WEB TECHNOLOGIES

PREPARED BY,

Mrs.C.RADHA,

ASSOCIATE PROFESSOR/MCA,

MEC.

19CAC02/ WEB TECHNOLOGIES

Objectives:

- To understand the concepts and architecture of the World Wide Web.
- To understand the basic network concepts.
- To understand and practice mark up languages
- To understand and practice embedded dynamic scripting on client side Internet Programming
- To understand and practice web development techniques on client-side

UNIT-I INTRODUCTION TO WWW

Internet Standards – Introduction to WWW – WWW Architecture – FTP – Overview of HTTP – HTTP request, response – Generation of dynamic web pages.

UNIT-II BASIC CONCEPTS OF NETWORKS & WEB

TCP & UDP Protocols - OSI Reference Model – TCP / IP Protocol Architecture – URI – MIME – Common Gateway Interface – Remote Login – PERL – SGML.

UNIT-III USER INTERFACE DESIGN

Mark up Language (HTML): Introduction to HTML & HTML 5 – Headings, Paragraphs line breaks etc – Frames – Tables – Lists – images.

Cascading Style sheets – The need for CSS – Introduction to CSS – Internal & External style sheets– Inline style sheets – Web Browser Architecture.

UNIT-IV INTRODUCTION TO JAVASCRIPT

Introduction - Core features - Data types and Variables - Operators, Expressions, and Conditional Statements - Functions - Objects - Document Object Model – Loops – Java Script and Form Processing.

UNIT-V AJAX & PHP

AJAX**:** Introduction – Working of AJAX – AJAX Coding.

PHP : Introduction – How web works - Setting up the environment (LAMP server) - Programming basics - Control structures and looping structures – Functions – Reading Data in Web Pages - Embedding PHP within HTML - Establishing connectivity with MySQL database.

Outcomes:

- Acquire knowledge about functionalities of world wide web
- Explore markup languages features and create interactive web pages using them
- Learn and design Client side validation using scripting languages
- Acquire knowledge about Open source JavaScript libraries
- Able to design front end web page and connect to the back end databases.

TEXT BOOKS :

1. Harvey & Paul Deitel & Associates, Harvey Deitel and Abbey Deitel, “Internet and World Wide Web - How To Program”, Fifth Edition, Pearson Education, 2011.

2. Achyut S Godbole and Atul Kahate, “Web Technologies”, Second Edition, Tata McGraw Hill, 2012.

3. Steven Holzner, “The Complete Reference - PHP”, Tata McGraw Hill, 2008.

REFERENCES:

1.Thomas A Powell, Fritz Schneider, “JavaScript: The Complete Reference”, Third Edition, Tata McGraw Hill, 2013.

2. David Flanagan, “JavaScript: The Definitive Guide, Sixth Edition”, O'Reilly Media, 2011.

3. Mike Mcgrath, “PHP & MySQL in easy Steps”, Tata McGraw Hill, 2012.

What is Web Technology?

- Web technology refers to the means by which computers communicate with each other using markup languages and multimedia software packages.
- It gives us a way to interact with hosted information, such as websites.
- It involves the use of Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS).
- The three basic languages that make up the World Wide Web are HTML, CSS, and JavaScript.

What are Web Technologies?

- Web technologies are the various tools and techniques that are utilised in the process of communication between different types of devices over the internet.
- Let's break it down into two pieces: 'web' and 'technology'.
- The web, in this case, refers to the World Wide Web, more commonly known as WWW. It first came into being in 1989 when famous scientist and engineer, Tim Berners-Lee, came up with an efficient mechanism to share resources between scientists all over the world.

What Are The Different Types Of Web Technologies?

- The basics, which will cover web browsers and some web app development fundamentals
- Programming languages and frameworks which are used in the development of websites
- Databases that are used at the backend to store data required or collected by websites
- Some protocols, that is, rules for communicating on the web
- Graphic, audiovisual, and other multimedia elements
- Some data formats that are usually used to transmit data over the internet
- Other miscellaneous web technologies

UNIT-I
INTRODUCTION TO WWW

INTERNET STANDARDS

ABOUT INTERNET

- A network is a group of connected, communicating devices such as computers and printers.
- An internet is two or more networks that can communicate with each other(composed of hundreds of thousands of interconnected networks).
- It is the largest network in the world that connects hundreds of thousands of individual networks all over the world.
- Private individuals as well as various organizations such as government agencies, schools, research facilities, corporations, and libraries in more than 100 countries use the Internet.

History of Internet

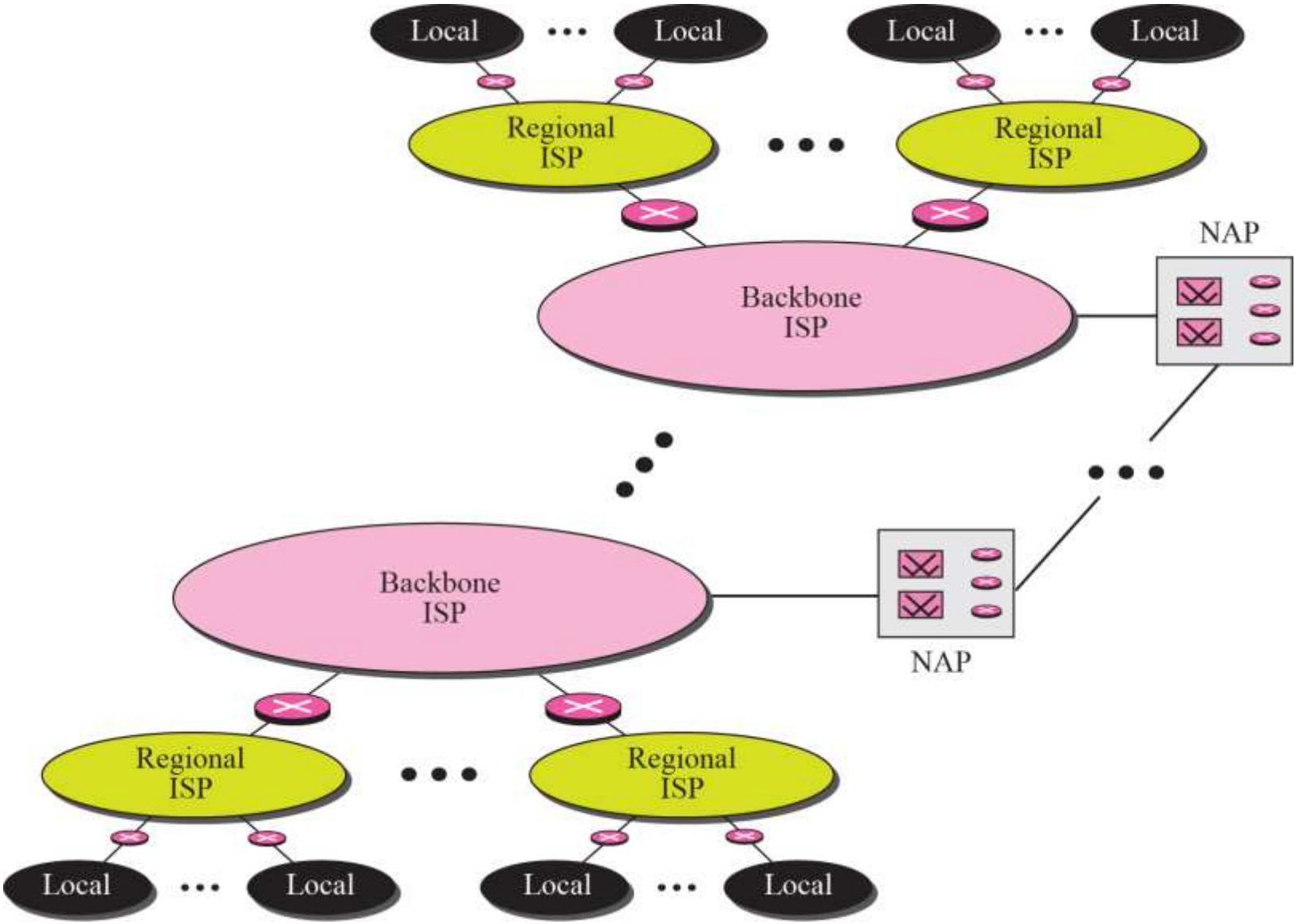
Early Networking

- Advanced Research Projects Agency
  - Created ARPAnet
    - Grandparent of Internet
    - Used packet switching to rout data
    - TCP/IP
      - Protocols to ensure Internet reliability and identification

Uses of the Internet

- Send e-mail messages.
- Send (upload) or receive (down load) files between computers.
- Participate in discussion groups, such as mailing lists and newsgroups.
- Surfing the web.

Internet Today



Protocols and Standards

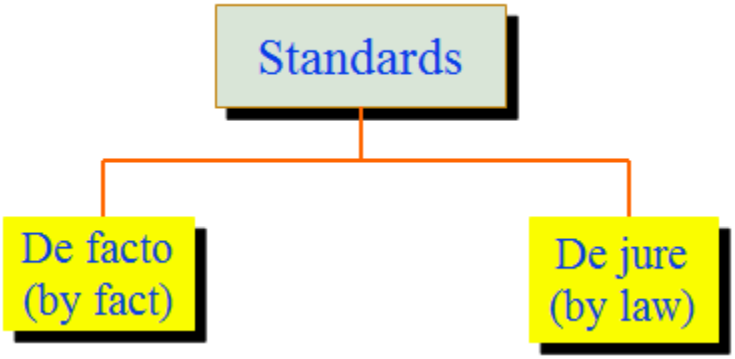
- Communication in computer networks
  - ~ occurs between entities in different systems
- Entity
  - ~ is anything capable of sending or receiving information
- Protocol
  - ~ is a set of rules that govern data communication

- Principle Elements
  - Syntax
    - ˜ refers to the structure or format of the data, meaning the order in which they are presented.
  - Semantics
    - ˜ refers to the meaning of each section of bits.
  - Timing
    - ˜ refers to two characteristics (when data should be sent and how fast it can be sent)

Standards

Standards

- are essential in creating and maintaining an open and competitive market for equipment manufactures and in guaranteeing national and international interoperability of data and communications technology and processes.
- Standards are developed through the cooperation of standards creation committees, forums, and government regulatory agencies.



Standards Organizations

- Standard Creation Committees
  - ISO (The International Standard Organization)
    - ~ created in 1947
    - ~ voluntary organization
    - ~ is an organization dedicated to worldwide agreement on international standards in a variety of fields (scientific, technological, economic activity)

- ITU-T(International Telecommunications Union - Telecommunications Standards Sector)
  - ~ is an international standards organization related to the United Nations that develops standards for telecommunications.
  - Two popular standards developed by ITU-T are the V series (data transmission over phone lines) and the X series (transmission over public digital networks)

- ANSI (American National Standard Institute)
  - ~ is a nonprofit organization and is the U.S. voting representative to be both the ISO and the ITU-T
- IEEE (Institute of Electrical and Electronic Engineers)
  - ~ is the largest national professional group involved in developing standards for computing, communication, electrical engineering, and electronics)
- EIA (Electronic Industries Association)
  - ~ is an association of electronics manufactures in the United States. (EIA-232-D, EIA-530 standards)

- Forums
  - ~ consist of representatives from corporation that test, evaluate and standardize new technologies.
  - Frame Relay Forum
  - ATM Forum and ATM consortium

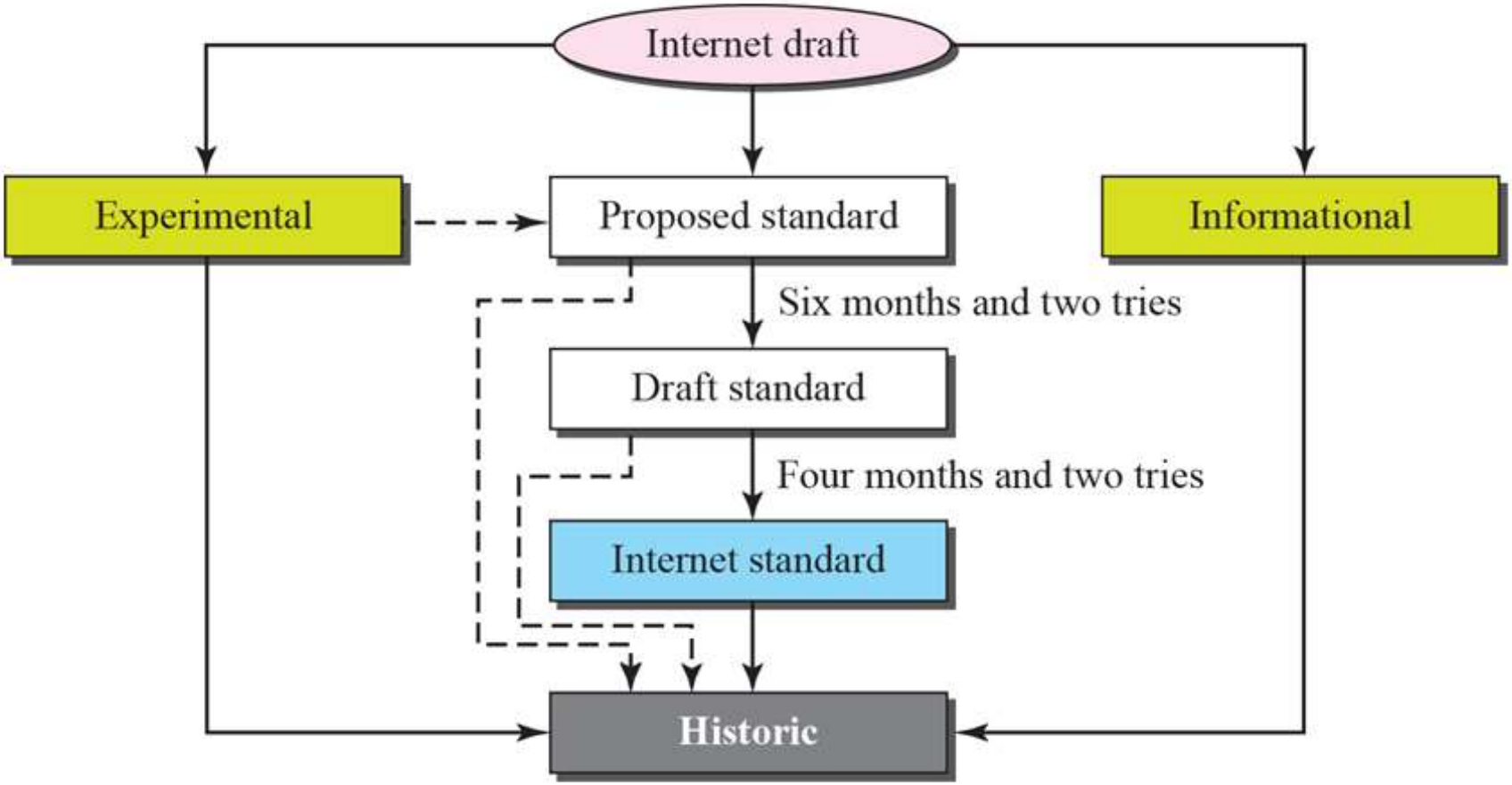
- Regulatory agencies
  - FCC (Federal Communications Commission)
  - KCC (Korea Communications Commission) &

    MSIP (Ministry of Science, ICT and Future Planning)

Internet Standards

- Internet standard
  - is a thoroughly tested specification that is useful to and adhered by those who work with the Internet
  - It is a formalized regulation that must be followed.
  - There is a strict procedure by which a specification attains Internet standard status.
- A specification begins as an Internet draft
  - Working document with no official status and six-month life-time
- RFC (Request for Comment)
  - Recommendation from Internet authorities

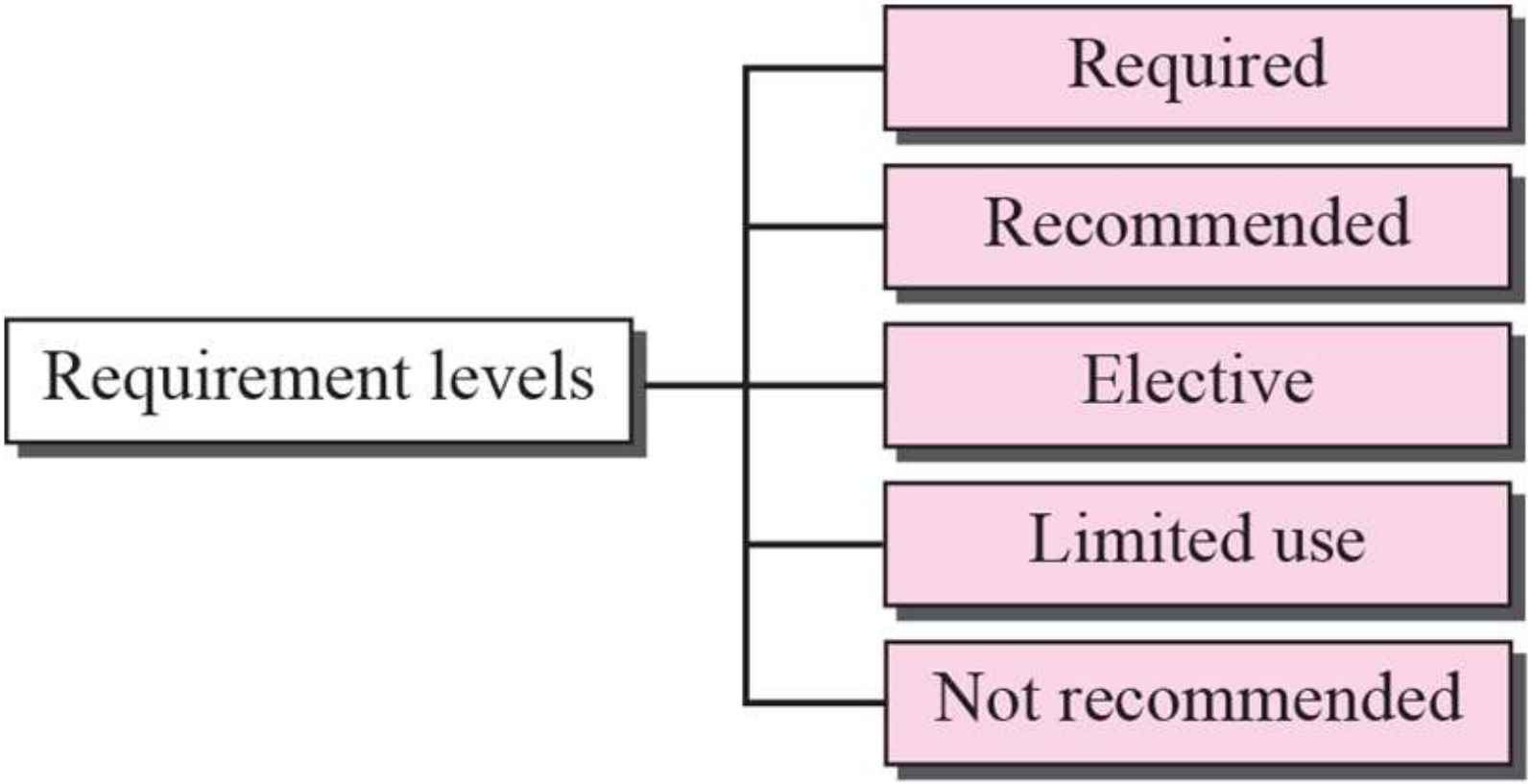
Maturity Levels



- Proposed Standard
  - Is a specification that is stable, well understood, and of sufficient interest to the Internet community.
- Draft Standard
  - Proposed is elevated to draft standard status after at least two successful independent and interoperable implementations
  - With modification, normally becomes an Internet standard
- Internet Standard
  - A draft standard reaches Internet standard status after demonstrations of successful Implementation

- Historic RFCs
  - Superseded by later specifications or never passed the necessary maturity levels to become an Internet standard
- Experimental RFCs
  - Not implemented in any functional Internet service
- Informational RFCs
  - Containing general, historical, or tutorial information related to the Internet

Requirement Levels



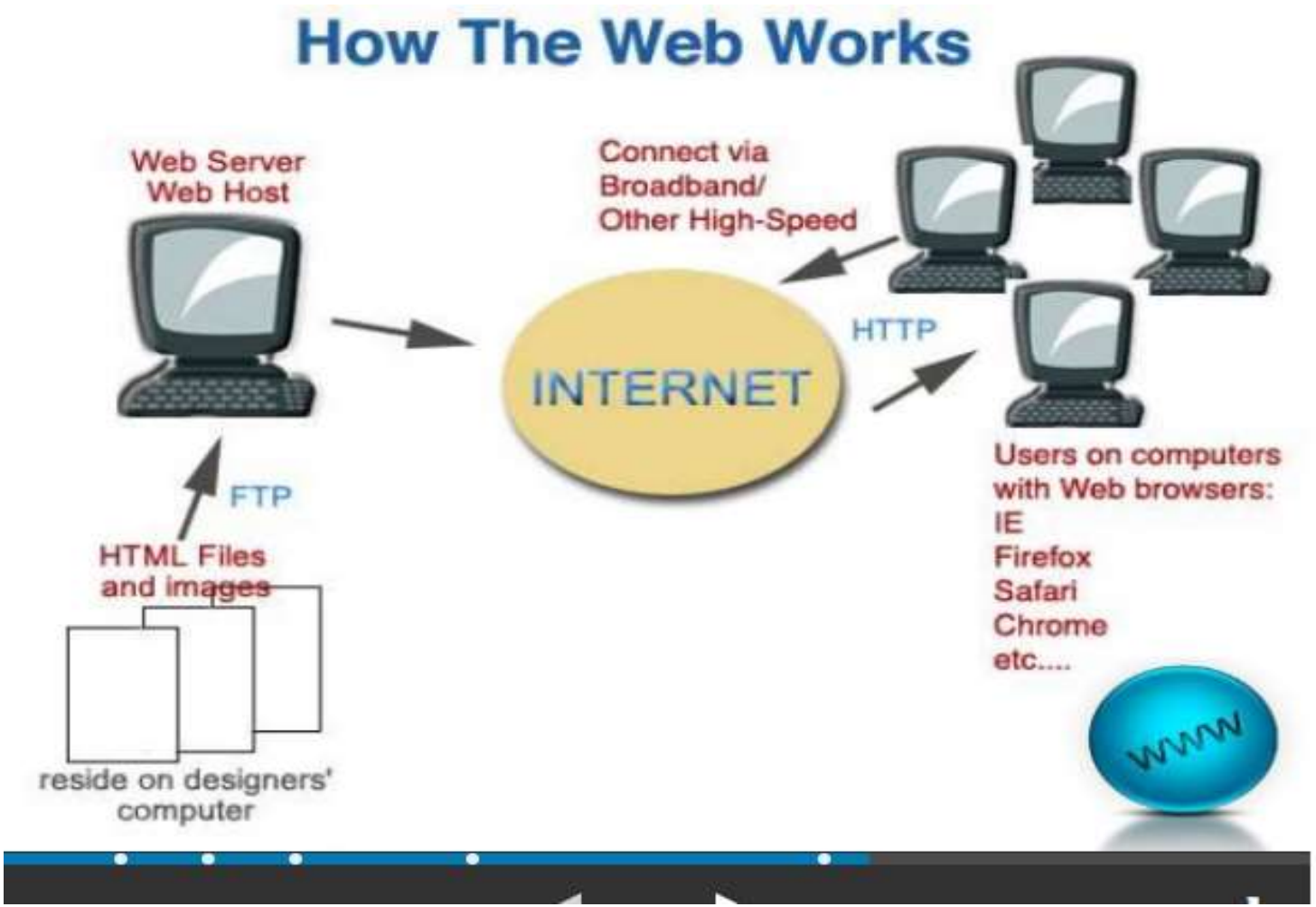
- Required protocols (RFCs)
  - Must be implemented by all Internet systems to achieve minimum conformance
  - For example, IP and ICMP
- Recommended protocols
  - Not required for minimum conformance; it is recommended because of its usefulness
  - For example, FTP and Telnet
- Elective protocols
  - Not required and not recommended; but a system can use it to its own benefit.

- Limited Use protocols
  - Used only in limited situations; most of the experimental RFCs fall under this category
- Not Recommended
  - Inappropriate for general use; normally a historic (obsolete) RFC may fall under this category

Introduction to WWW

- WWW", "Web" or "W3"
- The World Wide Web is a way of exchanging information between computers on the Internet.
- It is the network of pages of images, texts and sounds on the Internet which can be viewed using browser software .
- It is a way of accessing the information over the medium of the internet.
- The World Wide Web, or Web, consists of a worldwide collection of electronic documents (Web pages)
- A Web site is a collection of related Web pages and associated items.
- Simple and easy to use

- A Web server is a computer that delivers requested Web pages to your computer
- Web 2.0 refers to Web sites that provide a means for users to interact
- 1989 – 1990 Tim Berners Lee invents the World Wide Web (www) at Cern .
- Published in 1992 - Means for transferring text and graphics simultaneously
- The World Wide Web allows computer users to locate and view multimedia-based documents (i.e., documents with text, graphics, animations, audios or videos) on almost any subject.


How The Web Works
Web Server
Web Host
Connect via
Broadband/
Other High-Speed
HTTP
INTERNET
FTP
HTML Files
and images
reside on designers'
computer
Users on computers
with Web browsers:
IE
Firefox
Safari
Chrome
etc....
WWW

Structural Components

- Clients/browsers – to dominant implementations
- Servers – run on sophisticated hardware
- Caches – many interesting implementations
- Internet – the global infrastructure which facilitates data transfer

Semantic Components

- Hyper Text Transfer Protocol (HTTP)
- Hyper Text Markup Language (HTML)
- Extensible Markup Language (XML)
- Uniform Resource Identifiers (URIs)

Hypertext & Hypermedia

- **Hypertext** - Text which contains links to other texts. The term was coined by Ted Nelson around 1965 .
- **Hypermedia** - Term used for hypertext which is not constrained to be text: it can include graphics, video and sound

Web browser – It displays a web document and enables users to access web documents.

Web Server – A program that waits patiently for the browser to request a web page. The servers looks for the requested information, retrieves it and send it to the browser or sends an error message if the file is not found.

Uniform Resource Locator (URL) - These are the web addresses. The resource locator is an addressing system .

- Two basic steps required in making a web page.
  - Create an HTML File
  - Upload file to server

- Web pages are generally sets of text files coded with special notation called Hypertext Markup Language (HTML) that is interpreted by a web browser program reading the file.

Internet & WWW

- The internet came first.
- Internet and WWW are two different concepts.
- **Internet**- It is a global computer network made up of thousand of computers & computer network.
- **WWW**- It is a system that interlinked via Internet.

Difference between Internet and WWW

1) **Internet**- is a global system of interconnected computer network. This computers communicate with each other using a common protocol (TCP/IP i.e. Transmission Control Protocol / Internet protocol)

World Wide Web (WWW) : It is one of the services that run on internet. WWW is a system of interlinked hypertext documents accessed via Internet. Way of accessing the information .

2) The Internet and the Web are two separate but related things.

Internet – It is a massive network of networks. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the Internet.

WWW- The World Wide Web, or simply Web, is a way of accessing information over the medium of the Internet. So the Web is just a portion of the Internet.

Web Browsers

Eg:

- Internet Explorer
- Firefox
- Opera
- Safari
- Google Chrome

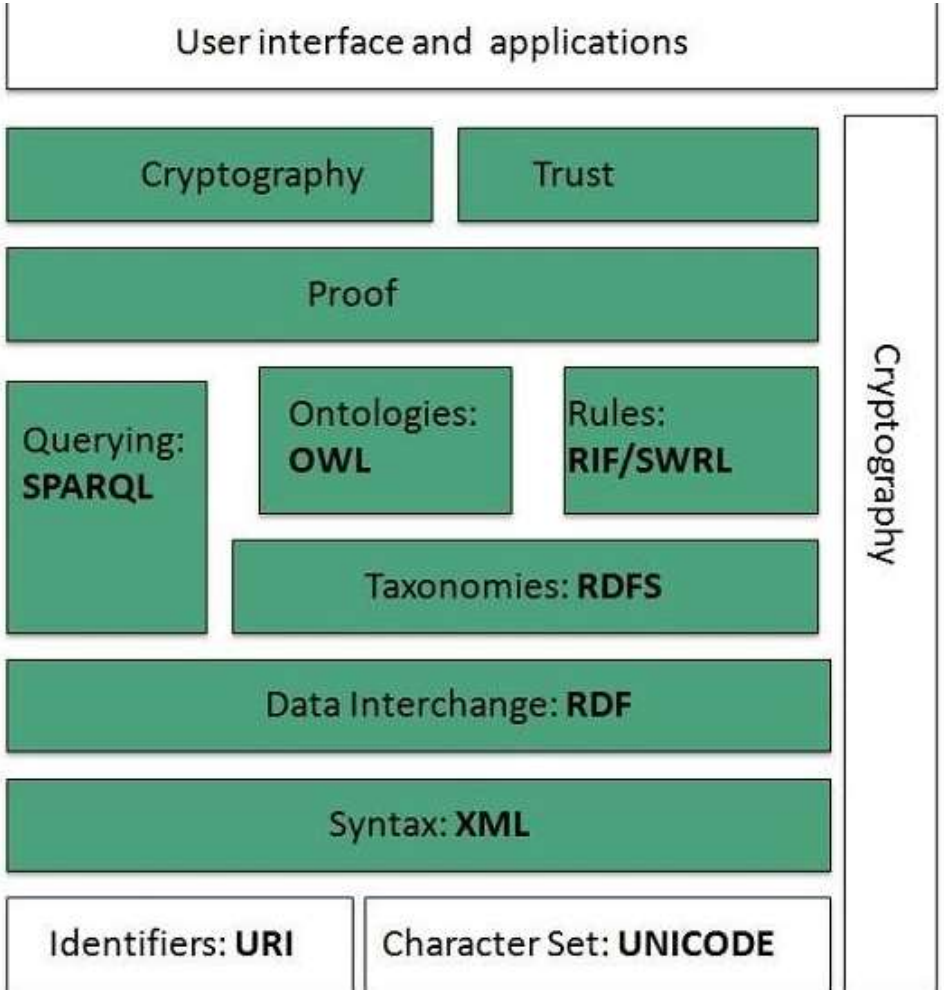
- A home page is the first page that a Web site displays
- Web pages provide links to other related Web pages
- Downloading is the process of receiving information

Search engine

- Finds information related to a specific topic
- Classifies Web pages in an organized set of categories- Images, Videos, Audio, News, Maps, Blogs, People or Businesses.

- Web publishing - It is the development and maintenance of Web pages
- WWW is a way of exchanging information. WWW is constructed using HTML. WWW is one of the services that run on internet.

WWW ARCHITECTURE



Identifiers and Character Set

- **Uniform Resource Identifier (URI)** is used to uniquely identify resources on the web and **UNICODE** makes it possible to built web pages that can be read and write in human languages.

Syntax

- **XML (Extensible Markup Language)** helps to define common syntax in semantic web.

Data Interchange

- **Resource Description Framework (RDF)** framework helps in defining core representation of data for web. RDF represents data about resource in graph form.

Taxonomies

- **RDF Schema (RDFS)** allows more standardized description of **taxonomies** and other **ontological** constructs.

Ontologies

- **Web Ontology Language (OWL)** offers more constructs over RDFS. It comes in following three versions:

  OWL Lite for taxonomies and simple constraints.

  OWL DL for full description logic support.

  OWL for more syntactic freedom of RDF

Rules

- **RIF** and **SWRL** offers rules beyond the constructs that are available from **RDFs** and **OWL.**
- Simple Protocol and **RDF Query Language (SPARQL)** is SQL like language used for querying RDF data and OWL Ontologies.

Proof

- All semantic and rules that are executed at layers below Proof and their result will be used to prove deductions.

Cryptography

- **Cryptography** means such as digital signature for verification of the origin of sources is used.

User Interface and Applications

- On the top of layer **User interface and Applications** layer is built for user interaction.

FTP (File Transfer Protocol)

FTP

- FTP stands for File transfer protocol.
- FTP is a standard internet protocol provided by TCP/IP used for transmitting the files from one host to another.
- It is mainly used for transferring the web page files from their creator to the computer that acts as a server for other computers on the internet.
- It is also used for downloading the files to computer from other servers.

Objectives of FTP

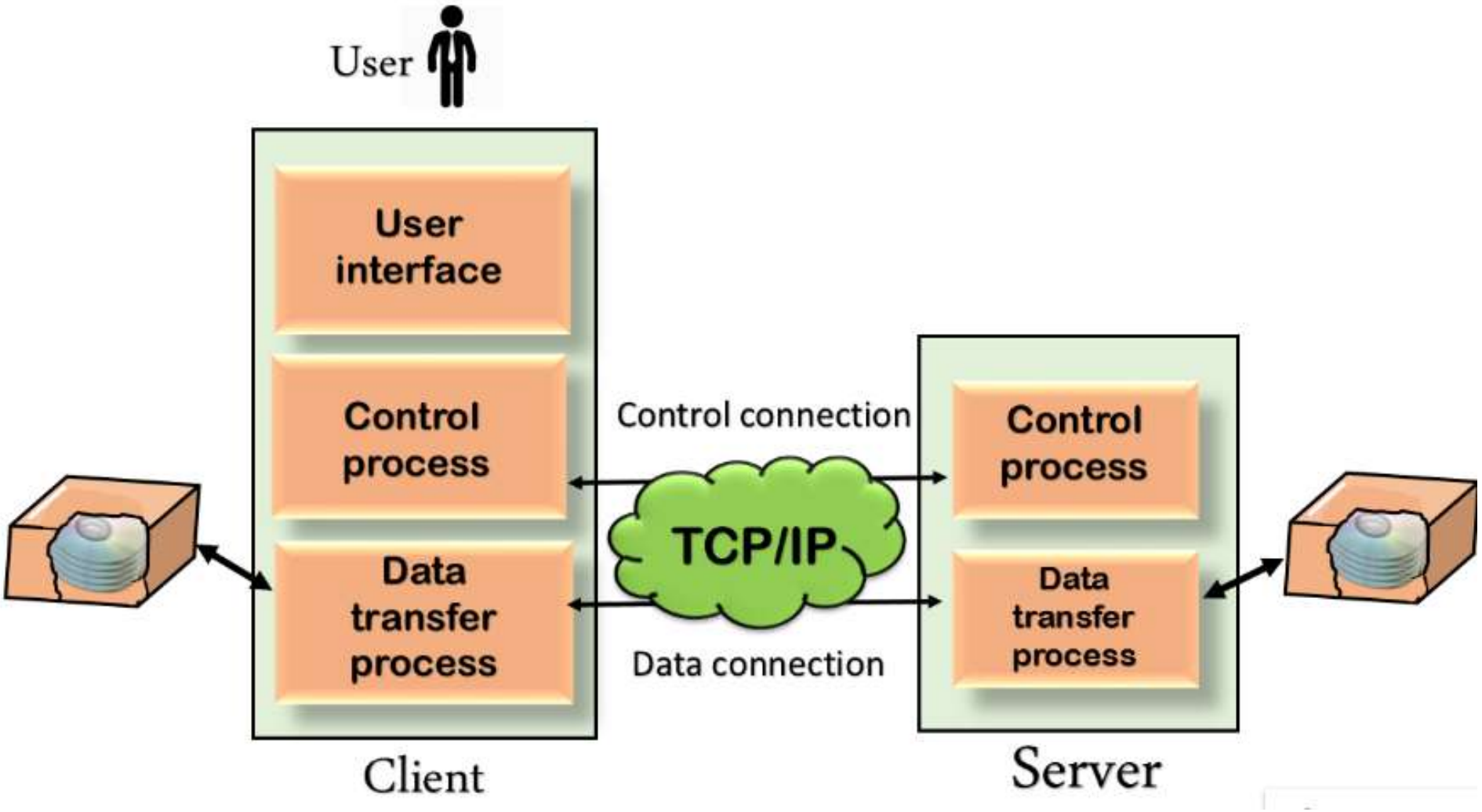
- It provides the sharing of files.
- It is used to encourage the use of remote computers.
- It transfers the data more reliably and efficiently.

Why FTP?

- Transferring files from one system to another is simple but sometimes it can cause problems.

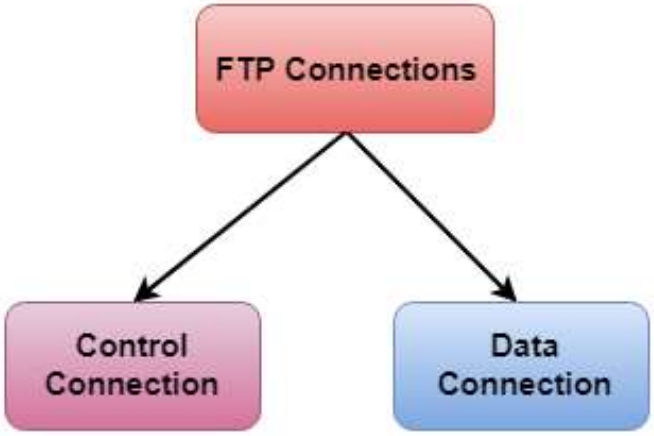
- For example, two systems may have different file conventions, different ways to represent text and data, different directory structures.
- FTP protocol overcomes these problems by establishing two connections between hosts.
- One connection is used for data transfer, and another connection is used for the control connection.

Mechanism of FTP



- The above figure shows the basic model of the FTP.
- The FTP client has three components: the user interface, control process, and data transfer process.
- The server has two components: the server control process and the server data transfer process.

There are two types of connections in FTP:



Control Connection:

- It uses very simple rules for communication.
- Through this connection, we can transfer a line of command or line of response at a time.
- The control connection is made between the control processes.
- It remains connected during the entire interactive FTP session.

Data Connection:

- It uses very complex rules as data types may vary.
- The data connection is made between data transfer processes.
- It opens when a command comes for transferring the files and closes when the file is transferred.

FTP Clients

- FTP client is a program that implements a file transfer protocol which allows you to transfer files between two hosts on the internet.
- It allows a user to connect to a remote host and upload or download the files.
- It has a set of commands that we can use to connect to a host, transfer the files between you and your host and close the connection.
- The FTP program is also available as a built-in component in a Web browser.
- This GUI based FTP client makes the file transfer very easy and also does not require to remember the FTP commands.

Advantages of FTP

- **Speed:** The FTP is one of the fastest way to transfer the files from one computer to another computer.
- **Efficient:** It is more efficient as we do not need to complete all the operations to get the entire file.
- **Security:** To access the FTP server, we need to login with the username and password.
- **Back & forth movement:** FTP allows us to transfer the files back and forth. Suppose you are a manager of the company, you send some information to all the employees, and they all send information back on the same server.

Disadvantages of FTP

- The standard requirement of the industry is that all the FTP transmissions should be encrypted. However, not all the FTP providers are equal and not all the providers offer encryption. So, we will have to look out for the FTP providers that provides encryption.
- FTP serves two operations, i.e., to send and receive large files on a network. However, the size limit of the file is 2GB that can be sent. It also doesn't allow you to run simultaneous transfers to multiple receivers.
- Passwords and file contents are sent in clear text that allows unwanted eavesdropping. So, it is quite possible that attackers can carry out the brute force attack by trying to guess the FTP password.
- It is not compatible with every system.

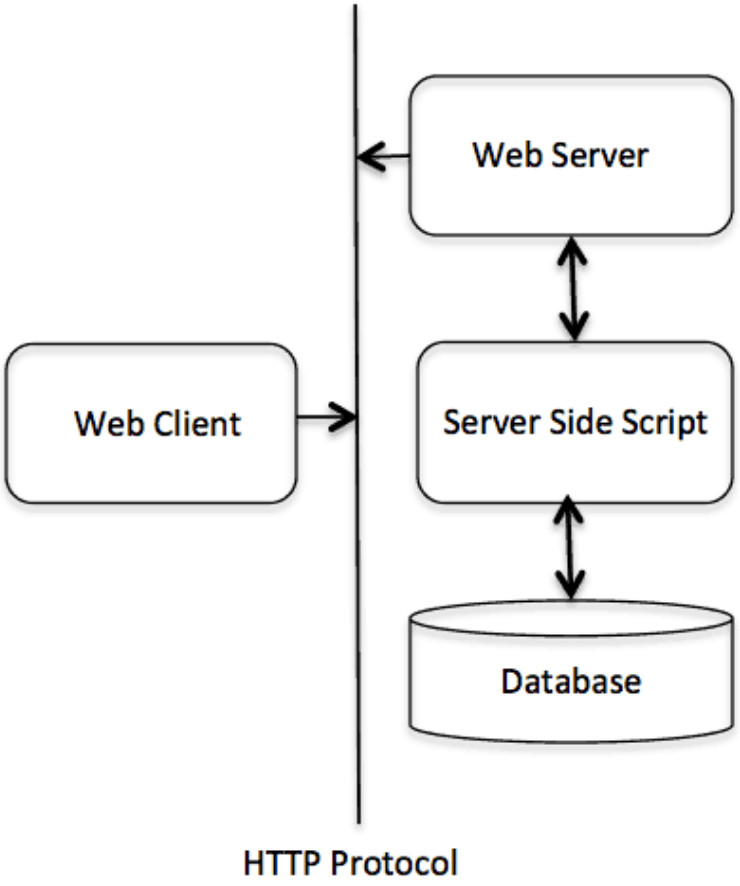
HTTP(HyperText Transfer Protocol)

- It is a protocol used to access the data on the World Wide Web (www).
- It can be used to transfer the data in the form of plain text, hypertext, audio, video, and so on.
- This protocol is known as HyperText Transfer Protocol because of its efficiency that allows us to use in a hypertext environment where there are rapid jumps from one document to another document.
- HTTP is similar to the FTP as it also transfers the files from one host to another host. But, HTTP is simpler than FTP as HTTP uses only one connection, i.e., no control connection to transfer the files.
- HTTP is used to carry the data in the form of MIME-like format.

Features of HTTP

- **Connectionless protocol:** It is a connectionless approach in which HTTP client i.e., a browser initiates the HTTP request and after the request is sent the client disconnects from server and waits for the response.
- **Media Independent:** It specifies that any type of media content can be sent by HTTP as long as both the server and the client can handle the data content.
- **Stateless:** It is a stateless protocol as both the client and server know each other only during the current request. Due to this nature, both the client and server do not retain the information between various requests of the web pages.

Basic Architecture of HTTP



- The diagram represents the basic architecture of web application and depicts where HTTP stands.
- HTTP is request/response protocol which is based on client/server based architecture. In this protocol, web browser, search engines, etc. behave as HTTP clients and the Web server like Servlet behaves as a server

Client

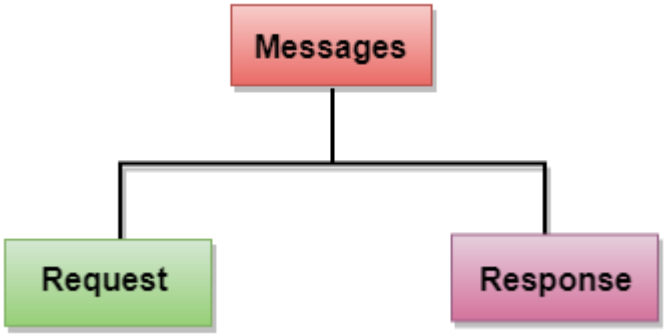
- The HTTP client sends a request to the server in the form of a request method, URI, and protocol version, followed by a MIME-like message containing request modifiers, client information, and possible body content over a TCP/IP connection.

Server

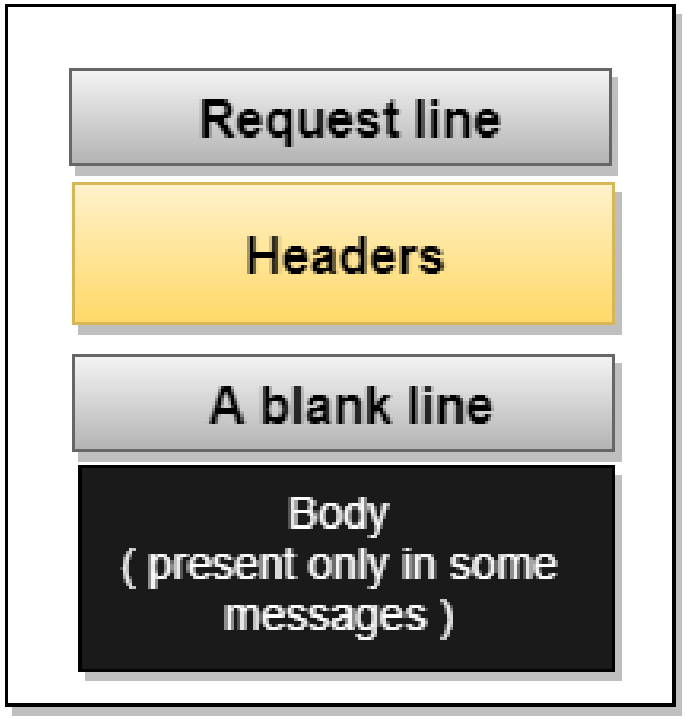
- The HTTP server responds with a status line, including the message's protocol version and a success or error code, followed by a MIME-like message containing server information, entity meta information, and possible entity-body content.

Messages

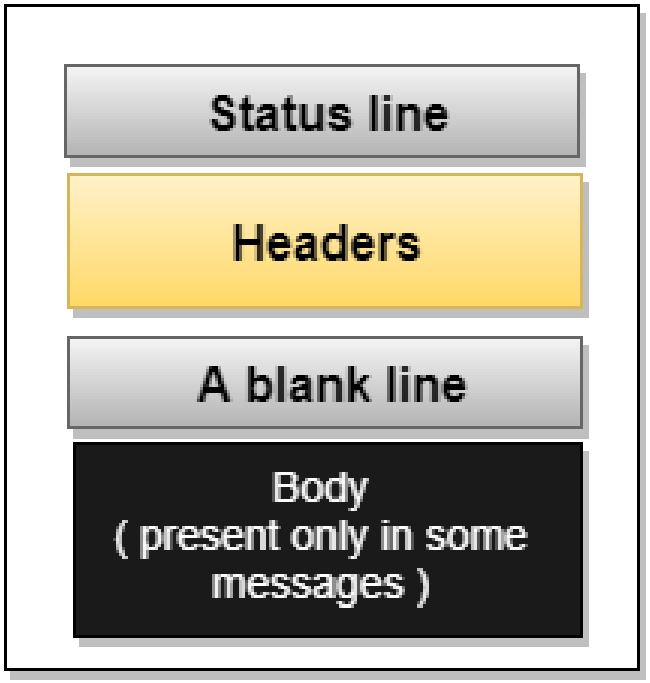
- HTTP messages are of two types: request and response. Both the message types follow the same message format.



- **Request Message:** The request message is sent by the client that consists of a request line, headers, and sometimes a body.


Request line
Headers
A blank line
Body
(present only in some
messages)

- **Response Message:** The response message is sent by the server to the client that consists of a status line, headers, and sometimes a body.



Uniform Resource Locator (URL)

- A client that wants to access the document in an internet needs an address and to facilitate the access of documents, the HTTP uses the concept of Uniform Resource Locator (URL).
- The Uniform Resource Locator (URL) is a standard way of specifying any kind of information on the internet.
- The URL defines four parts: method, host computer, port, and path.

URL

Uniform Resource Locator

- **Method:** The method is the protocol used to retrieve the document from a server. For example, HTTP.
- **Host:** The host is the computer where the information is stored, and the computer is given an alias name. Web pages are mainly stored in the computers and the computers are given an alias name that begins with the characters "www". This field is not mandatory.
- **Port:** The URL can also contain the port number of the server, but it's an optional field. If the port number is included, then it must come between the host and path and it should be separated from the host by a colon.
- **Path:** Path is the pathname of the file where the information is stored. The path itself contain slashes that separate the directories from the subdirectories and files.

HTTP Message

- HTTP Message is used to show how data is exchanged between the client and the server.
- An **HTTP** client is a program that establishes a connection to a server to send one or more HTTP request messages.
- An **HTTP server** is a program that accepts connections to serve HTTP requests by sending an HTTP response messages.

Message Type

- HTTP message consists of an initial request line and an initial response line.

Format:

HTTP-message = Request | Response ; HTTP/1.1 messages

1) Initial Request Line

- The initial line is different for the request and for the response. A request-line consists of three parts: a **method name**, **requested resource's local path**, and the **HTTP version** being used. All these parts are separated by spaces.

Syntax:

- GET /path/to/file/index.html HTTP/1.0

Here,

- **GET** is the most common HTTP method.
- The **path** shows the part of the URL after the host name. It is also called a request URI.
- The **version** of HTTP always takes the form "**HTTP/x.x**", uppercase.

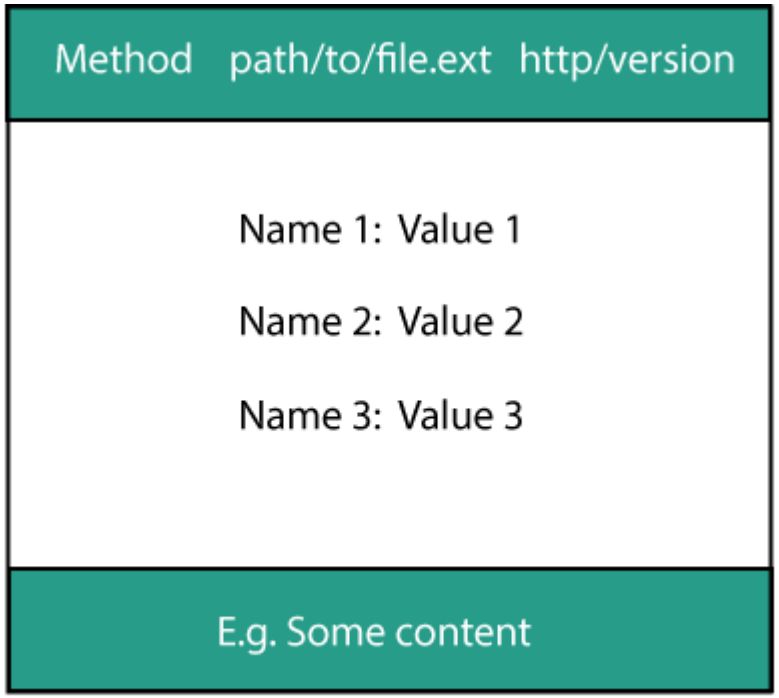


Fig:- Request HTTP Message

2) Initial Response Line

- The initial Response line is also known as the status line.
- It also has three parts: the HTTP version, a response status code that gives the result of the request, and the English reason phrase describing the status code.

Example:

HTTP/1.0 200 OK

or

HTTP/1.0 404 Not Found

Here,

- The HTTP version of the response line and request line are the same as "HTTP/x.x".

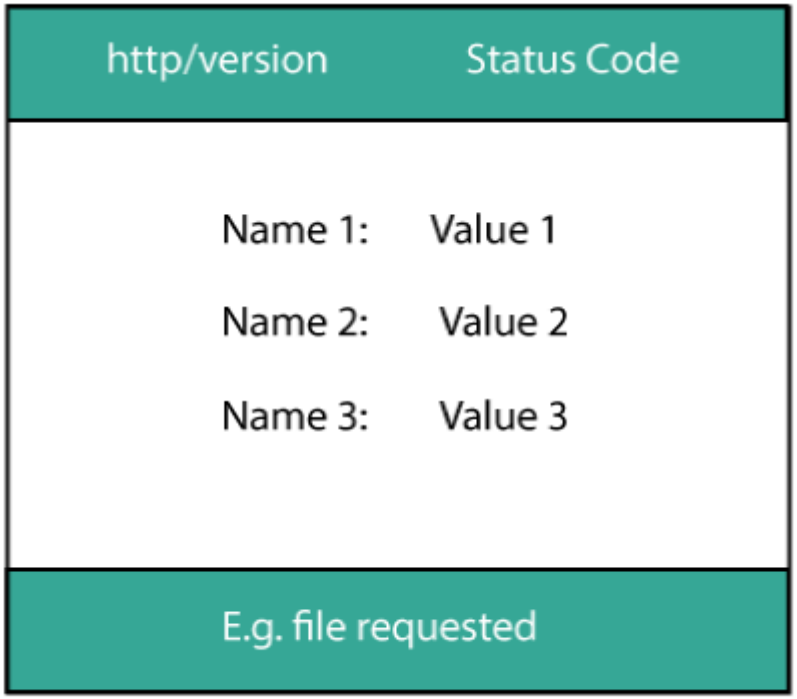


Fig:- Response HTTP Message

Message Headers

- The Message header provides information about the request and response. It also provides information about the object which is sent in the message body. Message Headers are of four types:

General Header: It has general applicability for both request messages and response messages.

Request Header: It has applicability only for the request messages.

Response Header: It has applicability only for the response messages.

Entity Header: It defines meta-information about the entity-body, and about the resource identified by request.

- All the above headers follow the same generic format. Each of the header fields consists of a name followed by a colon and the field values as follows:

message-header = field-name ":" [field-value]

Message Body

- The message body of an HTTP message is used to carry the entire body associated with the request and response. The message-body differs from the entire-body only when a transfer-coding has been applied, as indicated by the Transfer-Encoding header field.

Syntax

message-body = entity-body

| <entity-body encoded as per Transfer-Encoding>

- Transfer-Encoding MUST be used to indicate any transfer-codings which is applied by an application to ensure safe and proper transfer of the message. Transfer-Encoding is a property of the message.

Message Length

- The transfer-length of a message is the length of the message-body, and it appears in the message.
- In a message, when a message body is allowed, and Content-Length is given, its field value MUST exactly match the number of OCTETs in the message-body. When an invalid length is received and detected, the HTTP/1.1 user agents MUST notify the user.

General Header Fields

- Some header fields have the applicability for both the request and response messages. These header fields apply only when the message is transmitted.

Syntax

general-header = Cache-Control

HTTP Request

- HTTP Requests are messages which are sent by the client or user to initiate an action on the server.
- The first line of the message includes the request message from the client to the server, the method which is applied to the resource, identifier of the resource, and the protocol version.

Syntax

```
Request       = Request-Line
                *(( general-header
                | request-header
                | entity-header ) CRLF)
                CRLF
              [ message-body ]
```

Request Line

- The Request-Line starts with a method token, which is followed by the Request-URI, the protocol version, and ending with CRLF. Using the SP characters, the elements are separated.

Syntax

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

1) Method

The method token is used to indicate the method which was performed on the resource identified by the Request-URI. The method is case sensitive.

Syntax

Method = "OPTIONS"
| "GET"
| "HEAD"
| "POST"
| "PUT"
| "DELETE"
| "TRACE"
| "CONNECT"
| extension-method

extension-method = token

- A resource is allowed a list of methods and that methods can be specified in an Allow header field.
- The response's return code always notifies the client whether a method is currently allowed on a resource.
- Since the set of allowed methods can be changed dynamically.

Method and Description:

i) GET

This method retrieves information from the given server using a given URI. GET request can retrieve the data. It cannot apply other effects on the data.

ii) HEAD

The HEAD method is the same as the GET method. It is used to transfer the status line and header section only.

iii) POST

The POST request sends the data to the server. For example, file upload, customer information, etc. using the HTML forms.

iv) PUT

The PUT method is used to replace all the current representations of the target resource with the uploaded content.

v) DELETE

The DELETE method is used to remove all the current representations of the target resource, which is given by URI.

vi) CONNECT

The CONNECT method is used to establish a tunnel to the server, which is identified by a given URI.

2) Request-URI

The Request-URI is a Uniform Resource Identifier. It is used to identify the resource upon which to apply the request.

Syntax

Request-URI = "*" | absoluteURI | abs_path | authority

- On the nature of the request, these four options for Request-URI depend.

a) The asterisk "*" is used to show that the request does not apply to a particular resource, but it will apply to the server itself. It is allowed only when the method used does not necessarily apply to a resource.

Example

OPTIONS * HTTP/1.1

b) The **absoluteURI** form is used only when the request is being made to a proxy. The requested proxy is used to forward the request and return the response.

Example

GET http://www.javatpoint.com/WWW/TheProject.html HTTP/1.1

c) The **absolute path** can't be empty. If in the original URI, none is present, it must be given as "/".

d) The **authority** form is only used by the CONNECT method.

The Resource Identified by a Request

- Using the examination of Request-URI and the Host header field, we can determine the exact resource identified by the Internet request.
- An origin server must use the following rules for determining the requested resource on an HTTP/1.1 request if the origin server does differentiate based on the host requested.
- The host will be part of the Request-URI if Request-URI is an absoluteURI.
- The host will be determined by the Host header field value if the Request-URI is not an absoluteURI, and the request includes a header field of the host.
- The response MUST be a 400 (Bad Request) error message if the host as determined by rule 1 or 2 is not a valid host on the server.

Request Header Fields

- The request-header fields are used to allow the client to pass additional information to the server like the request and the client itself.
- The request header fields act as request modifiers, with semantics equivalent to the parameters on a programming language method invocation.

Syntax

request-header = Accept
| Accept-Charset
| Accept-Encoding
| Accept-Language
| Authorization
| Expect
| From
| Host

| If-Match
| If-Modified-Since
| If-None-Match
| If-Range
| If-Unmodified-Since
| Max-Forwards
| Proxy-Authorization
| Range
| Referer
| TE
| User-Agent

The name of the request-header field can be extended reliably only in combination with a change in the version of the protocol.

HTTP Response

- HTTP Response sent by a server to the client.
- The response is used to provide the client with the resource it requested.
- It is also used to inform the client that the action requested has been carried out.
- It can also inform the client that an error occurred in processing its request.
- An HTTP response contains the following things:
  - Status Line
  - Response Header Fields or a series of HTTP headers
  - Message Body

- In the request message, each HTTP header is followed by a carriage returns line feed (CRLF).
- After the last of the HTTP headers, an additional CRLF is used and then begins the message body.

Status Line

- In the response message, the status line is the first line. The status line contains three items:

a) HTTP Version Number

- It is used to show the HTTP specification to which the server has tried to make the message comply.

Example

HTTP-Version = HTTP/1.1

b) Status Code

- It is a three-digit number that indicates the result of the request.
- The first digit defines the class of the response.
- The last two digits do not have any categorization role. There are five values for the first digit, which are as follows:

Code and Description

1xx: Information

It shows that the request was received and continuing the process.

2xx: Success

It shows that the action was received successfully, understood, and accepted.

3xx: Redirection

It shows that further action must be taken to complete the request.

4xx: Client Error

It shows that the request contains incorrect syntax, or it cannot be fulfilled.

5xx: Server Error

It shows that the server failed to fulfil a valid request.

c) Reason Phrase

It is also known as the status text. It is a human-readable text that summarizes the meaning of the status code.

An example of the response line is as follows:

HTTP/1.1 200 OK

Here,

- HTTP/1.1 is the HTTP version.
- 200 is the status code.
- OK is the reason phrase.

Response Header Fields

- The HTTP Headers for the response of the server contain the information that a client can use to find out more about the response, and about the server that sent it.
- This information is used to assist the client with displaying the response to a user, with storing the response for the use of future, and with making further requests to the server now or in the future.

response-header = Accept-Ranges
| Age
| ETag
| Location
| Proxy-Authenticate
| Retry-After
| Server
| Vary
| WWW-Authenticate

- The name of the Response-header field can be extended reliably only in combination with a change in the version of the protocol.

Message Body

- The response's message body may be referred to for convenience as a response body.
- The body of the message is used for most responses. The exceptions are where a server is using certain status codes and where the server is responding to a client request, which asks for the headers but not the response body.

- For a response to a successful request, the body of the message contains either some information about the status of the action which is requested by the client or the resource which is requested by the client.
- For the response to an unsuccessful request, the body of the message might provide further information about some action the client needs to take to complete the request successfully or about the reason for the error.

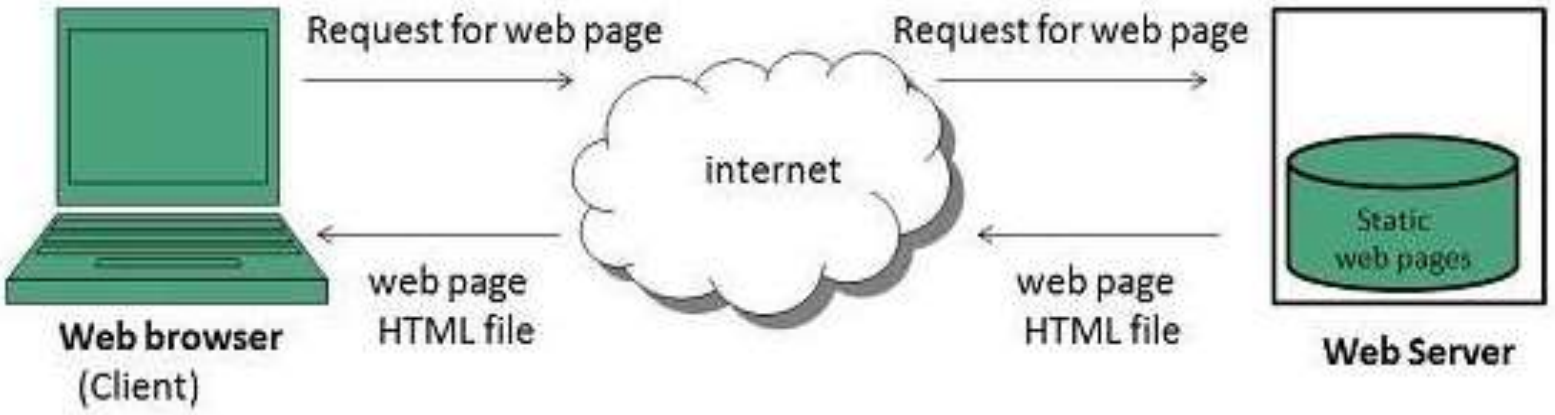
GENERATION OF DYNAMIC WEB PAGES

Web Page

- **Web page** is a document available on world wide web. Web Pages are stored on web server and can be viewed using a web browser.
- A web page can cotain huge information including text, graphics, audio, video and hyper links. These hyper links are the link to other web pages.
- Collection of linked web pages on a web server is known as **website.** There is unique **Uniform Resource Locator (URL)** is associated with each web page

Static Web page

- **Static web pages** are also known as flat or stationary web page.
- They are loaded on the client's browser as exactly they are stored on the web server. Such web pages contain only static information.
- User can only read the information but can't do any modification or interact with the information.
- Static web pages are created using only HTML.


Request for web page
Request for web page
internet
Static
web pages
web page
HTML file
web page
HTML file
Web browser
(Client)
Web Server

Dynamic Web page

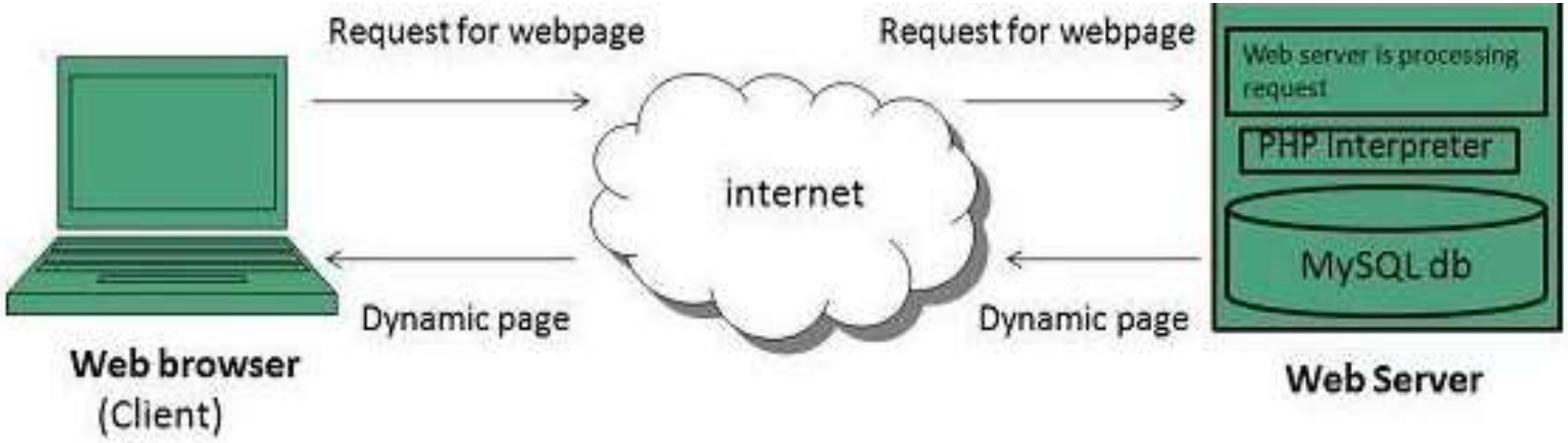
- **Dynamic web page** shows different information at different point of time.
- It is possible to change a portion of a web page without loading the entire web page.
- It has been made possible using **Ajax** technology.

1) Server-side dynamic web page

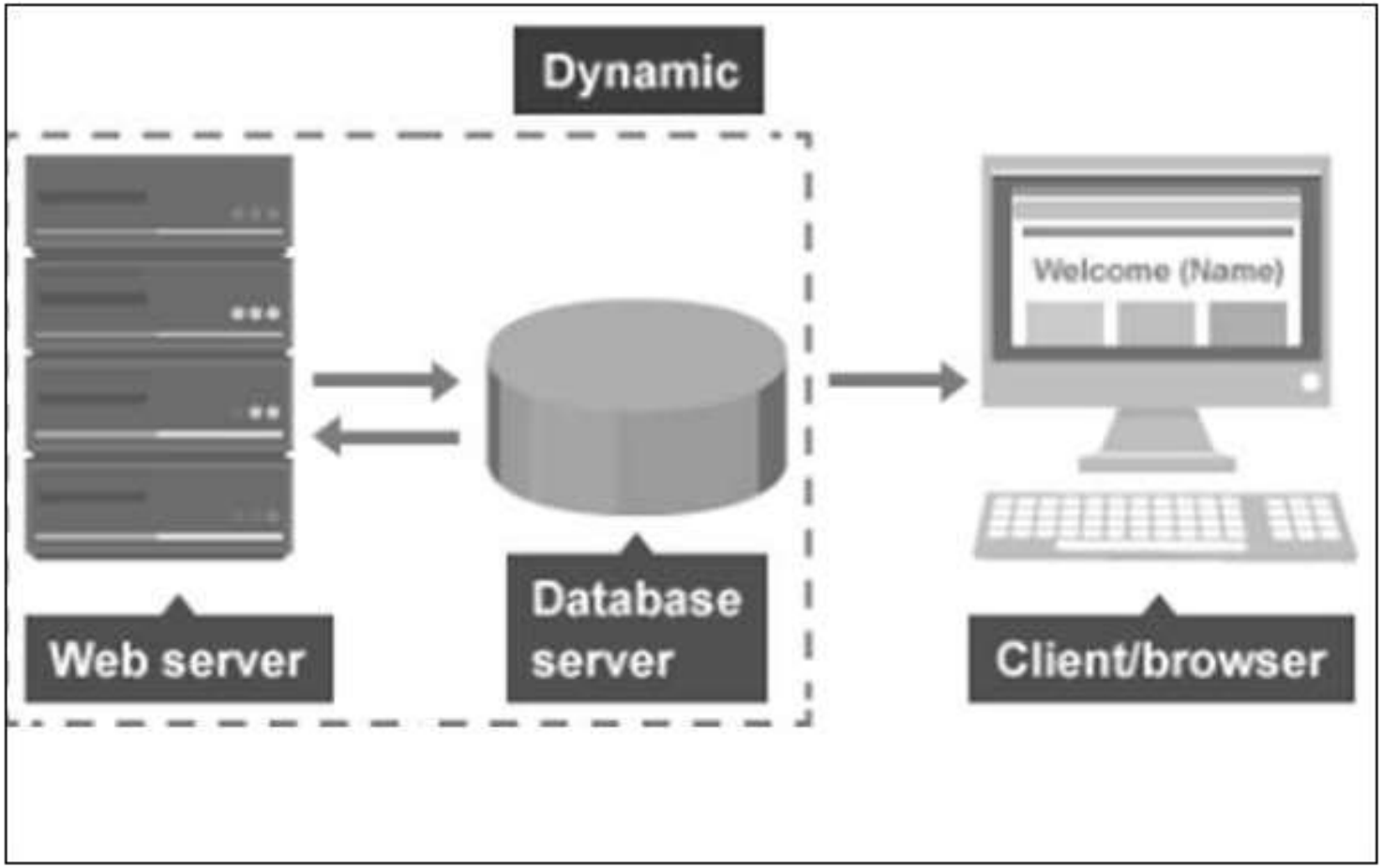
- It is created by using server-side scripting.
- There are server-side scripting parameters that determine how to assemble a new web page which also include setting up of more client-side processing.

2) Client-side dynamic web page

- It is processed using client side scripting such as JavaScript.
- And then passed in to **Document Object Model (DOM).**



Schematic view of a dynamic webpage

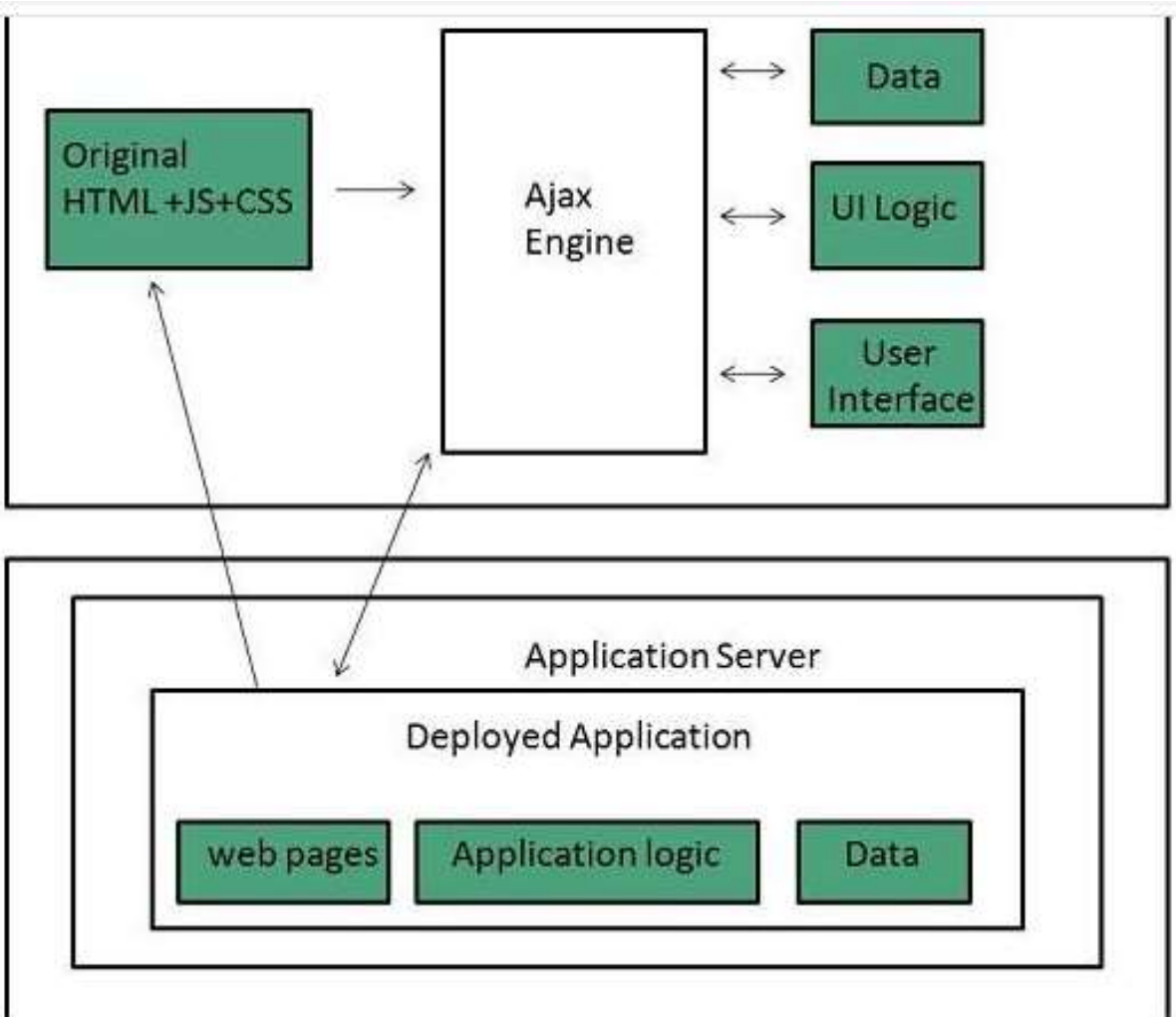


Scripting Languages

- Scripting languages are like programming languages that allow us to write programs in form of script.
- These scripts are interpreted not compiled and executed line by line.
- Scripting language is used to create dynamic web pages.

1) Client-side Scripting

- **Client-side scripting** refers to the programs that are executed on client-side. Client-side scripts contains the instruction for the browser to be executed in response to certain user's action.
- Client-side scripting programs can be embedded into HTML files or also can be kept as separate files.


Original
HTML +JS+CSS
Ajax
Engine
Data
UI Logic
User
Interface
Application Server
Deployed Application
web pages
Application logic
Data

Client-Side scripting languages

JavaScript

It is a prototype based scripting language. It inherits its naming conventions from java. All java script files are stored in file having **.js** extension.

ActionScript

It is an object oriented programming language used for the development of websites and software targeting Adobe flash player.

Dart

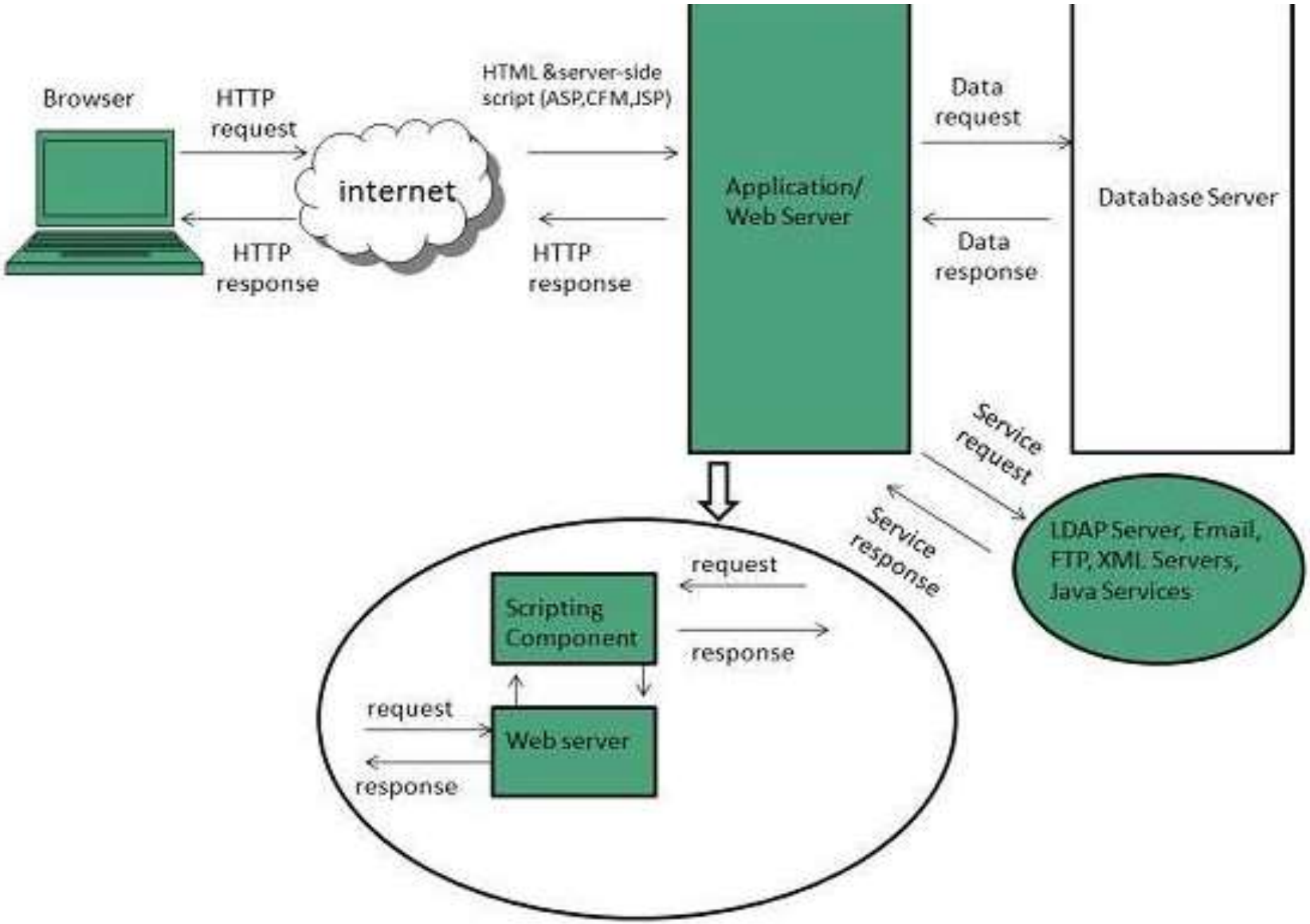
It is an open source web programming language developed by Google. It relies on source-to-source compiler to JavaScript.

VBScript

It is an open source web programming language developed by Microsoft. It is superset of JavaScript and adds optional static typing class-based object oriented programming.

2) Server-side Scripting

Sever-side scripting acts as an interface for the client and also limit the user access the resources on web server. It can also collects the user's characteristics in order to customize response.


Browser
HTTP request
internet
HTTP response
HTML &server-side script (ASP,CFM,JSP)
HTTP response
Application/ Web Server
Data request
Data response
Database Server
Service request
Service response
LDAP Server, Email, FTP, XML Servers, Java Services
request
response
Scripting Component
request
response
Web server

Server-Side scripting languages

ASP

Active Server Pages (ASP)is server-side script engine to create dynamic web pages. It supports **Component Object Model (COM)** which enables ASP web sites to access functionality of libraries such as DLL.

ActiveVFP

It is similar to PHP and also used for creating dynamic web pages. It uses native **Visual Foxpro** language and database.

ASP.net

It is used to develop dynamic websites, web applications, and web services.

Java

Java Server Pages are used for creating dynamic web applications. The Java code is compiled into byte code and run by **Java Virtual Machine (JVM).**

Python

It supports multiple programming paradigms such as object-oriented, and functional programming. It can also be used as non-scripting language using third party tools such as **Py2exe** or **Pyinstaller.**

WebDNA

It is also a server-side scripting language with an embedded database system.

Sr. No.	Key	Static Web Page	Dynamic Web Page
1	Definition	Static web pages are generally simple HTML written pages which serve as response from browser to server 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.
2	Complexity	As mentioned in above point as data in static web pages is static and do not require any interpretation before rendering so static web pages are simple in complexity.	Dynamic web pages on other hand does the interpretation process which make data dynamic in nature and due to which dynamic web pages become complex in complexity as compare to static web pages.
3	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.
4	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..
5	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.
6	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.