UNIT- I

1.1 BASIC CONCEPTS IN INTERNET

Internet:

- The internet is a collection of interconnected computer networks, linked by copper wires, fiber optic cables, wireless connections etc.
- Network is an interconnection of systems to share data and information. Internet is a network of network or collection of heterogeneous networks.

Web:

The web is a collection of interconnected documents, linked by hyperlinks and URLs and is accessible using the Internet.

Applications of internet:

- Electronic mail, World Wide Web, FTP, Telnet, Chat & Online transaction

HISTORY OF INTERNET AND WWW

History of the Internet:

- The USSR (Russia) launch of Sputnik spurred the U.S. to create the Advanced Research Project Agency (ARPA) in February 1958 to regain a technological lead.
- ARPA created the information processing technology office (IPTO), to further the research of the semi-Automatic Ground, Which had networked country-wide radar systems together for the first time.
- In late 1960's, one of the authors (HMD) research project MAC was funded by ARPA – of the Department of Defense (DOD).
- ARPA rolled out for networking the main computer systems of about a dozen ARPA – funded universities and research institutions.
- They were to be connected with communications lines operating at a then – stunning 56kbps (i.e., 56.000 bits per second) - this at a time of most people was connecting over telephone lines to computers at a rate of 110 bps.
- ARPA proceeded to implement the ARPANET which eventually evolved into today's internet.
- Researchers to share each other's computers, it rapidly became clear that enabling researchers to communicate quickly and easily (via) became known as electronic mail(e-mail).
- As e-mail facilities communications of all kinds among a billion people worldwide.

Goals for ARPANET:

- The Primary goal of ARPANET is to allow multiple users to send and receive information simultaneously over the same communication paths (e.g., Phone lines).
- The network operated with a technique called packet switching, in which digital data was sent in small bundles called packets.
- The packets contained address, error control and sequencing information. The address information allowed packets to be routed to their destination. The sequencing information helped in reassembling the packets; Packets from different senders were intermixed on the same lines. This packet switching

technique greatly reduced transmission costs, as compared with the cost of the dedicated communication lines.

- The network was designed to operate without centralized control. If a portion of the network failed, the remaining working portions would still route packets from senders to receivers over alternative paths for reliability. The protocol for communicating over the ARPANET became known as TCP – the Transmission Control Protocol.
- TCP ensured that messages were properly routed form sender to receiver and that they arrived intact. Internet involved, organizations worldwide were implementing their own networks for both
    - intero-organization (within the organization)
    - inter- organization (between organization)
- ARPA accomplished with internet protocol (IP), creating a "network of networks", the current architecture of the internet the combined set of protocols is commonly called TCP/IP.

Usage:

- Internet was limited to universities and research institutions; then the military began using the internet.
- The government decided to allow access to the internet for commercial purposes.

History of World Wide Web:

- The World Wide Web (WWW) allows computer users to locate and view multimedia – based documents over the internet.
- In 1989, Tim Berners – Lee of CERN (The European organization for Nuclear research) began to develop a technology for sharing information via a hyperlinked text documents.
- Berners – Lee called his invention the Hyper Text Markup Language(HTML)
- He also wrote communication protocols to form the backbone of his new information system, which he called the World Wide Web
- The Hyper Text Transfer protocol (HTTP) – a communication protocol used to send information over the web.
- Web use exploded with the availability in 1993 of the mosaic browser, which featured a user – friendly graphical interface.
- Mark Andreesen, whose team at NCSA developed mosaic, went on to found Netscape with initiating the explosive internet economy late 1990's.
- In September 1194, Berners-Lee founded the World Wide Web Consortium (W3c), which is the well-known standard making for free software's all over the world.
- The WWW became commercially viable during 1996-98 when a large number of dot-com companies used it for placing their services on the web.
- In past, most computer applications run on computers that were not connected to one another, whereas today's applications can be written to communicate among the world's computers.
- The internet mixes computing & communication technologies. It makes our work easier. It makes information instantly and conveniently accessible worldwide.
- It enables individuals and small business to get worldwide exposure. It is changing the way business is done.

- People can search for the best prices on virtually any product or service special interest communities can stay in touch with one another.
- HTTP – Communication between a web server and a web browser.
- HTTP is used for sending requests from a web client (browser) to a web server, returning web contents (web pages) from the server back to the client.

1.2 VARIOUS CONCEPTS IN HTML

- Hyper Text Markup Language (HTML) is a tag – based language and these tags are added to the pages, there by instructing the web browser about the format in which the page has to be displayed.
- The source code passed, the browser read web pages serially in the text format.
- HTML was display static content like text, with user specific formatting.

Fundamental HTML elements:

- HTML tags are usually used in pairs.
- All tags are enclosed between the angle brackets (< >)
- The start tag is usually between angle brackets, while the end also has a forwarding slash preceding
  the text (< / >).
- The element content is everything between the start and the end tag.
- Most HTML elements can have their attributes.
- HTML elements have empty content.
- Empty elements are closed in the start tag
- HTML tags are case sensitive.

The types of section are

1. Head section
2. Body section

Example:

```
<Html>
        <Head>
                <Title> Example Of Web </Title>
        </Head>
</Html>
```

Elements for the body section:

- The body tags <BODY> …. </BODY> contain the actual page content.
- All page’s text, images, forms links and other HTML come in the body block.
- The body tag has 3 commonly used attributes
    1. BGCOLOR
    2. TEXT
    3. LINK
- BGCOLOR specifies the background color for the page.
- TEXT specifies the color of the text on the page.
- LINK specifies the color of the hyperlink on the page.

HTML named colors:

Color values are represented in either hexadecimal numbers or color names. User can specify only 16 by names. It can specify any color by its hexadecimal values.

- White - #FFFFFF

- Yellow - #FFFF00
- Red - #FF00FF
- Green - #008000
- Blue - #0000FF
- Black - #000000

The default background, text and link colors for the web page are white, black and blue.
The various types of tags in HTML are

Example:

```
<HTML>
        <HEAD>
                <TITLE> HEADING OG VARIOUS SIZE </TITLE>
        </HEAD>
        <BODY>
                <H1> IP </H1>
                <H2> Web </H2>
                        .
                        .
                        .
                <H6> Technology </H6>
        </BODY>
</HTML>
```

Text level elements:

- &nbsp is used to insert a single blank space.
- <BR> indicates the line break sometimes user might want to use is to separate text into paragraphs.
- <P> …. </P> - Paragraph tag. These tags add a blank line before and after the block they enclose. Two line break tag can be used consecutively.
- <B>…..</B> - Bold tag
- <I>…..</I> - Italic tag
- <U>…..</U> - Underline tag
- It can effect as indicated in the modification of the font.

Lists of text:

- The most commonly list are of 2 types
    1. Ordered Lists (OL)
    2. Unordered Lists (UL)

- The ordered list tags <OL>……</OL> create ordered list item.
- The unordered list tags <UL>……</UL> create unordered list item.
- For each list item in the list, within either of these set of tag, user can use the listing tag (LI) – List Item, this tag is used singly and does not have an end tag.
- <HR> is used to draw a horizontal line on the web page requires only a single tag.

Definition List (DL) tag:

- Definition list are also commonly used to display information in the form of definitions which have common use to convey information to users.

- A definition has two parts
    1. A term (DT)
    2. A definition (DD)
- Definition list tag <DL>……</DL> - create a definition list
- Within these tags, user can use the definition term tag <DT> - for each list item term.
- Definition data tag <DD> - for each list item's data.
- Both the <DT> and <DD> tag have no associated end tags.

Font tag:

```
<Font color = “Red” Face = “Times” size = 5>
All data displayed here is red in color and has the times font and the size is 5
</Font>
```

Embedding Images:

- Images can be embedded into the web page. To add images as a background for the whole web page, the following format is used,

  <body background = “bg.gif”>

- To include the image as a normal figure in the web page the <img> tag is used.

Example:

< img src = “url “ height = “144” border = “1” width = “200” alt = “An image is here”>

</img>

The other related tag is <map> which is used to create hot spots. The syntax is

```
<map name = “ “>
……
</map>
```

Frames:

- Frame layout is one in which the browser windows is broken into multiple region called frames.
- Each frame contains different HTML documents.
- The <frameset> tag is a container for frames and replaces the body tag. </frameset>
- <frame> tag is used to place the contents into the frame.

Example:

<frameset rows = “value” cols = “value”>

- The attribute rows – The window is to be divided in horizontal stripes.
- The attribute cols – The window is to be divided in vertical stripes.

Tables:

- This form tag is the HTMLs best way of arranging information in space and controlling layout
- Table element to format a table
- The various tag in table as

  <table>……</table>

  align = left, center, right

border = make a border around the table & its cells

  - The <tr> element (table row) inserts a row in the table.
  - The <td> element (table detail) inserts a cell within a row
  - The <th> element (table header) to add headings to the rows and columns of the table.
- <caption> tag is used to be added to row and column headings. By default it will be aligned in center of a table.
- <border> is used for draw a border around the tables and the individual cells. Specify the border width in pixels.
- Color can be added to tables by using the bgcolor = and bordercolor = attributes. These attributes are available in the Table, <tr> and <td> elements; so user can apply colors to all the cell in a table, selected rows and individual cells.

Example:

    <table border = “1” cellspacing = “10%” cellpadding = “10%”> …. </table>

Spanning rows:

- The two types of attributes are
  - colspan – to span column
  - rowspan – to span rows

HTML Forms:

- Form provides a way to prompt the user for information and to carry out the actions based on the input.
- A form consists of one or more input controls that the user uses to enter text and select choices.
- Once the user provides the input, the form collects the data and sent it to a destination specified in the form element.
- To carry out the requested action, the server must have a script or other service that corresponds to the destination.
- A form can contain inputs like text fields, check boxes, passwords, radio-buttons, submit buttons and reset buttons.
- The <form> tag is used to create an HTML form.
  <form name=” frm1“ action=”index.jsp “ method=”get|post” >
  </form>
- HTML forms are used to pass the data to the server.

Input Types:

- Text Fields, Check Boxes, Password, Radio Button, Submit Button, Reset Button, Image Based Buttons, Scripted Buttons

The syntax is

    <input type = “text | passwd” name =”name” value =”default_value” size=”field size”>

Selection List (<select>):

- This form tag is used to set up a list of choices from
- </select>

Image Button:

    <input type=”image” src=” “ name=”submit” value=”submit”>

Filename:

```
<input type=”file” name=”filename” value=” “ size=25> Browser
```

Example:

Browse

- The form tag can be used in the HTML code.
  ```
  <form name=” frm1“ action=” hello.jsp “ method=”get | post” target=”response_frame”>
  </form>
  ```
- The method attribute is used to send the form data to the web server.
- The default attribute value for the method is get which appends the data to the end of the processing script URL.
- If the method has the value post, then the data is sent to the web server as a separate transaction.
- The value post is used when the form data is to be stored in a database or as a processing data in the web server.

1.3 INTERNET PROTOCOLS

- A protocol is a set of rules or an agreement that specifies a common language that computer on a network use for communication with other computers.
- It specifies the conditions under which a particular message should be sent or respond and the particular method of doing it.
- It specifies on how the computer talk with each other.
- The various protocols are
  - Hyper Text Transfer Protocol (HTTP)
  - Simple Mail Transfer Protocol (SMTP)
  - Post Office Protocol Version 3 (POP3)
  - Multipurpose Internet Mail Extension (MIME)
  - File Transfer Protocol (FTP)
  - Internet Protocol (IP)
  - Transport Control Message (TCP)
  - Internet Message Access Protocol (IMAP)

HTTP

- Hyper Text Transfer Protocol (HTTP) is a communication protocol used to send information over the web.
- HTTP protocol is used by the World Wide Web (WWW).
- HTTP defines how messages are formatted and transmitted and what actions web servers and browsers should in response to various commands.
- Example: When you enter a URL in your browser, this actually sends an HTTP command to the web server directing it to fetch and transmit the requested web page.
- It is a standard protocol for communication between the web browsers and the web servers.
- It is a stateless protocol that specifies how a client and a server establish a connection, how the client request data to the server, how the server responds to the client and finally how the connection is closed.
- Sending a request in the form of ACSII string and expects a reply (ASCII).

The basic structures of HTTP communications are:

1. Request Model
2. Response Model

HTTP REQUEST MESSAGE:

- A client sends an HTTP request to server by
  - Clicking a link on the web page
  - Submitting a form
  - Typing a web page address in the browser's address field
- The browser uses the URL (Uniform Resource Locator) to create the request message.

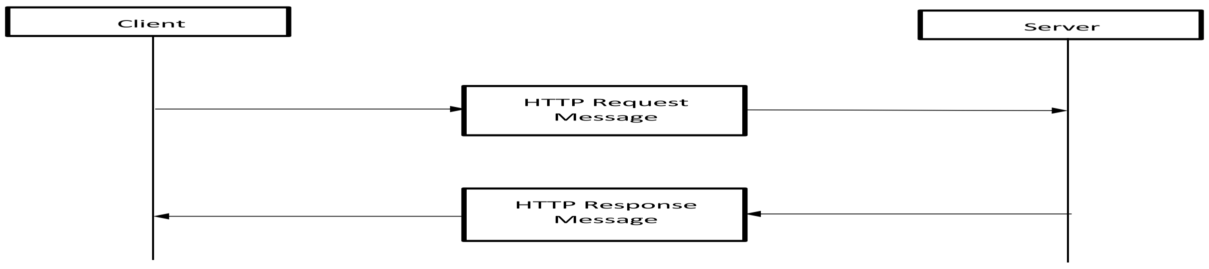


Fig 1.1: HTTP Request Message

The HTTP request message consists of

- Request / start line
- Request header / header field
- Blank line
- Request / message body (optional)

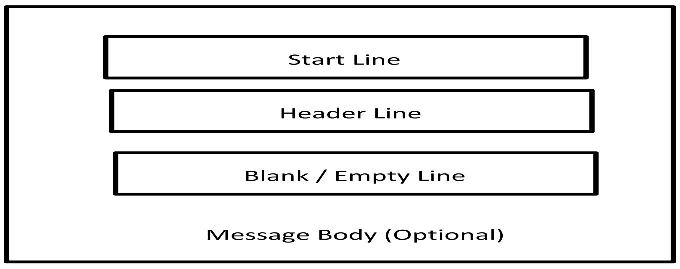


Fig 1.2 HTTP request message

a. **Start Line / Request Line**
It consists of 3 parts

i. Request Method (Example: GET)
ii. Request URI Portion of Web Address
iii. HTTP Version (Example: HTTP/1.1)
iv.

i. Request Method
The various methods supported by HTTP are:

METHODS	DESCRIPTIONS
GET	Gets a file from the server (or) tells the server the client wants to get some resource.
POST	Sends user information to the server (or) tells the server the client

	wants to get some resources and information will be sent by the client that may modify the request (ex: Submitting the form).
HEAD	Gets information about a file from the server.
PUT	Sends a file to be stored on the server (transfers a file from the client to the server).
DELETE	Deletes a file on the server.
OPTIONS	Requests the available server options.

ii. A Resource Identifier (Request URI)

- Uniform Resource Identifier (URI) is a string of characters used to identify a name or a resource on the internet.
- Such identification enables interaction with representations of the resource over a network (WWW) using a specific protocols.
- URI are composed of alpha-numeric names (punctuation characters are permitted) delimited by the character "/"
- URL (Uniform Resource Locator) is used for specifying any kind of information available on the internet.

The four elements of a URL specification are

- Method (Protocol)
- Host (Local hostname or IP address)
- Port (Port number for contacting server)
- Path (Pathname of the resource file)

iii. HTTP Version Identifier

- It specifies the version of the HTTP that the client understands.
- The string starts with the prefix HTTP / and is followed by a version number.
- Example: HTTP/1.1

HTTP RESPONSE MESSAGE:

- A server response is followed by a blank, status line and text of the requested page.

Example:

```
GET/HTTP/1.0
HTTP/1.0 200 OK
Date: Mon, 12 Dec 2011 11:40:40 GMT
Server: Apache/1.3.3.7 (UNIX)
Last_Modified: Wed, 08 Jan 2003 12:10:10 GMT
E tag: "3f80F-1b6-3e1cb03b"
```

Server receives a request and uses its URL to decide how to handle it.

- Accept-Range: bytes
- Content-Length: 438
- Connection: close
- Content-Type: text/html

- Etag (Entity Tag) header is used to determine if a cached version of the requested resource is identical to the current version of the resource on the server.
- Content-Type specifies the internet media type of the data conveyed by the HTTP message.
- Content-Length indicates it length in bytes.
- The HTTP/1.1 Web Server publishes its ability to respond to requests for certain byte range of the document – accept ranges bytes.
- Connection: Close is sent in a header. It means that the web server will close the TCP connection immediately after the transfer of this response.

SMTP

- The TCP/IP protocol that supports electronic mail on the internet is called the **Simple Mail Transfer Protocol (SMTP).**
- SMTP is the protocol used by mail servers to exchange email messages.
- It is a system for sending messages to other computer users based on e-mail addresses.

SMTP provides for mail exchanges between users on the same or different computers and supports are

- Sending a single message to one or more recipients.
- Sending messages that include text, audio, video format or graphics file.
- Sending messages to users on networks.

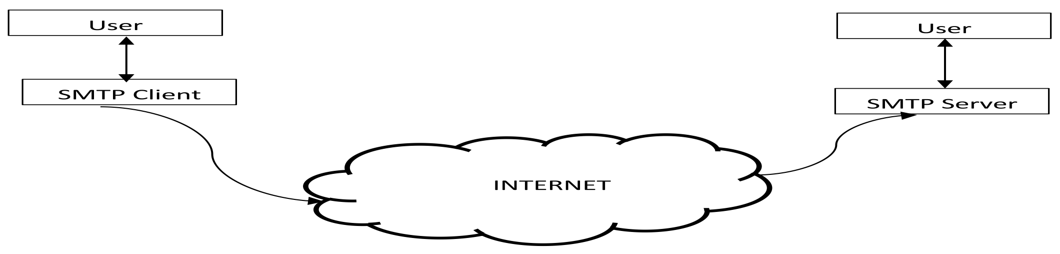


Fig 1.3: SMTP Concept

SMTP client and server has 2 components

a) User Agent (UA)
b) Mail Transfer Agent (MTA)

- The UA prepares the message, create the envelope and puts the message in the envelope.
- The MTA transfers the mail across the internet.

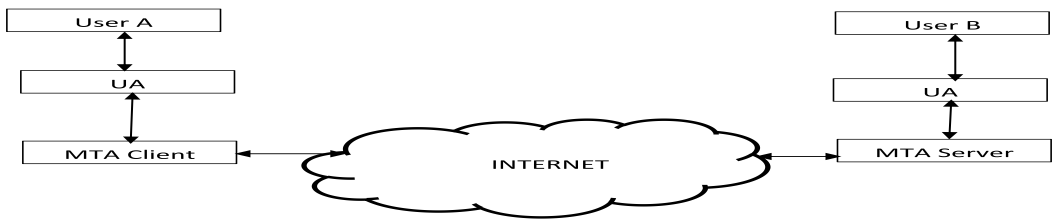


Fig 1.4: MTA transfers the mail across the internet.

- A mail gateway which is a relay MTA that can receive mail prepared by a protocol other than SMTP and transform it to SMTP format before sending it.

- It can also receive mail in SMTP format and change it to another format before sending it.

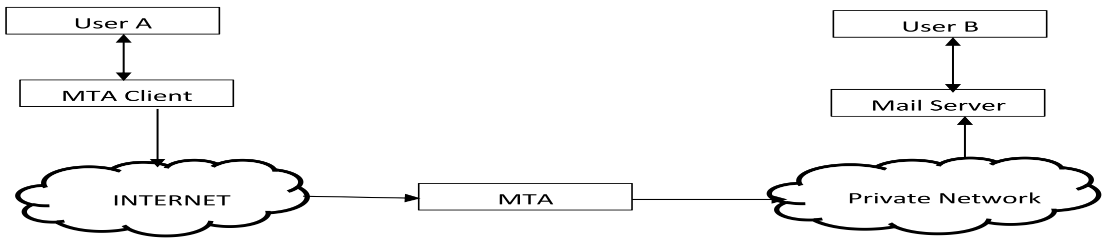


Fig1.5 : Mail gateway

POP3

- Post Office Protocol Version3 (POP3) is used to retrieve e-mail from an internet mailbox.
- POP3 is used to retrieve mail for a single user; typically the POP server has access to a database of email messages created by an SMTP server.
- POP3 is used to download messages from the server.
- POP3 connections require authentication in the form of a secret (i.e.,) shared by the user and the POP server (a password)
- POP3 session passes
  - Authentication
  - Transaction
  - Update
- In the authorization state, the client identifies itself to the server.
- If the authorization is successful, the server opens the client's mailbox and the session enters the transaction state.
- In this state, the client requests the POP3 server provide information or perform an action.
- When it enters the update state, the connection terminates.

POP commands:

- POP commands and replies are formatted as ASCII lines and all replies start with either "+OK" or "-ERR".

  The various POP commands are

Command	Description
USER	Requires a name that identifies the user (Specify username).
PASS	Specify password for the user/server.
STAT	Get mailbox status (no. of messages in the mailbox).
LIST	Get a list of messages and size, one per line, termination line contains a period.
RETR	Retrieves message from mailbox.
DELE	Marks a message for deletion.
LAST	The server returns the highest message no accessed.
RSET	Unmarks all messages marked for deletion.
QUIT	Remove marked messages and close the TCP connection.

- "+OK" is a positive success indicator similar to an ACK message
- The text "-ERR" is a negative success indicator similar to a NAK message

MIME

- MIME is an acronym of Multipurpose Internet Mail extensions.
- It is an international standard that deals with the format of messages exchanged between different e-mail systems.
- It is a specification for formatting non – ASCII messages so that they can be sent over the internet.
- Many e-mail clients now support MIME, which enables them to send and receive graphics, audio and video files via the internet mail system.
- MIME supports messages in character set other than ASCII.
- MIME was defined in 1992 by the internet engineering task force (IETF).
- A new version called s/MIME supports encrypted message.
- The format of Internet mail to allow Non – US – ASCII textual messages, non – textual messages, multipart message bodies and non – US – ASCII information in message headers.

Need for MIME:

- Messages contain only ASCII characters.
- Messages with only 1000 characters.
- Messages should not exceed certain length.

Features of MIME:

MIME allow mail messages to have

- Multiple objects in a single message.
- Message with any no. of lines or unlimited overall length.
- Characters other than ASCII, allowing non – English messages.
- Multi – font messages.
- Binary or application specific files.
- Images, audio, video and multipart messages.

- MIME transforms non-ASCII data at the sender site to NVT ASCII data and delivers them to the client MTA to be sent through the Internet.
- The message at the receiving side is transformed back to the original data.
- NVT→(Network Virtual Terminal)

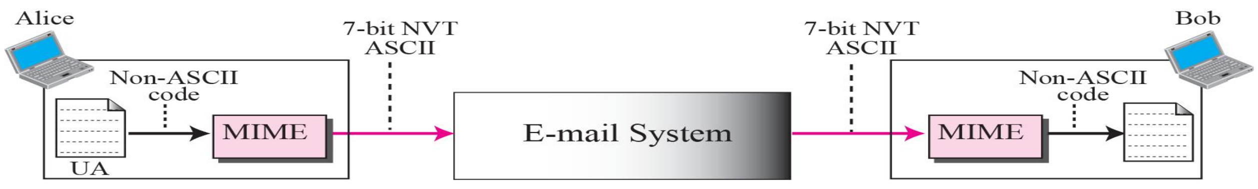


Fig1.6: MIME

MIME Headers:

The 5 header fields to internet e-mail messages.

1. MIME version
2. Content type
3. Content transfer encoding
4. Content id
5. Content description

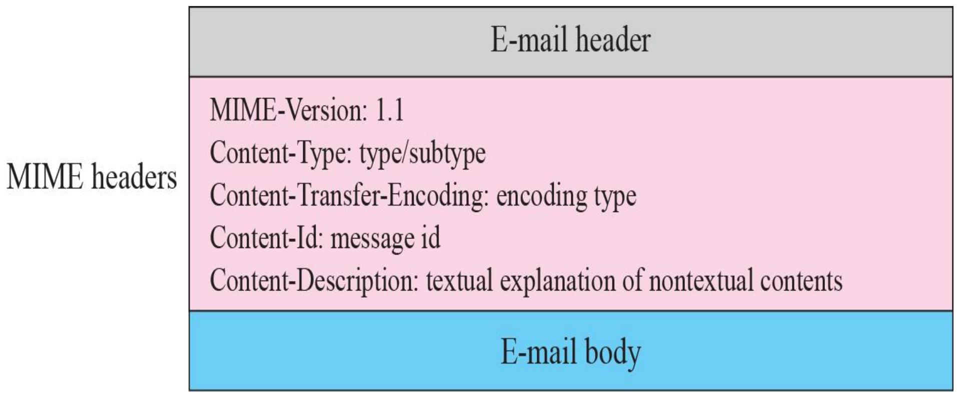

E-mail header
MIME headers
MIME-Version: 1.1
Content-Type: type/subtype
Content-Transfer-Encoding: encoding type
Content-Id: message id
Content-Description: textual explanation of nontextual contents
E-mail body

Fig 1.7: MIME Headers

1. **MIME-Version:**
   - This header defines the version of MIME used.
   - The current version is 1.1

     MIME-Version: 1.1

2. **Content type:**
   - This header defines the type of data used in the body of the message.
   - The content type and the content subtype are separated by a slash.
   - Depending on the subtype, the header may contain other parameters.

     Content-Type : < type / sub-type >

IMAP

- IMAP stands for Internet Message Access protocol.
- This protocol is used to access the messages in e-mail or electronic bulletin board that are in mail server
- It is an application layer Internet protocol that allows e-mail client to access e-mail on a remote mail server
- The current version IMAP4 is defined by RFC3501.
- IMAP server on well-known port no.143

The objectives of IMAP are

1. Compatible with internet messaging standards Ex: MIME
2. Allow message access from multiple computers.
3. It supports for online, offline and disconnected access nodes.
4. Support for concurrent access to shared mailboxes.
5. Client software needs no knowledge about the servers file store format.

- The feature of IMAP is the mail messages remain on the server, instead of being downloaded to a computer.
- Checking the mail with a client or web-based environment using the protocol.
- IMAP supports the use of folders for mail organization, but instead of organizing the messages on the local computer, these folders are kept on the server.
- IMAP is quicker access to mail.
- The message headers are initially downloaded so the user can choose to download, open and read only this message.

- Using IMAP and saving messages on the server is that the user will be restricted.

The protocols includes operations for
- Creating mailboxes.
- Deleting mailboxes.
- Renaming mailboxes.
- Checking for new messages.
- Permanently removing messages.
- Setting and clearing flags.
- Selective fetching of message attributes, text and portion of efficiency.

1.4 DNS

Domain Name System (DNS)

- To identify an entity, TCP/IP protocols use the IP address which uniquely identifies the connection of a host to the internet.
- We prefer to use names instead of address.
- We need a system that can map a name to an address and conversely an address to a name. In TCP/IP, this is the domain name system (DNS).
- What internet users use to reference anything by the name of the internet?
- The mechanism by which Internet software translates name to addresses.

The Namespace

- The namespace is the structure of the DNS database.
  - An inverted tree with the root node at the top.
- Each node has a label.
  - The root node has a null node written as " ".

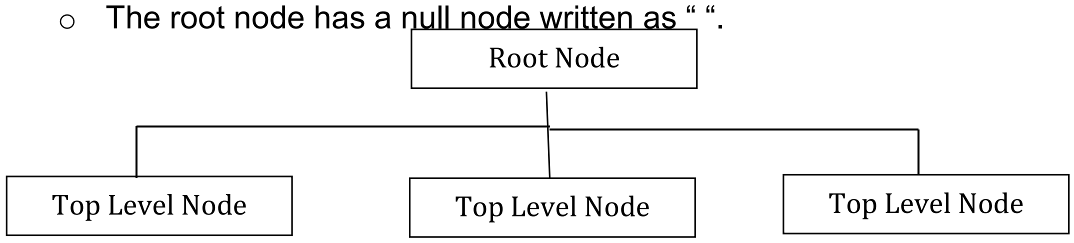


Fig 1.8 : Structure of the DNS Database

DNS in the Internet:

- A domain name is the sequence of labels from a node to the root separated by dots (".”’s), read left to right.
  - The namespace has a maximum depth of 127 levels.
  - Domain names are limited to 64 characters in length.
- A node’s domain name identifies its position in the namespace.
- DNS is a protocol that can be used in different platforms.

In the internet, the domain namespace (tree) is divided onto 3 different sections.
- Generic domains
- Country domains
- Inverse domains

1. Generic Domains:

- The generic domains define registered host according to their generic behavior.

- Each node in the tree defines a domain, which is an index to the domain namespace database.
- It represents as 3 character labels.
- Ex: com, edu.

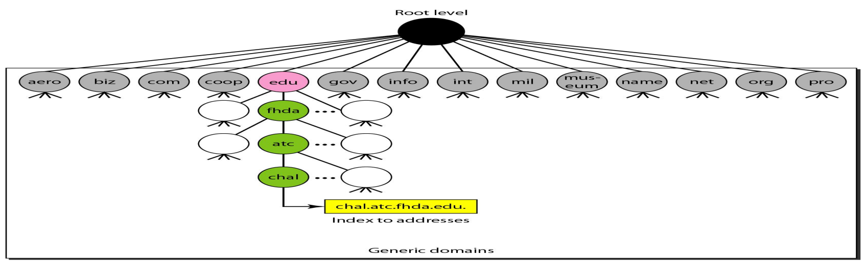


Fig 1.9 : Generic Domain

Sub-Domain

- One domain is a sub domain of another if its apex node is a descendant of the others apex node.
- One domain is a sub domain of another if its domain name is in the other's domain name.
- Ex: sales.google.com is a subdomain of
  - google.com
  - com
- Administrators can create subdomains to group hosts.
- The parent domains retain links to the delegated subdomains.

1.5 WEB BROWSERS AND WEB SERVERS

WEB BROWSERS:

- The internet is an essential medium for communicating and interacting with people worldwide.
- A web browser is used to display web pages. Common web browsers are Netscape navigator and internet explorer.
- The web browser is the interpreter of our web sites.
- Web browsers are software programs that allow users to access the web content.
- Millions of people use web browsers to access the tremendous amount of information available on the web and to store or exchange this content with other users.
- The popular web browsers are
  - Microsoft's Internet Explorer
  - Mozilla's Firefox
  - Apple's Safari
  - Opera Software's Opera
  - Google Chrome
- Web browsers, a software application used to locate, retrieve and also display content on the worldwide web, including web pages, images, videos and other files.
- As a client/server model, the browser is the client run on a computer that contacts the web server and request information.

- The web server sends the information back to the web browser which displays the results on the computer or other internet enabled device that supports a browser.
- Browsers are fully functional software suites that can interpret and display HTML web pages, applications, java scripts, AJAX and other content hosted on web servers.
- Many browsers offer plug-in which extend the capabilities of a browser so it can display multimedia information (including sound and video) or the browser can be used to perform tasks such as video conferencing, to design web pages or add ant phishing filters and other security features to the browser.

Web Servers:

- Web servers are computers that deliver (serves up) web pages.
- Every web server has an IP address and possibly a domain name.
- Ex: if you enter the URL http://www.google.com/index.html in your web browser. This sends a request to the web server whose domain name is google.com. The server then fetches the page named index.html and sends it to your browser.
- Any computer can be turned into a web server by installing server software and connecting the machine to the internet.
- Web server software applications including public domain software from NCSA (National Computer Security Association) and Apache web server.
- The web browser includes 3 categories as
    - Static web document
    - Dynamic web document
    - Active web document

Static Documents:

- Static documents are fixed content documents that are created and stored in a server.
- The client can get only a copy of the document.
- The content in the server can be changed but the user cannot change it.
- When a client accesses the document, a copy of the document is sent. The user then uses a browser to display the content.
- Ex: HTML web page file

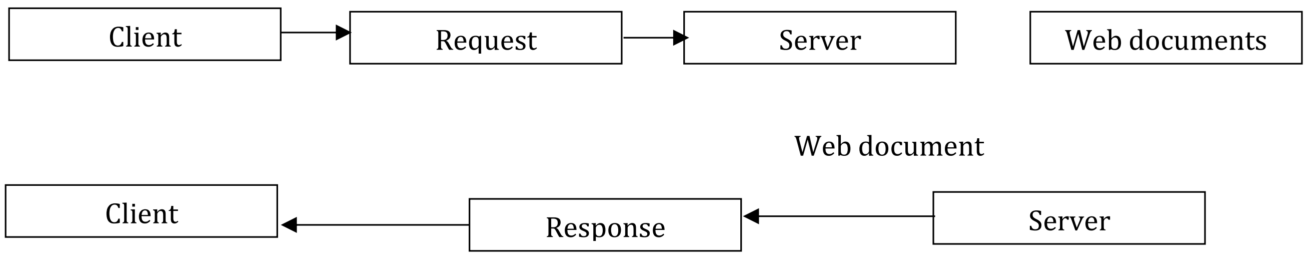
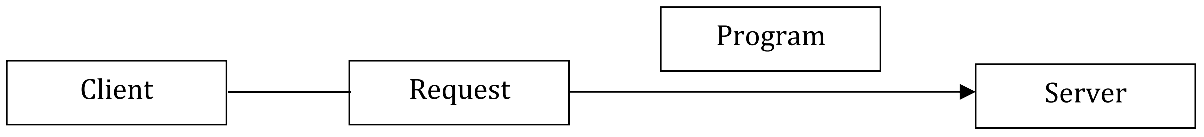


Fig 1.10: Static Document

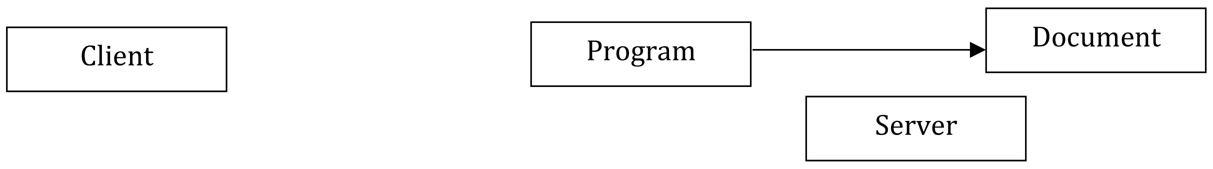
Dynamic Document:

- A dynamic document is created by a web server whenever a browser requests the document.

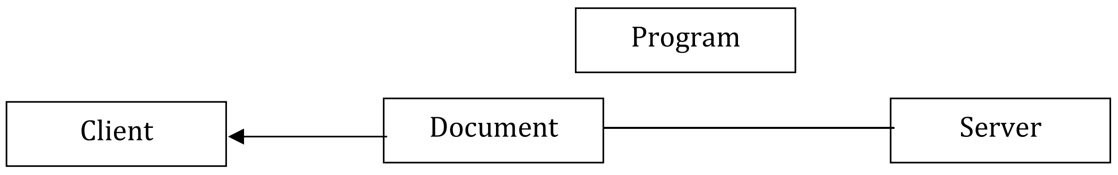
- When a request arrives, the web server runs an application program that creates the dynamic document.
- The server returns the output of the program as a response to the browser that requested the document.
- Because a fresh document is created for each request, the contents of a dynamic document can vary from one request to another.
- Ex: Dynamic document is getting the time and date from the server.
- Time and date are kinds of information that is dynamic and they change from moment to moment.
- Cricket refresh button.



a. Request for running a program



b. Running the program and creating the document



c. Response

Fig 1.10: Dynamic Document

- A server that handles dynamic documents as
  - The server examines the URL to find if it defines a dynamic document.
  - If the URL defines a dynamic document, the server executes the program.
  - The server sends the output of the program to the client (browser).

Common Gateway Interface (CGI):
- It creates and handles dynamic documents.
- CGI is a set of standards that defines how a dynamic document should be written, how input supplied to the program and how the output result is used.
- CGI program is a gateway that can be used to access other resources such as database, graphic, packages.

Active Document:
- We need a program to be a run at the client side.
- Ex: we want to run a program that creates animated graphics on the screen or interacts with the user.

- The program can be run at the client side where the animation or interaction taken place.
- An active document in the server is stored in the form of binary code.
- Ex: Java Applet

1.6 CSS

Introduction to (CSS):

- **CSS** stands for **C**ascading **S**tyle **S**heets
- CSS describes **how HTML elements are to be displayed on screen, paper, or in other media**
- CSS **saves a lot of work**. It can control the layout of multiple web pages all at once
- External style sheets are stored in **CSS files.**
- CSS are rules or styles for organizing the layout of an HTML document including its color, typefaces, margins, links and other formatting elements.
- Style sheets make it easier to create an index because indexing software has only to read the structural elements rather than full content page.
- User includes multiple set of style sheet information.
- It contains rules, composed of selectors and declarations that define how styles are applied.
- The selector (HTML element, class name or ID name) is the link between the HTML document and the style.
  - HTML element tags
  - Attributes (class, ID name)

Rule or Syntax:

A CSS rule set consists of a selector and a declaration block:

Selector { property1: value1; property2: value2; } or **tagname {style_attribute value:}**

```
h1      { color        : blue;
          font-size   : 12px;    }
```

- The selector points to the HTML element you want to style.
- The declaration block contains one or more declarations separated by semicolons.
- Each declaration includes a property name and a value, separated by a colon.

Example 1:

A CSS declaration always ends with a semicolon, and declaration groups are surrounded by curly braces:

```
p {
        color:red;
        text-align:center;
  }
```

Example 2:

```
<html>
    <head>
            <style type="text/css" >
                    body {background-color:yellow;}
                    h1   {font-size:36pt;}
                    h2   {color:blue;}
                    p    {margin-left:50px;}
            </style>
</head>
<body>
            <h1>This header is 36 pt</h1>
            <h2>This header is blue</h2>
            <p>This paragraph has a left margin of 50 pixels</p>
</body>
</html>
```

- To assign more than one kind of style information at the same time, separate the styles with semicolon.

1.7 JavaScript

JavaScript is a scripting language mainly used for writing dynamic Web pages. When a script written in JavaScript is embedded in a Web page, it will be executed by the Web browser on the client machine.
JavaScript supports functions as first-class functions - Functions are really objects. Like regular objects, functions can be created during execution, stored in data structure, and passed to other functions as arguments.

- JavaScript is a client – side scripting language for the World Wide Web that is similar to the syntax of the Java programming language.
- JavaScript is designed to provide limited programming functionality.

A simple JavaScript program:

```
<html>
<head>
<Title> Hello World  </Title>
</head>
<body>
<script language="javascript">
document.write("Hello,World wide web");
</script>
</body>
 </html>
```

Client-side JavaScript:

- Client-side JavaScript is the most common form of the language. The script should be included in or referenced by an HTML document for the code to be interpreted by the browser.
- It means that a web page need not be a static HTML, but can include programs that interact with the user, control the browser, and dynamically create HTML content.
- The JavaScript client-side mechanism provides many advantages over traditional CGI server-side scripts. For example, you might use JavaScript to check if the user has entered a valid e-mail address in a form field.
- The JavaScript code is executed when the user submits the form, and only if all the entries are valid, they would be submitted to the Web Server.
- JavaScript can be used to trap user-initiated events such as button clicks, link navigation, and other actions that the user initiates explicitly or implicitly.

Advantages of JavaScript:

The merits of using JavaScript are

- **Less server interaction** − You can validate user input before sending the page off to the server. This saves server traffic, which means less load on your server.
- **Immediate feedback to the visitors** − They don't have to wait for a page reload to see if they have forgotten to enter something.
- **Increased interactivity** − You can create interfaces that react when the user hovers over them with a mouse or activates them via the keyboard.
- **Richer interfaces** − You can use JavaScript to include such items as drag-and-drop components and sliders to give a Rich Interface to your site visitors.

Limitations of JavaScript:

The JavaScript as a full-fledged programming language. It lacks the following important features

- Client-side JavaScript does not allow the reading or writing of files. This has been kept for security reason.
- JavaScript cannot be used for networking applications because there is no such support available.
- JavaScript doesn't have any multithreading or multiprocessor capabilities.
- JavaScript is a lightweight, interpreted programming language that allows you to build interactivity into otherwise static HTML pages.

1.8 JavaScript Operators

JavaScript

JavaScript is a scripting language mainly used for writing dynamic Web pages. When a script written in JavaScript is embedded in a Web page, it will be executed by the Web browser on the client machine.

JavaScript supports functions as first-class functions - Functions are really objects. Like regular objects, functions can be created during execution, stored in data structure, and passed to other functions as arguments.

- JavaScript is a client – side scripting language for the World Wide Web that is similar to the syntax of the Java programming language.
- JavaScript is designed to provide limited programming functionality.

A simple JavaScript program:

```
<html>
<head>
<Title> Hello World  </Title>
</head>
<body>
<script language="javascript">
document.write("Hello,World wide web");
</script>
</body>
  </html>
```

1.9 JavaScript in Perspective

Javascript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted programming language with object-oriented capabilities.

JavaScript Syntax

- JavaScript can be implemented using JavaScript statements that are placed within the **<script>... </script>** HTML tags in a web page.
- You can place the **<script>** tags, containing your JavaScript, anywhere within you web page, but it is normally recommended that you should keep it within the **<head>** tags.
- JavaScript can be placed in the <body> and the <head> sections of an HTML page.
- The **<script> tag** alerts the browser program to start interpreting all the text between these tags as a script.

A simple syntax of your JavaScript will appear as follows.

```
<script…>
    JavaScript code
</script>
```

The script tag takes two important attributes −

- **Language** − This attribute specifies what scripting language you are using. Typically, its value will be Javascript. Although recent versions of HTML have phased out the use of this attribute.
- **Type** − This attribute is what is now recommended to indicate the scripting language in use and its value should be set to "text/javascript".

The JavaScript segment will look like –

```
<script language="Javascript"  type="text/Javascript">
    JavaScript code
</script>
```

Example Program:

```
<html>
  <body>
    <script language="Javascript" type="text/Javascript">
            document.write("Hello World!")
    </script>
  </body>
</html>
```

Output:

This code will produce the following result –

Hello World!

1.10 JavaScript Variables:

- JavaScript variables are containers for storing data values.
- You can place data into these containers and then refer to the data simply by naming the container.
- Before you use a variable in a JavaScript program, you must declare it.

Variables are declared with the **var** keyword as follows.

```
<script type="text/javascript">
    var money;
    var name;
```

```
</script>
```

You can also declare multiple variables with the same **var** keyword as follows –

```
<script type="text/javascript">
    var money, name;
</script>
```

JavaScript Identifiers:

- All JavaScript **variables** must be **identified** with **unique names**.
- These unique names are called **identifiers**.
- JavaScript identifiers are case-sensitive.
- Identifiers can be short names (like x and y), or more descriptive names (age, sum, totalVolume).

The general rules for constructing names for variables (unique identifiers) are:

- Names can contain letters, digits, underscores, and dollar signs.
- Names must begin with a letter.
- Names can also begin with $ and _.
- Names are case sensitive (y and Y are different variables).
- Reserved words (like JavaScript keywords) cannot be used as names.

Storing a value in a variable is called **variable initialization**. You can do variable initialization at the time of variable creation or at a later point in time when you need that variable.

For instance, you might create a variable named **money** and assign the value 2000.50 to it later. For another variable, you can assign a value at the time of initialization as follows.

```
<script type="text/javascript">
    var name = "Ali";
    var money;
    money = 2000.50;
</script>
```

1.11 JavaScript Data Types:

JavaScript provides different **data types** to hold different types of values. There are two types of data types in JavaScript.

1. Primitive data type
2. Non-primitive (reference) data type

JavaScript is a **dynamic type language**, means you don't need to specify type of the variable because it is dynamically used by JavaScript engine. You need to use **var** here to specify the data type.

It can hold any type of values such as numbers, strings etc.

For example:

- var a=40;   //holding number
- var b="Rahul";   //holding string

1. JavaScript primitive data types:

There are five types of primitive data types in JavaScript. They are as follows:

(i) JavaScript String:

- A string (or a text string) is a series of characters like "John Doe".

- Strings are written with quotes. You can use single or double quotes:

Example:

```
var carName = "Volvo XC60";         //Using double quotes
var carName = 'Volvo XC60';   // Using single quotes
```

(ii) Javascript Number:

- JavaScript has only one type of numbers.
- Numbers can be written with, or without decimals:

Example:

```
var x1 = 34.00;     // Written with decimals
var x2 = 34;        // Written without decimals
```

(iii) JavaScript Boolean:

- Booleans can only have two values: true or false.
- Booleans are often used in conditional testing.

Example:

```
var x = true;
var y = false;
```

(iv) Undefined:

In JavaScript, a variable without a value, has the value **undefined**. The typeof is also **undefined**.

Example:

```
var person;                // Value is undefined, type is undefined
```

Any variable can be emptied, by setting the value to **undefined**. The type will also be **undefined**.

(v) Empty Values

- An empty value has nothing to do with undefined.
- An empty string variable has both a value and a type.

Example:

```
var car = "";              // The value is "", the typeof is string
```

(vi) Null:

- In JavaScript null is "nothing". It is supposed to be something that doesn't exist.
- Unfortunately, in JavaScript, the data type of null is an object.

You can empty an object by setting it to null:

Example:

```
var person = null;         // Value is null, but type is still an object
```

2. JavaScript non-primitive data types:

The non-primitive data types are as follows:

(i) JavaScript Object:

- JavaScript objects are written with curly braces.
- Object properties are written as name:value pairs, separated by commas.

Example:

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

The object (person) in the example above has 4 properties: firstName, lastName, age, and eyeColor.

(ii) JavaScript Array:

- JavaScript arrays are written with square brackets.
- Array items are separated by commas.

The following code declares (creates) an array called cars, containing three items (car names):

Example:

var cars = ["Saab", "Volvo", "BMW"];

Array indexes are zero-based, which means the first item is [0], second is [1], and so on.

(iii) JavaScript RegExp:

- A regular expression is an object that describes a pattern of characters.
- Regular expressions are used to perform pattern-matching and "search-and-replace" functions on text.

Syntax:

*/*pattern*/*modifiers*;*

Example:

var patt = /w3schools/i

- **/w3schools/i** is a regular expression.
- **w3schools** is a pattern (to be used in a search).
- **i** is a modifier (modifies the search to be case-insensitive).

1.12 JavaScript Statements

Conditional statements are used to perform different actions based on different conditions.

In JavaScript we have the following conditional statements:

i. Use **if** to specify a block of code to be executed, if a specified condition is true.
ii. Use **else** to specify a block of code to be executed, if the same condition is false.
iii. Use **else if** to specify a new condition to test, if the first condition is false.
iv. Use **switch** to specify many alternative blocks of code to be executed.

The if Statement:

Use the **if** statement to specify a block of JavaScript code to be executed if a condition is true.

Syntax:

```
if (condition)
{
   block of code to be executed if the condition is true
}
```

(ii) The else Statement:

Use the **else** statement to specify a block of code to be executed if the condition is false.

Syntax:

```
if (condition)
{
   block of code to be executed if the condition is true
}
else
{
   block of code to be executed if the condition is false
}
```

(iii) The else if Statement:

Use the **else if** statement to specify a new condition if the first condition is false.

Syntax:

```
if (condition1)
{
   block of code to be executed if condition1 is true
}
else if (condition2)
{
   block of code to be executed if the condition1 is false and condition2 is
true
}
else
 {
   block of code to be executed if the condition1 is false and condition2 is
false
}
```

Arithmetic Operators:

Arithmetic operators are used to perform arithmetic on numbers (literals or variables).

Operator	**Description**
+	Addition

-	Subtraction
*	Multiplication
/	Division
%	Modulus
++	Increment
--	Decrement

Comparison Operators:

JavaScript supports the following comparison operators.

Operator	Description
= =	Equal
!=	Not Equal
>	Greater than
<	Less than
>=	Greater than or Equal to
<=	Less than or Equal to

Logical (or Relational) Operators:

JavaScript supports the following logical operators.

Operator	Description
&&	Logical AND
||	Logical OR
!	Logical NOT

Bitwise Operators:

JavaScript supports the following bitwise operators.

Operator	Description
&	Bitwise AND
|	Bitwise OR
^	Bitwise XOR
~	Bitwise Not
<<	Left Shift
>>	Right Shift
>>>	Right shift with Zero

Assignment Operators:

JavaScript supports the following assignment operators.

Operator	Description
=	x=y
+=	x+=y

-=	x-=y
*=	x *= y
/=	x /= y
%=	x %= y

Miscellaneous Operator:

The two Miscellaneous operators are

- conditional operator (? :)
- typeof operator.

Conditional Operator (? :)

The conditional operator first evaluates an expression for a true or false value and then executes one of the two given statements depending upon the result of the evaluation.

S.No	Operator and Description
1	**? : (Conditional)** If Condition is true? Then value X : Otherwise value Y

typeof operator:

- The **typeof** operator is a unary operator that is placed before its single operand, which can be of any type. Its value is a string indicating the data type of the operand.
- The *typeof* operator evaluates to "number", "string", or "boolean" if its operand is a number, string, or boolean value and returns true or false based on the evaluation.
- The list of the return values for the **typeof** Operator.

Type	**String Returned by typeof**
Number	"number"
String	"string"
Boolean	"boolean"
Object	"object"
Function	"function"
Undefined	"undefined"
Null	"object"

1.13 JavaScript Literals

String literals:

A **string literal** is zero or more characters, either enclosed in single quotation (') marks or double **quotation** (") marks. You can also use + operator to join strings. The following are the examples of string literals:

i. string1 = "w3resource.com"
ii. string1 = 'w3resource.com'
iii. string1 = "1000"

iv. string1 = "google" + ".com"

Array literals

In Javascript an **array literal** is a list of expressions, each of which represents an array element, enclosed in a pair of square brackets ' [ ] ' .

When an array is created using an array literal, it is initialized with the specified values as its elements, and its length is set to the number of arguments specified. If no value is supplied it creates an empty array with zero length.

Creating an empty array :

```
var fruits = [ ];
```

Creating an array with four elements

```
var fruits = ["Orange", "Apple", "Banana", "Mango"]
```

Integer literals:

An **integer** must have at least one digit (0-9).

- No comma or blanks are allowed within an integer.
- It does not contain any fractional part.
- It can be either positive or negative, if no sign precedes it is assumed to be positive.

Floating pointing literals:

A floating number has the following parts.

- A decimal integer.
- A decimal point ('.').
- A fraction.
- An exponent.

The exponent part is an "e" or "E" followed by an integer, which can be signed (preceded by "+" or "-").

Boolean literals:

The Boolean type has two literal values:

- true
- false

Object literals:

An **object literal** is zero or more pairs of comma separated list of property names and associated values, enclosed by a pair of curly braces.

In JavaScript an object literal is declared as follows:

1. An object literal without properties:

```
var userObject = {}
```

2. An object literal with a few properties :

```
var student = {
First-name : "Suresy",
Last-name : "Rayy",
Roll-No : 12
};
```

1.14 JavaScript Functions

- A function is a group of reusable code which can be called anywhere in your program. This eliminates the need of writing the same code again and again. It helps programmers in writing modular codes.
- Functions allow a programmer to divide a big program into a number of small and manageable functions.
- A JavaScript function is a block of code designed to perform a particular task.
- A JavaScript function is executed when "something" invokes it (calls it).

JavaScript Function Syntax:

- A JavaScript function is defined with the **function** keyword, followed by a **name**, followed by parentheses **()**.
- Function names can contain letters, digits, underscores, and dollar signs.
- The parentheses may include parameter names separated by commas:**(*parameter1, parameter2, ...*)**

The code to be executed, by the function, is placed inside curly brackets: **{}**

```
function name(parameter1, parameter2, parameter3)
 {
   code to be executed
 }
```

- Function **parameters** are the **names** listed in the function definition.
- Function **arguments** are the real **values** received by the function when it is invoked.
- Inside the function, the arguments behave as local variables.

Function Invocation:

The code inside the function will execute when "something" **invokes** (calls) the function:

- When an event occurs (when a user clicks a button)
- When it is invoked (called) from JavaScript code
- Automatically (self-invoked)

Function Return:

- When JavaScript reaches a **return statement**, the function will stop executing.
- If the function was invoked from a statement, JavaScript will "return" to execute the code after the invoking statement.
- Functions often compute a **return value**. The return value is "returned" back to the "caller":

Example:

Calculate the product of two numbers, and return the result:

```
var x = myFunction(4, 3);        // Function is called, return value will end up in x

function myFunction(a, b) {
   return a * b;              // Function returns the product of a and b
}
```

The result in x will be:

12

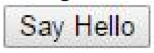
Calling Function:

To invoke a function somewhere later in the script, you would simply need to write the name of that function as shown in the following code.

```
<html>
  <head>
      <script type="text/javascript">
      function sayHello()
      {
        document.write ("Hello there!");
      }
    </script>
  </head>
  <body>
    <p>Click the following button to call the function</p>
      <form>
          <input type="button" onclick="sayHello()" value="Say Hello">
    </form>
        <p>Use different text in write method and then try...</p>
  </body>
</html>
```

Output:

```
    Click the following button to call the function
```



```
    Use different parameters inside the function and then try...
                Hello there!
```

Function Parameters:

There is a facility to pass different parameters while calling a function. These passed parameters can be captured inside the function and any manipulation can be done over those parameters. A function can take multiple parameters separated by comma.

Example:

```
    <html>
      <head>
        <script type="text/javascript">
          function sayHello(name, age)
          {
            document.write (name + " is " + age + " years old.");
          }
        </script>
       </head>
      <body>
        <p>Click the following button to call the function</p>
        <form>
          <input type="button" onclick="sayHello('Zara', 7)" value="Say Hello">
        </form>
        <p>Use different parameters inside the function and then try...</p>
      </body></html>
```

Output:

Click the following button to call the function

Say Hello

Use different parameters inside the function and then try...

Zara is 7 years old.

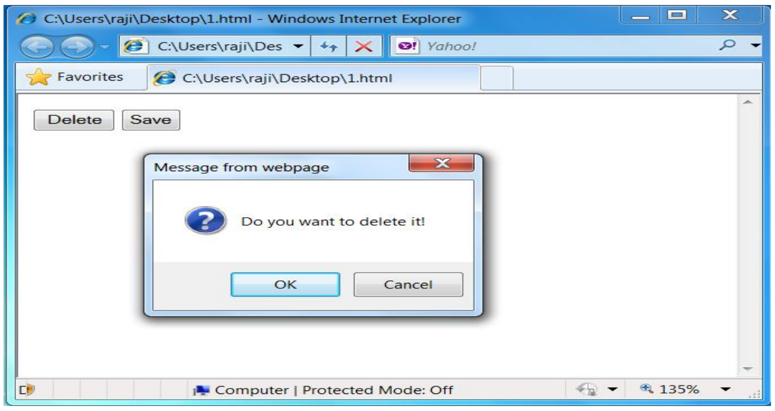
Function arguments:

Through variable you can pass argument to function. The output of the function looks on the arguments given by you.

Example:

```
<html>
<head>
  <script language="javascript">
        function myfunction(text)
        {
            confirm(text)
        }
 </script>
</head>
<body>
        <form>
                <input type="button" onclick="myfunction('Do you want to delete it!')"
value="Delete">
                <input type="button" onclick="myfunction('Do you want to save it!')"
value="Save">
        </form>
</body>
</html>
```

Output:


C:\Users\raji\Desktop\1.html - Windows Internet Explorer
C:\Users\raji\Des
Yahoo!
Favorites
C:\Users\raji\Desktop\1.html
Delete
Save
Message from webpage
OK
Cancel
Computer | Protected Mode: Off
135%

Do you want to delete it!

1.15 JavaScript objects

JavaScript is an Object Oriented Programming (OOP) language. A programming language can be called object-oriented if it provides four basic capabilities to developers

- **Encapsulation** − the capability to store related information, whether data or methods, together in an object.
- **Aggregation** − the capability to store one object inside another object.
- **Inheritance** − the capability of a class to rely upon another class (or number of classes) for some of its properties and methods.

- **Polymorphism** − the capability to write one function or method that works in a variety of different ways.

Objects are composed of attributes. If an attribute contains a function, it is considered to be a method of the object, otherwise the attribute is considered a property.

Object Properties:

The name:values pairs (in JavaScript objects) are called **properties**.

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

property	**Property Value**
firstName	John
lastName	Doe
age	50
eyeColor	blue

Object Methods

- Methods are **actions** that can be performed on objects.
- Methods are stored in properties as **function definitions**.

property	**Property Value**
firstName	John
lastName	Doe
age	50
eyeColor	blue
fullName	function() {return this.firstName + " " + this.lastName;}

Object Definition:

You define (and create) a JavaScript object with an object literal:

Example:

var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

Accessing Object Properties:

You can access object properties in two ways:

Syntax:

```
objectName.propertyName
        (or)
objectName["propertyName"]
```

Accessing Object Methods:

You access an object method with the following syntax:

Syntax:

```
objectName.methodName()
```

Example:

```
name = person.fullName();
```

The Object() Constructor:

A constructor is a function that creates and initializes an object. JavaScript provides a special constructor function called **Object()**to build the object. The return value of the **Object()** constructor is assigned to a variable.

The variable contains a reference to the new object. The properties assigned to the object are not variables and are not defined with the **var** keyword.

1.16 JavaScript Arrays

- An array is a special variable, which can hold more than one value at a time.
- JavaScript arrays are used to store multiple values in a single variable.

Creating an Array:

Using an array literal is the easiest way to create a JavaScript Array.

Syntax:

var *array-name* = [*item1*, *item2*, ...];

Example:

```
var cars = ["Saab", "Volvo", "BMW"];
```

Using the JavaScript Keyword new:

The following example also creates an Array, and assigns values to it:

Example:

```
var cars = new Array("Saab", "Volvo", "BMW");
```

Access the Elements of an Array:

An array element by referring to the **index number**.

This statement accesses the value of the first element in cars:

```
var name = cars[0];
```

Arrays are Objects:

- Arrays are a special type of objects. The **typeof** operator in JavaScript returns "object" for arrays.
- But, JavaScript arrays are best described as arrays.

Arrays use **numbers** to access its "elements". In this example, person[0] returns John:

Array:

```
var person = ["John", "Doe", 46];
```

Objects use **names** to access its "members". In this example, person.firstName returns John:

Object:

```
var person = {firstName:"John", lastName:"Doe", age:46};
```

Array Properties and Methods:

The real strength of JavaScript arrays are the built-in array properties and methods:

Examples:

```
var x = cars.length;        // The length property returns the number of elements in cars
var y = cars.sort();        // The sort() method sort cars in alphabetical order
```

The length Property:

The **length** property of an array returns the length of an array (the number of array elements).

Example:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.length;                      // the length of fruits is
 4
```

Adding Array Elements:

The easiest way to add a new element to an array is using the push method:

Example:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.push("Lemon");               // adds a new element (Lemon) to fruits
```

New element can also be added to an array using the length property:

Example:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits[fruits.length] = "Lemon";    // adds a new element (Lemon) to fruits
```

Adding elements with high indexes can create undefined "holes" in an array:

Example:

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits[10] = "Lemon";               // adds a new element (Lemon) to fruits
```

1.17 JavaScript Built-in Objects

(i) Math Object:

- The Math object allows you to perform mathematical tasks.
- The Math object includes several mathematical methods.

Example:

```
Math.min(0, 150, 30, 20, -8, -200);    // returns -200
```

String HTML Wrapper Methods:

The HTML wrapper methods return the string wrapped inside the appropriate HTML tag.

Method	Description
anchor()	Creates an anchor
big()	Displays a string using a big font
blink()	Displays a blinking string
bold()	Displays a string in bold
fixed()	Displays a string using a fixed-pitch font
fontcolor()	Displays a string using a specified color

fontsize()	Displays a string using a specified size
italics()	Displays a string in italic
link()	Displays a string as a hyperlink
small()	Displays a string using a small font
strike()	Displays a string with a strikethrough
sub()	Displays a string as subscript text
sup()	Displays a string as superscript text

(iii) Date object:

- JavaScript's Date object provides methods for date and time manipulations.
- Date objects are created with new Date().
- Date and time processing can be performed based on the computer's local time zone or based on World Time Standard's Coordinated Universal Time (UTC) - formerly called Greenwich Mean Time (GMT).

There are four ways of instantiating a date:

1. var d = new Date();
2. var d = new Date (milliseconds);
3. var d = new Date (dateString);
4. var d = new Date (year, month, day, hours, minutes, seconds, milliseconds);

Date Methods:

- When a Date object is created, a number of **methods** allow you to operate on it.
- Date methods allow you to get and set the year, month, day, hour, minute, second, and millisecond of objects, using either local time or UTC (universal, or GMT) time.

(iv) Boolean objects:

- JavaScript Boolean objetcs can have one of two values: true or false.
- These wrappers define methods and properties useful in manipulating Boolean values.

```
var b = new Boolean ( booleanValue );
```

(v) Number objects:

- JavaScript has only one type of number.
- Numbers can be written with, or without, decimals:

(vi) Array object:

- The Array object is used to store multiple values in a single variable:
- var cars = ["Saab", "Volvo", "BMW"];

(vii) Regular Expression object:

- A regular expression is an object that describes a pattern of characters.
- Regular expressions are used to perform pattern-matching and "search-and-replace" functions on text.

Syntax:

```
var patt=new RegExp(pattern,modifiers);
            (or)
var patt=/pattern/modifiers;
```