content user.
- Web 3.0 isn't entirely in place. However, we are already seeing elements of Web 3.0 working their way into our Internet experiences, such as NFTs, Blockchain, Distributed ledgers, and the AR cloud

S. No.	Web 1.0	Web 2.0	Web 3.0
1	Mostly Read-Only	Wildly Read-Write	Portable and Personal
2	Company Focus	Community Focus	Individual Focus
3	Home Pages	Blogs / Wikis	Live-streams / Waves
4	Owning Content	Sharing Content	Consolidating Content
5	WebForms	Web Applications	Smart Applications
6	Directories	Tagging	User behavior
7	Page Views	Cost Per Click	User Engagement
8	Banner Advertising	Interactive Advertising	Behavioral Advertising
9	Britannica Online	Wikipedia	The Semantic Web
10	HTML/Portals	XML / RSS	RDF / RDFS / OWL
11	Data was not Focused.	Data of many was controlled by some	Data was personalized and no use of mediatory.
12	Information sharing is the goal.	Interaction is the goal.	Immersion is the goal.
13	It connects information as its primary goal.	It aims to connect people.	Focuses on relating knowledge.
14	Static websites	Introduction of web applications	Intelligent web-based functions and apps
15	A simpler, more passive web.	An enhanced social Web	A semantic web exists.
16	Web and File Servers, HTML, and Portals are technologies connected to Web 1.0.	AJAX, JavaScript, CSS, and HTML5 are examples of related technology.	Web 3.0 technologies include blockchain, artificial intelligence, and decentralized protocols.
17	**Associated Technologies** • Web and File Servers • Search Engines (including AltaVista and Yahoo!) • E-mail accounts (Yahoo!, Hotmail) • Peer-to-Peer File Sharing (Napster, BitTorrent) and others.	**Associated Technologies** • Frameworks for Ajax and JavaScript • Microsoft.NET • Blogs • Wikis and others.	**Associated Technologies** • Searching Using Semantics • Databases of Information • Ontologies • Intelligent Digital Personal Assistants and others.

Web Design Issues

Browser & Operating Systems

- Web pages are written using different HTML tags and viewed in browser window.
- The different browsers and their versions greatly affect the way a page is rendered, as different browsers sometimes interpret same HTML tag in a different way.
- Different versions of HTML also support different sets of tags.
- The support for different tags also varies across the different browsers and their versions.
- Same browser may work slightly different on different operating system and hardware platform.
- To make a web page portable, test it on different browsers on different operating systems.

Bandwidth and Cache

- Users have different connection speed, i.e. bandwidth, to access the Web sites.
- Connection speed plays an important role in designing web pages, if user has low bandwidth connection and a web page contains too many images, it takes more time to download.
- Generally, users have no patience to wait for longer time than 10-15 seconds and move to other site without looking at contents of your web page.
- Browser provides temporary memory called cache to store the graphics.
- When user gives the URL of the web page for the first time, HTML file together with all the graphics files referred in a page is downloaded and displayed.

Display Resolution

- Display resolution is another important factor affecting the Web page design, as we do not have any control on display resolution of the monitors on which user views our pages.
- Display or screen resolution is measured in terms of pixels and common resolutions are 800 X 600 and 1024 X 786.
- We have three choices for Web page design.
    - Design a web page with fixed resolution.
    - Make a flexible design using HTML table to fit into different resolution.
    - If the page is displayed on a monitor with a higher resolution, the page is displayed on left- hand side and some part on the right-hand side remains blank. We can use centered design to display page properly.

Look & Feel

- Look and feel of the website decides the overall appearance of the website.
- It includes all the design aspects such as
    - Web site theme
    - Web typography
    - Graphics
    - Visual structure
    - Navigation etc…

Page Layout and Linking

- Website contains of individual web pages that are linked together using various navigational links.
- Page layout defines the visual structure of the page and divides the page area into different parts to present the information of varying importance.
- Page layout allows the designer to distribute the contents on a page such that visitor can view it easily and find necessary details.

Locating Information

- Webpage is viewed on a computer screen and the screen can be divided into five major areas such as center, top, right, bottom and left in this particular order.
- The first major area of importance in terms of users viewing pattern is the center, then top, right, bottom and left in this particular order.

Making Design user-Centric

- It is very difficult for any Web designer to predict the exact behavior of the Web site users.
- However, idea of general behavior of common user helps in making design of the Web site user-centric.
- Users either scan the information on the web page to find the section of their interest or read the information to get details.

Sitemap

- Many a times Web sites are too complex as there are a large number of sections and each section contains many pages.
- It becomes difficult for visitors to quickly move from one part to other.
- Once the user selects a particular section and pages in that section, user gets confused about where he/she is and where to go from there.
- To make it simple, keep your hierarchy of information to few levels or provide the navigation bar on each page to jump directly to a particular section.

Tips for Effective Navigation.

- Navigation links are either text based, i.e. a word or a phrase is used as a link, or graphical, i.e. a image, i.e. a icon or a logo is used as a link.
- Navigation links should be clear and meaningful.
- It should be consistent.
- Link should be understandable.
- Organize the links such that contents are grouped logically.
- Provide search link, if necessary, usually on top of the page. Use common links such as ‘about us’ or ‘Contact us’.
- Provide the way to return to first page.
- Provide the user with information regarding location
- Horizontal navigation bar can be provided on each page to directly jump to any section