WEB BROWSERS:- * They are software programs that allow users to access web's Content.

* Documents provided by servers on the web are requested by browsers, which are programs running on client machines.

* The web operates in cliene- server configuration.

* The user can browser the resources available on servers using web browsers.

* The first browser was text- based - they were not capable of displaying graphic information, nor they had Graphical User Interface (GUI).

* In 1993, Mosaic the first GUI browser was released.

* Mosaic was developed at the National Center for Super computer Applications (NCSA) at the University of Illinois.

* Mosaic first ran on Unix systems then versions for Apple Macintosh, Microsoft Windows was released by late 1993.

* The result of Graphical browser provided explosive growth in web usage.

* A Browser is a client on the web because it initiates the communication with a server, which waits for a request from the client before doing anything.

* The Browser requests a static document from a server.

* The server locates the document among the several servable documents and sends it to the browser, which displays it for the user.

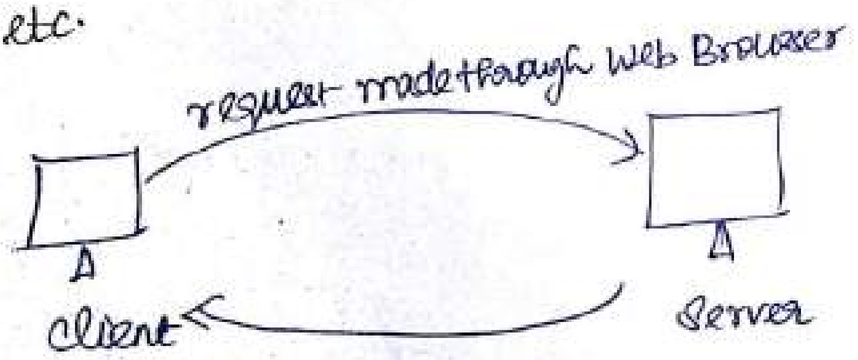
* Sometimes, some document will be displayed after inputting values by the user in the browser from the server.

* Although the web supports a variety of protocols, the most common one is the HTTP.

* HTTP provides a standard form of communication between browser & web server.

* The most commonly used web browsers are Microsoft Internet Explorer, Firefox, Chrome, Safari, Opera, etc.

(Windows)
various OS
(Macintosh)



Response received from web server in the client's web browser.

* Each browser differs in functionality, performance & features.
* The same web page may display differently in different Browser.

WEB SERVERS:

* They are programs that provides documents to requesting browsers.

* The first web server was hosted on Tim Berners-Lee's desktop computer.

* The most commonly used web servers are Apache (opensource) and Microsoft's Internet Information server (IIS).

Web server operation:

* Web Browsers initiate network communications with servers by sending them URLs, (Uniform Resource Locator) which is used to identify resources on the internet.

* All the communication between a web client and a web server use the standard web protocol, HTTP.

* When a web server begins execution, it informs the OS under which it is running that it is ready to accept incoming network connections through a specific port on the machine.

* A web client or a browser opens a network connection to a web server sends a information requests and possibly data to the server, receives information from the server and closes the connection.

* The primary task of web server is to monitor a communication port on its host machine, accept HTTP commands through that port, and performs operations specified by those commands.

* In between DNS, Network routers are present.

General Server Characteristics:-

* Most of the available servers share common characteristics, regardless of their origin or the platform on which they run.

* The file structure of a web server has two separate directories.

  (1) Document Root

  (2) Server Root.

* The file hierarchy that grows from the Document root stores the web documents to which the server has direct access and normally serves to clients through URL's.

* The server root directory along with its descendant directories, stores the server and its support software.

* The clients do not access the document root directly in URL's, rather server maps requested URLs to document root.

* Many servers allow part of the servable document collection to be stored outside the directory in the secondary areas at the document root called virtual document trees.

* Some servers support more than one site on a computer, such secondary hosts are called virtual hosts.

* Some servers can serve documents that are in the document root of other machines on the web called proxy servers.

* Web servers can support ftp, gopher, news, mailto.

* Web servers can interact with database systems through server-side scripts.

* The application stack to run a website include OS, web server software, Database software & Scripting software (LAMP).

SEARCH ENGINE:-

* Search engines are the primary tools which are used to find information on the web.

* Searches are performed with keywords or phrases in search engine and it finds matching web pages and shows the Search Engine results page (SERP) with recommended pages listed and sorted by relevance.

* People- assisted search engines, such as Kahalo pays people to develop search results.

* Vertical search engines focuses on specific type of topic or industry.
  Eg: Mocavo, yelp, Trulia, Library of Congress, Indeed, Monster.

* Traffic to the search engine is growing rapidly.

* Search engine helps users to quickly find and filter the information they want.

I) Google Search:

* It is the leading search engine founded by Larry page and Sergey Brin.

* Google search is based on Page Rank algorithm and its unique infrastructure of servers that uses to achieve faster responses.

* The page Rank algorithm considers the number of links into a web page and the quality of the linking sites to determine the importance of the page.

* Sites with the highest page rank appears at the top of the search results

* Google offers speciality search engines for images, news, videos, blogs, etc.

* Adwords - Advertising program. ads appear next to search results on the Google site.

* Adsense is Google's advertising program for publishers.

2) Yahoo:-

It was started in 1994 by Jerry Yang and David Filo, as a web directory rather than a search engine.

(3) MSN:-

* It was created in 1998 by Microsoft.

* It allows to search the web, performing specialized searches (news, images, etc).

* Location based search uses geographic information about the searcher to provide more relevant search results. Eg: To find Restaurants, ATM. (Google maps)

* Search engines can also be customized.

* Search Engine Optimization (SEO) is the process of designing + tuning website to maximize findability and improve rankings in search engine results.

* Two ways of employing SEO — White hat SEO (methods approved by SE to maximize findings)
— Black Hat (used to deceive search engines)

* Link Building is the process of increasing search engine rankings by generating inbound links to a particular website.

* The 3 methods for building links
  - Reciprocal linking
  - Link Baiting
  - Natural Linking.

* Reciprocal Linking is an exchange in which two related websites link to each other, increasing the link popularity of both sites and adding value for site users.

* Link Baiting involves creating attention-grabbing webcontent specifically for viral exposure through social media.

* Natural Linking is the process of building one-way inbound links by optimizing web content and user experience. This has higher weightage in page ranking.

* Search Engine marketing (SEM) is the method of promoting the website to increase traffic and search results by raising the site's visibility on search engine results pages.

* SEO is the most popular form of SEM.

* Search Engine watch is a search engine marketing resource site.

* SEWatch include search engine submission tips, web searching tips, search engine resources, etc.

* It includes articles, tutorials, conferences and more.

* Search Engine land - site provides news and information on major search engines.

* Search engine Strategies is a global conference series focused on search engine advertising.

HTTPS:- (Hyper Text Transfer Protocol Secure):

* It is the HTTP protocol running on top of the Transport Layer security (TLS)/ SSL (secure sockets Layer).

* using HTTPS, encryption to websites can be done.

* From a client's perspective to check whether a site is secured, we can look at the URL in which a little padlock icon is present. Eg: 🔒 https:// mail. gogle. com

Secure Handshake:-

* The client initiates the handshake by sending the time, and a list of cipher suites (set of cryptographic algorithms) its browser support to the server.

* The server in response sends back which of the client's ciphers it wants to use as well as a certificate, which contains information including a public key.

* The client can then verify if the certificate is valid.

* Then the client can send a premaster secret (encrypted with public key received from the server) back to the server.

* using the random premaster secret both client & server can compute a symmetric key.

* After a brief client message and server message declaring their readiness, all transmission begins to be encrypted with the agreed symmetric key.

Certificates:

* The certificate that is transmitted during handshake is

called X.509 certificate.

* The X.509 certificate contains details like algorithm used, domain it was issued for, and some public key information.

* It also contains a signature mechanism which can be used to validate that the domain is really who they claim tobe.
* This signature relies on third party to sign the certificate on behalf of the website so that to trust the website.

Certificate Authority:-

* It allows users to place their trust in the certificate since a trusted independent third party signs it.
* It validates the requestor of the Certificate is who they claim to be, and issue and sign the certificate containing the public keys so that everyone trust they are genuine.
* Signed certificate is essential for any website that process payment, booking, etc. CA Examples are Go Daddy Gray, Verisign, VISA, etc.

Self- Signed Certificates:

* An alternative to pay a certificate Authority is to sign the certificates yourself.
* Self- Signed certificates provide same level of encryption, but the validity of the server is not confirmed and are useful for development and testing environments.
* Most Browsers will warn users that the site is not completely secure.

#Website Architecture:-

- It is the planning and design of the technical, functional and visual components of a website before it is designed, developed & deployed.
- It is used by website designers and developers as a means to design and develop a website.
- The website architecture includes (1) usability (2) Interaction Design (3) User-Interface Design (4) Information Design (5) Web design (6) Graphic design (7) Content Strategy.

Creating a website Architecture plan:

- Developing the architecture for a website is an essential part of the web design process. The steps involved are:

Framing the approach
Understand the business goals
Understand user
Create personas
Gather Content
Run the application.

- Framing the approach gives the thorough understanding of the current state before moving on to anything else, if its a existing website.
- Understanding the business goals gives a firm understanding of the goals of business to make sound decisions.

* Engaging with users and conducting research to understand their needs, motivations, etc.

* Once user research is completed, translate those findings into usable format through personas.

* Before designing, gather and assemble all the content to ensure it is current, accurate and consistent.

* Run the application that is designed.

Factors:

The factors that are part of website architecture are:

(1) Technical constraints such as server storage, memory and communication interfaces.

2) Functional aspects such as type of services or processes the website will provide.

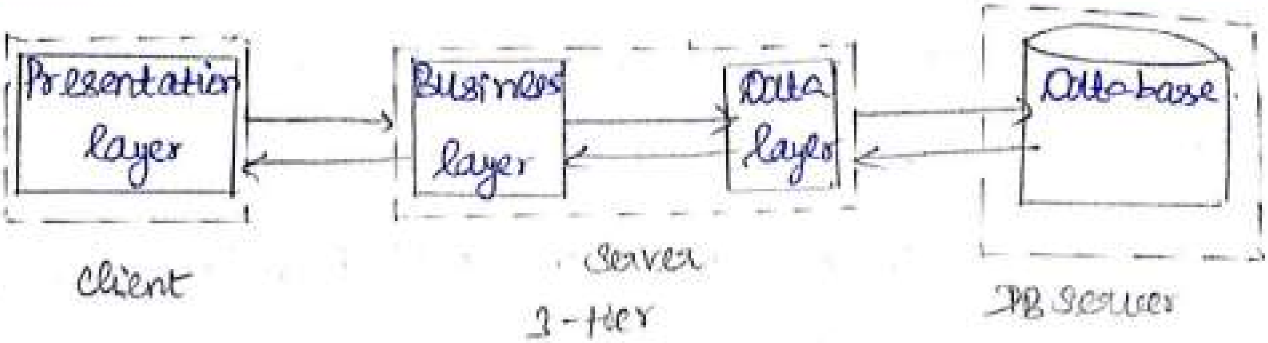
(3) Visual appearance i.e) user interface, colors, buttons and other visual design elements.

(4) Security parameters i.e) how the website will ensure secure access control and transactions.

* The web application can have 3-tier architecture or MVC architecture.

3tier architecture:

* The 3-tier architecture for web apps have presentation layer, Business layer and data layer.



3-tier

* Presentation Layer provides user interface, it handles the interaction with the user called GUI or front end. (HTML)

* Business or logic layer has set of rules for processing information sometimes called middleware or back end processing layer. (PHP)

* Data layer manage access to Database (ie) storing & retrieving information from database. (RDBMS)

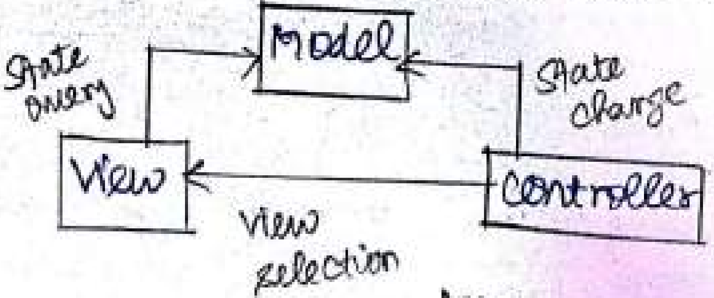
MVC Architecture:-

* Model contains domain specific knowledge. It records the state of the application and often linked to database.
Eg: What items are in a shopping cart. (DB)

* View presents data to the user & allows user interaction. (HTML, CSS)

* Controller defines how user interface reacts to user input (events), receive messages from view, sends messages to model. (Client side scripting)

* MVC is easier to construct and maintain than 3-tier.



MVC

Security:

* The Internet and web are fertile grounds for security problems.

* One aspect of web security is getting one's data from the browser to the server and having the server deliver data back to the browser without intercepting or corrupting the data along the way.

Example scenario:

* Consider a simple case of transmitting a credit card number to a company from which a purchase is being made. The security issues for this transactions are as follows

1. Privacy: Credit card number can be stolen on its way to the company's server.

2. Integrity: Credit card number must not be modofied on its way.

3. Authentication: Purchaser & seller has to be certain about each other's identity.

4 Non- Repudiation: Legally the message is sent and received has to be proved.

* The basic tool to support privacy and integrity is Encryption.

* Data to be transmitted is converted into different or encrypted form such that someone who is not suppose to access the data cannot decrypt it.

* Encryption and Decryptionare done with the same key or public- private key pair.

* Diffie Hellman key exchange and RSA are the public key algorithm that can be used.

* Denial of service attacks is created by flooding a web server with requests, overwhelming its ability to operate effectively.

* Virus are programe that arrive as an attachment in e-mail or attached to free downloaded programs.

* They replicate with attached device and destroys programs and data.

* Worms damage memory but spread on their own, rather than being attached to other files. The most famous worm is Blaster worm in 2003.

* DOS, virus & worm attacks are created by malicious people called hackers.

* Protection against virus & worms is provided by Anti-Virus software, and it must be updated frequently to protect against continuous stream of new viruses and worms.

Search Engine Index:

It collects, parses, and stores data to facilitate fast and accurate information retrieval.

Indexing:

The purpose of storing an index is to optimize speed and performance in finding relevant documents for a search query. Without an index, the search engine would scan every document in the corpus (large set of texts) which would require considerable time and computing power.

Span & div tags:

* Span element is inline and usually for a small chunk of HTML inside a line whereas div (division) element is blockline and used to group larger chunks of code.

Example:

```
<div class = "First">
    <p> India </p>
</div>
```

```
<p> India <span class="second"> Pakistan </span></p>

        p. First { color: red; }

        p. second { color: pink; }
```

*#CSS (CASCADING STYLE SHEET)

* CSS is used to define the presentation of HTML documents.

* A style sheet is a syntactic mechanism for specifying style information.

* It can be added directly to any HTML element within a <head> element, or in a separate text file that contains only CSS.

Benefits of CSS:-

(1) Improved Control over formatting

(2) Improved site maintainability

(3) Improved accessibility

(4) Improved page download speed.

(5) Improved output flexibility (Responsive design)

CSS Versions:-

CSS Level 1, CSS Level 2, CSS 2.1, CSS 3 are the various levels of versions in CSS.

CSS Syntax:-

* A CSS document consists of one or more stylerules.

* A rule consists of a selector that identifies the HTML element or elements that will be affected, followed by a series of property: value pairs.

$$\text{selector } \{ \underbrace{\text{property : value}}_{\text{declaration}} ; \text{ property 2 : value 2 ; } \} \rightarrow \text{Rule}$$

General css syntax.

Example:

```
  selector
em { color : red ; }
     property  value

selector
   P {
       font-weight : bold ;            Value
property  font-family : Arial, Helvetica, sansserif ;
   }
```

(1) Selectors :-

* Every CSS rules begins with a selector.

* The selector identifies which element in the HTML document will be affected by the declarations in the rule.

2) Properties :-

* Each individual css declaration must contain a property.

* The property names are predefined by the CSS standard.

* The CSS2.1 defines over a 100 different property names.

* The most commonly used CSS properties are:

Fonts	font font- family font - size font - style font - weight font - face
Text	letter- spacing line - height text- align text - decoration text - indent
Color+ Background	background background- color background - image background - position background - repeat color.

* The others are borders, spacing, sizing, layout and lists. [Refer text book for other properties].

(3) Values:

* Each CSS declaration also contains a value for a property.

* The unit of any given value is dependent upon the property.

* For Eg: Property color has values like Name, RGB, Hexadecimal, HSL.

```
Color : red;                     Color : #FF0000;

Color : rgb(255,0,0);            Color : hsl(0,100%,100%);
                                 hue, saturation & light
```

* There are multiple ways of specifying a measurements. They are

px	-	pixel (Relative)
em	-	Equal to the value of the font-size property of the element. (Relative)
%	-	A measure that is always related to another value (Relative)
in	-	inches (Absolute)
cm	-	Centimeters (Absolute)
mm	-	millimeters (Absolute)
pt	-	points (equal to 1/72 of an inch) (Absolute)
pc	-	pica (equal to 1/6 of an inch) (Absolute)

#Location of styles(or) style specification formats (or) levels of stylesheet:

* CSS style rules can be located in 3 different locations.

(1) Inline stylesheet

(2) Embedded style sheet (Internal styles)

(3) External style sheet

(1) Inline style sheet:-

* They are the style rules placed within an HTML element, via the style attribute.

* It only affects the element it is defined within.

* Selectors are not necessary and semicolons are only required for separating multyple rules.

* They can be used to quickly test out a style change.

* They increase bandwidth and decrease maintainability, because presentation & content are intermixed so it becomes difficult to make consistent inline style changes across multiple files.

Example: <h1> Share your travels</h1>

<h2 style = " font-size : 24pt"> Description</h2>

<h3 style = " font-size : 24pt ; font-weight : bold ;">READY</h3>

General syntax:

Style = "property_1 : value_1 ; property_2 : value_2;"

(*) Embedded style sheet:-

* These are the style rules that are placed within the style element and placed inside the <head> tag.

* It is used to quickly test out a style that is used in multiple places within a single HTML document.

* Each HTML document has its own <style> element, so it is difficult to consistently style multiple documents when using embedded styles. When Embedded style is used it applies to the whole body of the document.

Eg:-

```
<html>
<head>
 <title>   New york </title>
<style type = "text/css" > (or) <style>
   h1 { font- size : 24pt ; }
   h2 {
       font- size :18pt;
       font- weight : bold;
      }
 </style>
 </head>
<body>
   <h1>  share your travels </h1>
   <h2>  NewYork - centralpark </h2> </body>
</html>
```

General syntax:-

```
<style type="text/css">
 rule-list
</style>
```

(3) External Style sheet:

* They are the style rules placed within an external text file with the .css extension. When styles are to be applied to more than one document External styles are used.

* When changes are made to the external style sheet, all HTML documents that reference that style sheet will automatically get updated.

* It provides better maintainability because CSS & markup are separated.

* To reference an external style sheet <link> element is used.

* Several link sheets can be linked at a time.

Eg:

```
<head>
<title>   Central park </title>
<link   rel = " Stylesheet"   href = "Style.css"/>
</head>
```

Selector Forms:-

A selector specifies the element to which the following style information applies. The selector can have a variety of forms.

(1) Simple Selector Forms:-

* It is a single element name, such as h1. The property values in the rule apply to all occurrences of the named element.

eg:

```
h1 { color : red;}
```

(2) Class selectors:-

* It allows different ocurences of the same element to use different style specifications.

```
<html>
<head>
<style>
p. normal { color: red ;}
p. first { color: pink;}
</style>
<body>
<p class = "normal" > India </p>
<p Class = "first" > Russia </p>
</body>
</html>
```

(3) Generic selectors:-

* If a class of style specifications applies to the content of more than one kind of element, then it is generic selector.

* NO element name in its selector are given rather generic class is given.

```
<html>
<head>
<style>
 ~~P~~ . first {color : red ;}
</style>
<body>
<h3 class = "first" > America </h3>
<p class = " first" > India </p>
<p class = " first"> Russia </p>
</body>
</html>
```

(4) ID selector:-

* It allows application of a style to one specific element.

* The element with a given ID applies the style.

```
<h2 id= "cat" > This is my pen </h2>
```

```
# cat { color : Brown;} .
```

(5) Contextual selector:-

* If the style needs to be applied only to elements in certain positions in the document, this selector can be used.

Selector	Matches	Example
Descendant	A specified element that is contained somewhere within another specified element.	div p
child	A specified element that is a direct child of the specified element.	div > h2.
Adjacent sibling	A specified element that is the next sibling of the specified element.	h3 +p.
General sibling	A specified element that shares the same parent as the specified element.	h3 ~p

Example:

```
#main p { color: red;}
  <div id= "main" >
        <p> This is my BOOK </p>
  </div>
```

(b) Pseudo - class selectors:-

* It specifies that certain style applies when something happens, rather than because the target element simply exists.

* The most common use of this is for targeting links.

Selector	Type	Description
a: link	Pseudo- class	selects links that have not been visited.
a: visited	Pseudo- Class	selects elements that have been visited.
: focus	Pseudo- class	selects elements that have the input focus. (text or list box)
: active	Pseudo- class	selects an element that is being activated by the user.
: hover	Pseudo- Class	selects elements that the mouse pointer is currently above.
: checked	Pseudo- class	selects a form element that is currently checked. (Radio button or Check box)
: first-letter, : first - line	Pseudo - element	selecting first letter & line of an element

Example:

Styling a link using Pseudoclass selector:

```
<html>
<head><title>share your travels</title>
 <style>
   a: link {
    text - decoration : underline;
     color : blue;
   }
  a: visited {
   text - decoration : underline;
    color: purple;
   }
  a: hover {
   text - decoration : none;
   font - weight : bold;
   }
 </style>
</head>
 <body>
   <p><a href="#">Canada </a>
   <p> <a href="#"> Germany </a>
   </body>
   </html>
```

(7) universal selector(or) Element selector:-

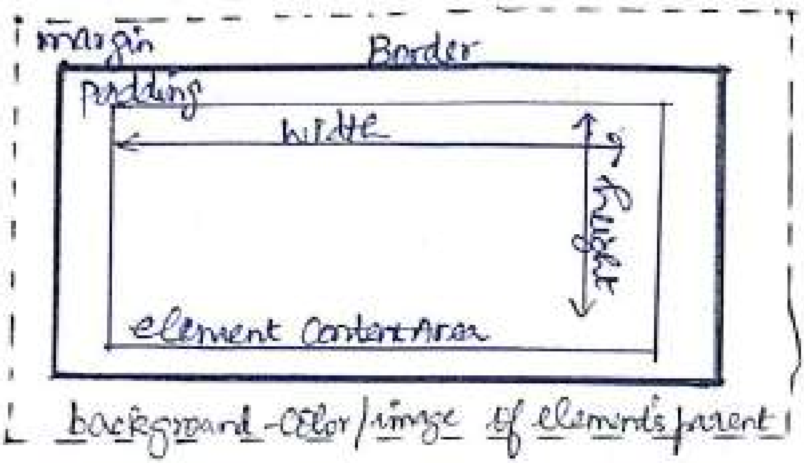
It selects all instances of a given HTML element. It is done by using the asterisk (*) symbol. or the group elements can be selected by separating the different element names with commas.

Eg:

```
* { color : red ;}
P, div, h1 { color:red;}
```

Box Model:

* In CSS, all the elements exists within an element box.



CSS Box Model

* The amount of space between the content of an element and its border, is padding.

* The space between the border and an adjacent element known as margin.

Background:

The background color or image of an element fills an element out to its border.

Property	Description:
background	A combined short-hand property to give multiple background values.
background- attachment	It specifies whether the background image scrolls with the document or

	remains fixed. values are fixed, scroll.
background -color	sets the background color of the element.
background- image	specifies the background image for the element.
background- position	specifies where on the element the background image will be placed. Some possible values include bottom, center, left & right. Numeric position can be given.
background- repeat	Determines whether the background image will be repeated. Possible values are repeat, repeat-x, repeat-y and no-repeat.
background- size	modify size of the Background image.

Borders:

* It provides a way to visually separate elements. Every element has the border - style property, which controls whether the elements content has a border and also specifies the style of the border.

* CSS has several border styles such as

dotted , dashed, solid and double. Some of the CSS properties are:

Property:	value:
border Eg: {border: dotted 1pt green;}	A shorthand property allows to set the style, width & color of a border in one property. The order should be border-style border-width border-color.
border-style	Specifies the line type of the border. Possible values are solid, dotted, dashed, double, groove, ridge, inset & outset.
border-width border-top-width border-bottom-width border-right-width border-left-width	The width of the border in a unit. Border-width values are thin, medium, thick.
border-color. border-top-color, border-bottom-color, border-left-color, border-right-color	The color of the Border in a Color unit.
border-radius	The radius of a rounded-corner
border-image	The URL of an image to use as a border.

Margins & Padding:

* When there is no border, the margin plus the padding is the space between the content of an element and its neighbors.

* The margin properties are: margin-left, margin-right, margin-top & margin-bottom.

* The padding properties are: padding-left, padding-right, padding-top & padding-bottom.

Eg:

```
p
  {
    border: solid 1pt red;
    margin: 0;
    padding: 0;
    }
```

```
p
 {
border: solid 1pt red;
margin: 30px;
padding: 0;
 }
```

```
p {
  border: solid 1pt red;
  margin: 30px;
  padding: 30px;
  }
```

Font - properties:-

* These are the most commonly used style-sheet properties. It includes properties like:

(1) Font-family:-

* It is used to specify a list of font names.

Eg: font - family : Arial, Helvetica, Futura, sans-serif

If a browser is not able to support Arial, it uses Helvetica or futura. Generic fonts can be mentioned.

Generic name:	Examples:
Serif	TimesNew Roman, Garamond
Sans- serif	Arial, Helvetica.
monospace	Courier, Prestige.

(2) Font - sizes:-

* Its value specifies the size of the font. Two categories of values are absolute & Relative.

* In absolute, the size value can be given as points, picas or pixels or as a keyword from the list xx- small, x-small, small, medium, large, x-large and xx- large. The size of the keywords are different on browsers.

* The relative size values are smaller, larger, and percentage

* Percentages and em are good choices for setting font sizes.

Eg: font - size : 1.2 em

(3) <u>Font - variant:</u>

* The default value of the font - variant property is <u>normal</u>, which specifies the usual character font.

* It can be set to <u>small- caps</u> to specify small capital letters.

Eg: font - variant : small- caps

(4) <u>Font - style:</u>

* It specifies the font style for text. It has 3 values (ie) <u>normal</u>, <u>italic</u> and <u>oblique</u>.
↳ default.

Eg: font - style : italic

(5) <u>Font - weights:</u>

* It is used to specify the degree of boldness in text.

* The possible values are <u>normal, bold, bolder &amp; lighter</u>. The bolder &amp; lighter values are taken as relative to the level of boldness of the parent element. It can have a value between 100 &amp; 900.

Eg: Font - weight : bold;

(6) Font - short hands:-

* If more than one font property must be specified the values can be stated via list as the value of the font property.

Ex: font : bold 1.1em 'Times New Roman'

HTML program:

```
<html lang="en">
<head> <title> Font properties </title>
<meta charset = "UTF-8" />
<style>
  p. major { font-size : 1.1em;  font-style : italic;
                          font - family : 'Times New Roman'; }
  p. minor { font : bold 0.9em 'Courier New'; }
  h2 { font - family : 'Times New Roman'; font-size : 2em;
                 font - weight : bold; }
  h3 { font - family : 'Courier New'; font-size : 1.5em; }
 </style>
</head>
<body> <p class = "major"> India </p>
 <p class = "minor"> Srilanka </p>
 <h2> Portugal </h2>
 <h3> France </h3>
</body>
</html>
```

#Text Properties(or) paragraph properties.

* There are a range of properties that affect text independently of the font. They include.

(1) Text- alignment:-

The text- align property aligns the text horizontally in a container element. Possible values are left, right, Center and Justify.

```
Eg: h1 {text - align: Center;}
```

(2) Text- Color:

The color property is used to set the color of a text.

```
Eg:     body { color: "#ff0000"; }
```

(3) Text- decoration:

It specifies whether the text will have lines below, through or over it. Possible values are none, overline, line- through, underline and blink.

```
Eg: p {text - decoration : underline }
```

(4) Text- direction:

It specifies the direction to the text, left to right (ltr) or right to left (rtl).

Eg:
```
P{ text-direction: ltr;}
```

(5) Text- indent:

* It is used to specify the indentation of the first line of a text by a specific amount.

Eg:
```
P{ text- indent : 50px;}
```

(6) Text- shadow:

It adds a shadow effect toa text.

Eg:
```
h1 { text- shadow: 3px  2px  red;}
                   x axis y axis color
```

(7) Text- Transform:

* It changes the capitalization of text. Possible values are none, capitalize, lowercase and uppercase.

Eg:
```
P { text- transform: Capitalize;}
```

(8) Letter spacing:

It adjusts the space between letters. It can be normal or length unit.

Eg:
```
h2 { letter- spacing: 3px;}
```

(IX) Line - height:

It specifies the space between baselines. Default value is normal, and it can be set to any length unit.

Eg: p { line- height : 0.8 ; }

(X) Word - Spacing:

It adjusts the space between words. It can be normal or a length unit.

Eg: h1 { word - spacing : 10px; }

(XI) Vertical- align:

It aligns the text vertically in a container element. Most common values are: top, bottom, middle.

Eg: h1 { vertical- align: bottom; }

(XII) White- Space:

It specifies how a white- space inside an element is handled.

Scanned by CamScanner

MIME (Multipurpose Internet Mail Extension):-

* It is an Internet standard that extends the format of e-mail to support.

* It allows to exchange different kinds of data files on the internet: audio, video, images, ASCII text, etc.

* MIME consists of 3 basic pieces. (1) header line (2) Set of content type (3) Encode

* The header line has the data being carried in the message body.

* They include MIME -VERSION, Content - Description, Content type, Content - Transfer - Encoding.

* The content type for images image / gif and image/jpeg are supported and for text is text/plain and text/richtext. markuptext.

* It also defines an application type such as application/ postscript and application/ msword.

* MIME also defines a multipart type that sayshow a message carrying more than one data type is structured. For Eg: mixed.

* MIME uses encoding of binary data into the ASCII character set called Base64.

* The idea is to map every 3 bytes of the original binary data into 4 ASCII Characters.

* A message containing plaintext, jpg image, and a post script file will look like this.

MIME-Version : 1.0

Content-Type : multipart/mixed;

boundary = " ---- 417CA6E2DE4ABCAFBC5"

From: Alice <Alice@ cisco.com>

TO: Bob@cs.princeton.edu

Subject: promised material

plaintext

Date: Mon, 07 Dec 2018 19:45:19 -0400

------ 417CA6E2DE4ABCAFBC5

Content-Type : text/plain ; Charset = us-ascii

Content-transfer-Encoding : 7 bit

Bob,

jpg image

-- Alice

---- 417CA6E2DE4ABCAFBC5

jpg image

Content-type : image/jpg

Content-transfer-Encoding : base64

--- unreadable encoding of a jpg figure

--- 417CA6E2DE4ABCAFBC5

Content-type: application/postscript; name="draft.ps"

Content-Transfer-Encoding: 7bit

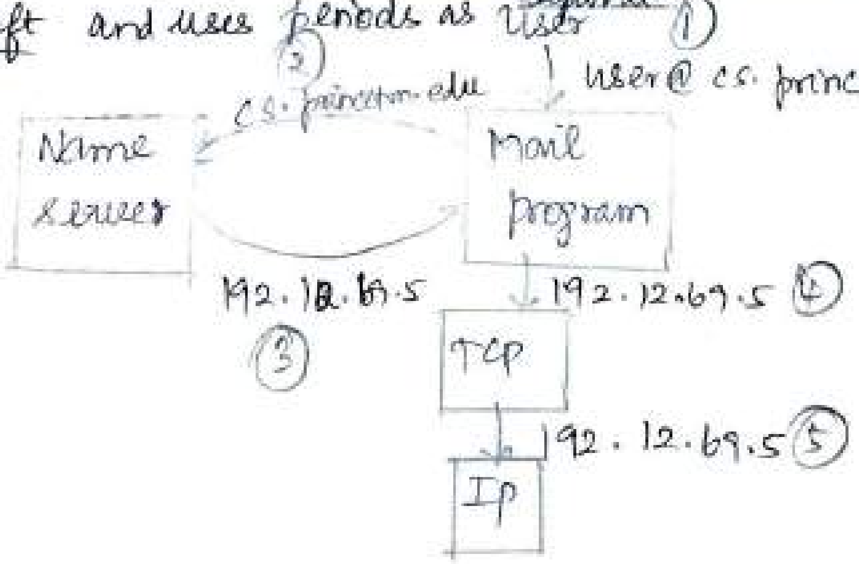
post-script file: readable encoding of a postscript document.

* Each type has its own content-type and Content transfer encoding lines.

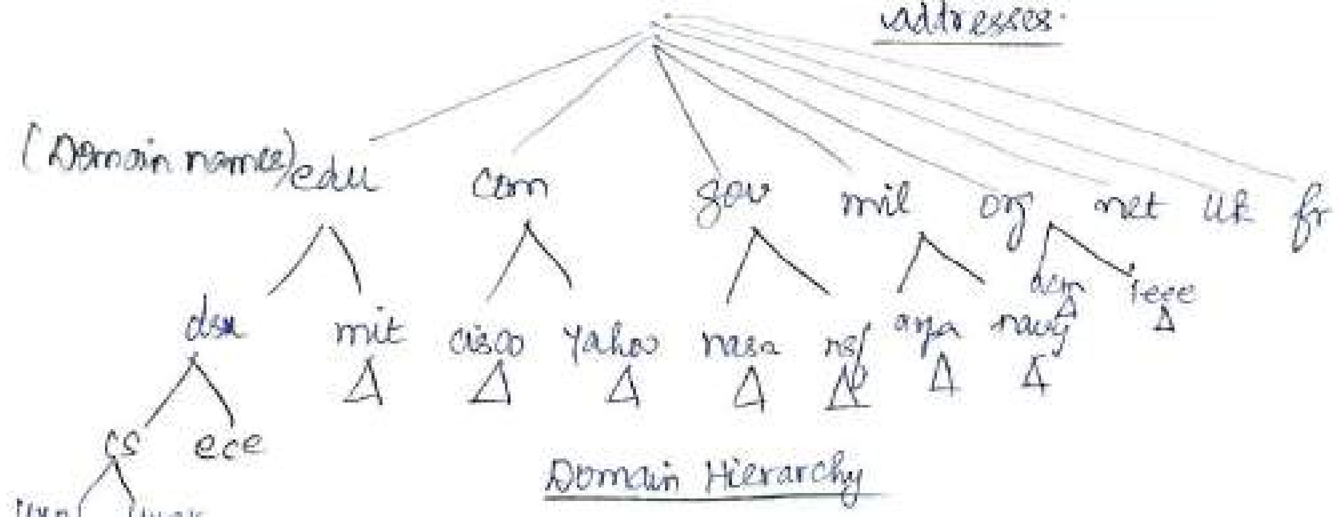
DNS (Domain Name system):

* A directory service that translates hostnames to IP addresses.

* DNS employs a hierarchical name space processed from right to left and uses periods as separator.



Sequence of steps to translate hostname to addresses.



Domain Hierarchy

Eg: Cicada. cs. princeton. edu

DNS Servers:-

* There are 3 types of DNS hierarchy servers. (1) Root DNS servers (2) Top-level Domain (TLD) (3) Authoritative

* In the Internet there are 13 root DNS servers.

* TLD servers are responsible for top-level domains such as com, org, net, edu, gov, etc.

* Authoritative DNS server has publicly accessible DNS records.

* Local DNS server such as servers present in university are also another important type of DNS server.

Working of DNS:-

*(1) The application will invoke the client side of DNS, specifying the hostname that needs to be translated.

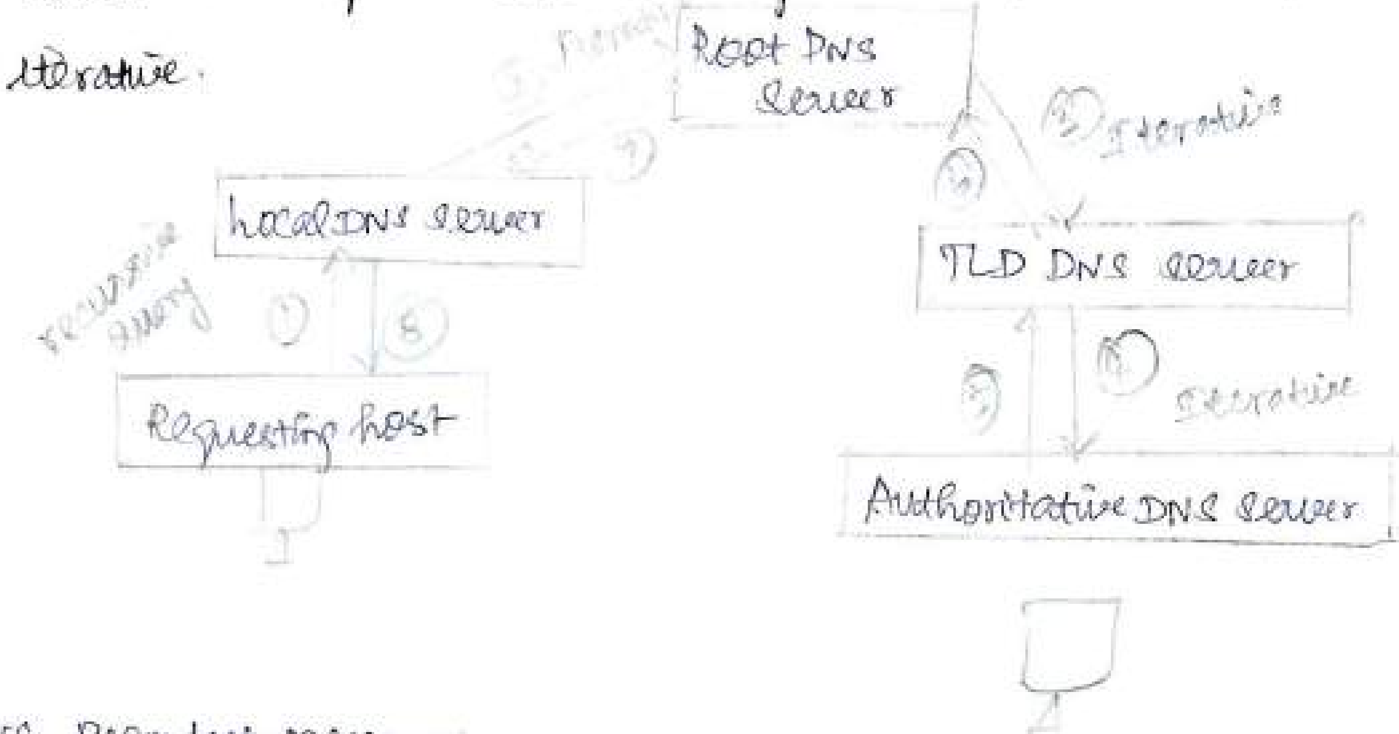
*(2) DNS in the user's host then takes over, sending a Query message into the network.

*(3) DNS in the user's host receives a DNS reply message that provides desired mapping. then it is passed into invoking application.

• There are two types of Queries: recursive & Iterative

* When the Query is sent to obtain mapping on someone behalf it is recursive.

* When the replies are directly returned, it is called iterative.



DNS Records of messages:

* Each name server implements the zone information as a collection of resource records.

* A resource-record is a name to value binding, a 5 tuple that contains the following fields.

  < Name, Value, Type, Class, TTL >

* The DNS message has various field such as

  (1) Header section (2) Question section (3) Answer section
  (4) Authority section (5) Additional section.

* DNS is a decentralized naming system for Computers connected to the Internet.

* DNS uses User Datagram Protocol (UDP) on port 53 to serve queries.

Principles / Needs of Application Layer protocols:-

* The application layer is the topmost layer used for establishing process to process communication.

* A particular networked application might need to make use of a reliable transport protocol, authentication and privacy mechanisms.

* Applications need their own protocols to perform their function.

* Application programs & Application protocols are (Chrome, [illegible]) (www) different.

* To develop an network application layer, the architecture should be designed. It may be client-server, P2P (Peer to peer) can be used.

* Interface between the process & computer network is done and for the process to communicate it should exchange messages.

* The data transfer should be reliable, and application layer relies on transport layer protocols such as TCP & UDP.

* For Example, SMTP exchange e-mail, HTTP used to communicate between web browser & web server.

WWW & HTTP:

* HTTP is web's application layer protocol, is at heart of web and defines structure of messages.

* The original goal of the web is to find a way to organize + retrieve information

* The hypertext is that one document can link to another document, and the protocols HTTP and document language HTML were used for this purpose.

* Web Browser allows the user to obtain an object by opening an URL. For Example,

http:// www.cs.princeton.edu/index.html

* If the above URL is opened, web Browser would open a TCP connection to the web server at a machine called www.cs.princeton.edu and immeditely retrieve and display the file called index.html.

* HTTP is a text-oriented protocol. It is a request/response protocol, where every message has the general form

```
START_LINE <CRLF>
MESSAGE_HEADER <CRLF>
<CRLF>
MESSAGE_BODY <CRLF>
```

<CRLF> stands for Carriage Return + Line Feed.

* Start_line indicates whether it is a request message or a response message.

In specific Start_Line identifies the remote procedure to be executed when it is a request message or the status of the request when it is a response message.

* The Message_Header specifies header types followed by blank line.

* Message_Body is where a server would place the requested page when responding to a request.

* HTTP runs over TCP since to provide reliable delivery, flow control & congestion control although it has few issues.

Request messages:

* The HTTP request message specifies 2 things:

(1) Operation to be performed

(2) Web page the operation should be performed.

(3) Version of HTTP being used.

* The two most common operations are GET & POST.

* The other operations are

OPTIONS Request information about available options.

GET Retrieve document identified in URL.

HEAD Retrieve metainformation about document identified in URL.

POST Give information to server.

PUT Store document under specified URL.

DELETE Delete specified URL.

TRACE . Loopback request message

CONNECT For use by proxies.

For Example, $\overset{\text{Request method}}{\underline{\text{GET}}}$ http://$\underbrace{\text{www.cs.princeton.edu}}_{\text{Host}}$/index.html

START_LINE $\underbrace{\text{HTTP/1.1}}_{\text{(HTTP Version)}}$

MESSAGE_HEADER GET index.html HTTP/1.1

Host : www.cs.princeton.edu

Response Messages:

* It begins with a single start_line.

* It has a three-digit code indicating whether or not the request was successful, and a text string giving the reason for the response.

For Eg:

START_LINE $\underbrace{\text{HTTP/1.1}}_{\text{Version}}$ $\underbrace{\text{202}}_{\text{Response code}}$ $\underbrace{\text{Accepted}}_{\text{Text string}}$

MESSAGE_HEADER HTTP/1.1 404 not found

Location: http://www.princeton.edu/cs/index.html

* The response message will also carry the requested page.

* Five types of HTTP result codes.

Code	Type	Example Reasons
1xx	Informational	Request received, continuing process.
2xx	Success	action successfully received, understood and accepted.
3xx	Redirection	further action must be taken to complete the request
4xx	Client Error	request contains bad syntax or cannot be fulfilled.
5xx	Server Error	server failed to fulfill an apparently valid request.

Uniform Resource Identifiers (URI):-

* The URL's that HTTP uses as addresses are one type of URI.

* A URI is a character string that identifies a resource, where a resource can be anything that has identity, such as a document, an image, or a service.

* It has two parts : (1) scheme (2) scheme - specific part

Eg: file : /// C :/ foo.html

scheme

scheme - specific part.

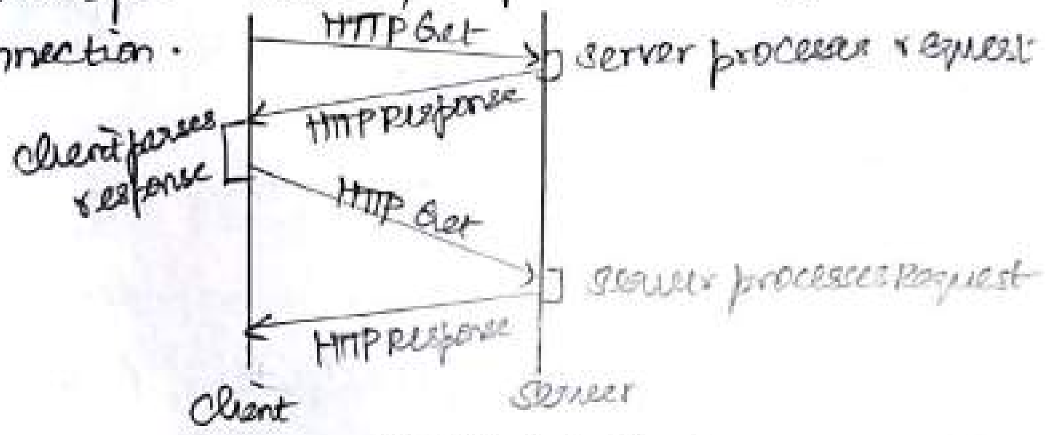
* URL is different from URI because URI is not a locator they provide a global unique identifier for the name space.

* The response m...

TCP connections:

* HTTP (1.0) → Non- persistent connections establishes a separate TCP connection for each data item retrieved from the server. It is inefficient.

* HTTP (1.1) → persistent connections, the client & server can exchange multiple request/response messages over the same TCP connection.



HTTP 1.1 behavior

Caching:

* Caching web pages has many benefits, and it is done by HTTP.

* For example, a user's browser can cache recently accessed pages and display the cached copy if the user visits the same page again, so that it reduces the load on the server.

* There are also a set of cache directives that must be obeyed by all caching mechanisms along the request/response chain.

* The default port number of HTTP is 80.

* A web cache also called proxy server - is a network entity that satisfies HTTP requests on behalf of web server.

#FTP (File Transfer Protocol):

* It is a standard network protocol used for the transfer of computer files between a client and server on a computer network.

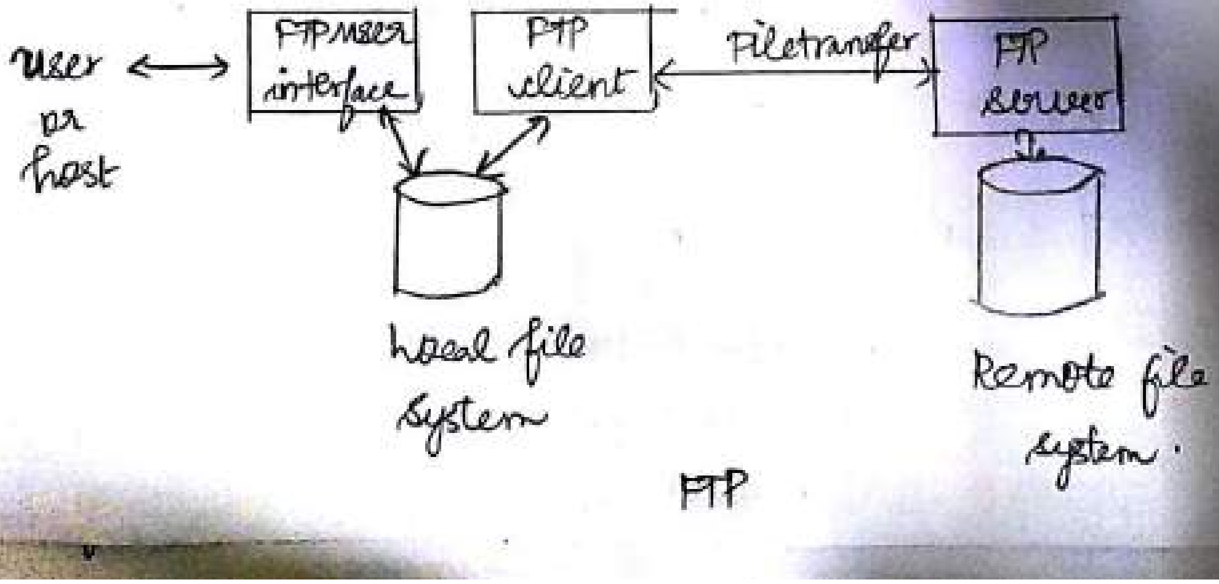
* It is built on a client - server model architecture using separate control and data connections between client and server.

* The user interacts with FTP through an FTP user agent.

*The user first provides the hostname of the remote host, causing the FTP client process in the local host to establish a TCP connection with the FTP server process in the remote host.

* The user then provides the user identification+ password.

* Once the server has authorized the user, the user copies one or more files stored in the local file system into the remote file system.



FTP

FTP (File Transfer Protocol):

④

* It is a standard network protocol used for the

* HTTP and FTP are both file transfer protocols and have common characteristics; like both run on top of TCP.

* The difference is that FTP uses two parallel TCP connections to transfer a file, a control connection and a data connection.

* The control connection is used for sending control information between the two hosts such as user identification, password, and commands like get+put files.

* The data connection is used to send a file.

* The FTP send its control information out of band and HTTP sends its control information in-band.

For Example,

* When a user starts an FTP session with a remote host, the client side of FTP first initiates a control TCP connection on the server port Number 21, with the server side.

* The client side of FTP sends the user Identification and password over this control connection.

* When the server side receives a command for a file transfer over the control connection, the server side initiates a data connection on ~~the~~ port number 20 and then closes the data connection.

Scanned by CamScanner

* New data connection is created for each file transfered within session.

* Throughout a session, the FTP server must maintain the state about the users whereas HTTP is stateless - it does not keep track of any user state.

FTP commands & Replies:

* The commands and replies are sent across the control connection in 7 bit ASCII format.

* FTP commands are readable by people and a carriage return & line feed is present at the end of each command to delineate successive commands. Some of the commands are:

USER username: used to send user identification to the server.

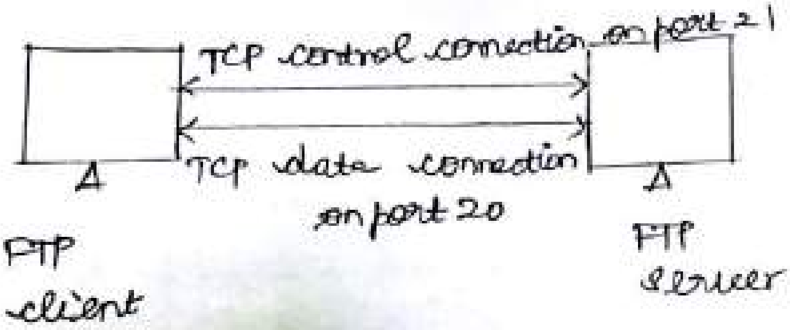
PASS password: used to send the user password to the server.

LIST: used to ask server to send list of files over a data connection in the current remote directory.

RETR filename: Used to retrieve a file from the current remote directory.

STOR filename: used to store a file into the current directory of the remote host.

* Each command consists of four uppercase ASCII characters, some with optional arguments.


TCP control connection on port 21
TCP data connection on port 20
FTP client
FTP server

Control and Data connections.

* Each command is followed by a reply, sent from server to client.

* The replies are 3 digit numbers, with an optional message.

* Some replies are:

331 Username ok, password Required

125 Data connection already open; transfer starting

425 Can't open data connection

452 Error writing file.

SMTP (Simple Mail Transfer protocol):

* SMTP defined in RFC 5321, is at the heart of Internet E-mail.

* SMTP transfers messages from sender's mail servers to the recipients mail servers.

* It restricts the body of all mail messages to simple 7-bit ASCII.

* For Example, suppose Alice wants to send Bob a simple ASCII message.

(1) Alice invokes her user agent for e-mail, provides Rob's e-mail address (bob@someschool.edu) composes a message, and instructs the user agent to send the message.

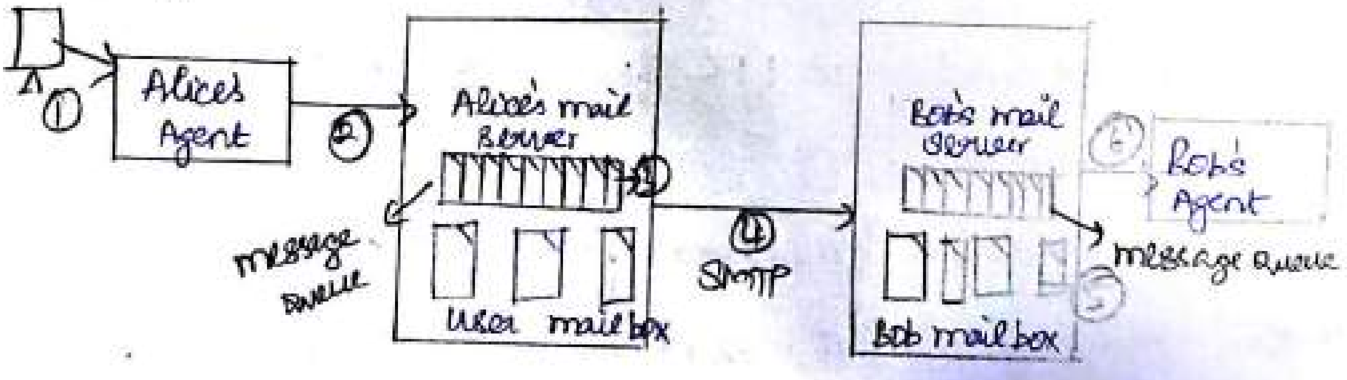
(2) Alice's user agent sends the message to her mail server, where it is placed in a message Queue.

(3) The client side of SMTP, sees the message in the message Queue. It opens a TCP connection to an SMTP server, running on Bob's mail server.

(4) After some initial SMTP handshaking, the SMTP client sends Alice's message into the TCP connection.

(5) At Bob's mail server, the server side of SMTP receives the message. Bob's mail server then places the message in Bob's mailbox.

(6) Bob invokes his user agent to read the message at his convenience.



Alice sends a message to Bob

* SMTP is a push protocol - the sending mail server pushes the file to the receiving mail server, the TCP connection is

Scanned by CamScanner

initiated by the machine that wants to send the file.

* TCP connection is a direct connection between sender and receiver and does not use intermediate mail servers for sending mail.

* The client SMTP has TCP establish a connection to port 25 at the server SMTP.

* If the server is down, the message remains in client's server and waits for new attempt to try later.

* SMTP client indicates the e-mail address of the sender and recipient, after introducing themselves, the client sends the message during the handshaking phase.

* The SMTP client uses the same TCP connection to send messages to the server, otherwise it instructs TCP to close the connection.

* SMTP relies on TCP for reliable transfer of messages to servers without errors.

* Telnet is used to directly communicate with a SMTP server. Eg telnet servername 25

* HTTP & SMTP transfers files from one host to another. HTTP → Web server to Web client. SMTP → one mail server to another mail server.

#POP3 (Post office protocol version 3):-

* It is a simple mail access protocol.

* POP3 begins when the user agent opens a TCP connection to the mail server on port 110.

* With the TCP connection established, POP3 progresses through 3 phases: (1) Authorization (2) Transaction (3) Update.

* During the I phase, authorization, the user agent sends a username and a password to authenticate the user.

* During the II phase, Transaction, the user agent retrieves messages, mark messages for deletion, remove deletion marks and obtain mail statistics.

* The III phase, update, occurs after the client has issued the Quit command, ending the POP3 session:

* In a POP3 transaction, the user agent issues commands, and the server responds to a each command with a reply.

* There are 2 possible responses: +OK and -ERR.

* The authorization phase has 2 principal commands: user< username> and pass < password>.

* In the transaction phase, there are two modes: (1) download and delete mode (2) download and keep mode.

#

- In the download - and - delete mode, the user agent will issue the list, retr and dele + quit commands.

- In the download and keep mode, the user agent leaves the message on the mail server after downloading them.

- After processing quit command, POP3 server enters the update phase.

- The POP3 server maintains some state information, during a session butnot across sessions of POP3.

IMAP (Internet Mail Access Protocol):

- It is an internet standard protocol used by e-mail clients to retrieve e-mail messages from a mail server over a TCP/IP connection.

- IMAP is defined by RFC 3501.

- It allows to access, organize, read and sore e-mail messages without having to download them first.

- IMAP is a remote file server, where all emails remain on the server until the client deletes them.

- In POP3 access, all the messages are downloaded to the local machine, so the remote access is not possible.

* An IMAP server will associate each message with a folder; when a message first arrives at the server, it is associated with the recipient's INBOX folder.

* The recipient can then move the message into a new, user created folder, read the message, delete the message, and so on.

* The IMAP provides commands to allow users to create folders and move messages from one folder to another.

* It also provides commands to search remote folders for messages matching specific criteria.

*IMAP server maintains user state information across IMAP sessions.

* Another feature of IMAP is that it has commands that permit a user agent to obtain components of messages due to low bandwidth connection.

For Eg audio + video clip are avoided.

* Web based E mail is provided by Google, Yahoo, etc where Web browser is an user agent and the user communicates with its remote mailbox via HTTP.

*SNMP (Simple Network Management Protocol):-

* The most widely used protocol that allows to read, write, various pieces of State information on different network nodes is SNMP.

* SNMP is a specialized request / reply protocol that supports two kinds of request messages : GET and SET.

* GET is used to retrieve a piece of state from some node, and SET is used to store a new piece of state in some node.

* A system administrator interacts with a client program that displays information about the network, using SNMP that runs top of UDP.

* A SNMP server running on that node receives the request, locates the appropriate piece of information and returns it to the client program, which then displays it to the user.

* SNMP depends on Management Information Base (MIB)

* The MIB defines the specific pieces of information - the MIB variables - that is retrieved from a network node.

* The current version of MIB, called MIB II, organizes variables into 10 different groups.

For Example,

System - General parameters of the system such as system's name, where the node is located, how long it has been up.

Interfaces - Information about all the network interfaces attached to the node such as physical address of each interface and how many packets have been sent + received on each interface.

Address Translation: Information about ARP and contents of address Translation table.

IP: variables related to IP, including routing table, how many datagrams it has successfully forwarded, and statistics about datagram reassembly.

TCP - Information about TCP connections such as number of passive & active opens, the no. of resets, the no. of timeouts, default timeout settings and so on.

UDP - Information about UDP traffic, including total NO. of UDP datagrams that have been sent + received.

SNMP Working :-

* The SNMP client puts the ASN.1 (Abstract Syntax Notation One) identifier for the MIB variable it wants to get into the request message, and sends this message to the server.
* The server then maps this identifier into local variable retrieves the current value held in this variable, and uses ASN.1/BER to encode the value it sends back to the client.

E-MAIL:-

* It is one of the oldest networking applications.

* It remains one of the Internet's most popular and important utilized applications.

* E-mail is an asynchronous communication medium - people send and read messages when it is convenient for them without having to co-ordinate with each other.

* Modern e-mail has features like mailing lists, spam, attachments, hyperlinks, HTML- formatted Text and photos.

* Internet E-mail system has 3 major components:
(1) User Agents (2) Mail servers (3) Simple Mail Transfer Protocol (SMTP)

* Eg: When Alice wants to send a message to Bob, User Agent allow users to read, reply to, forward, save and compose messages.

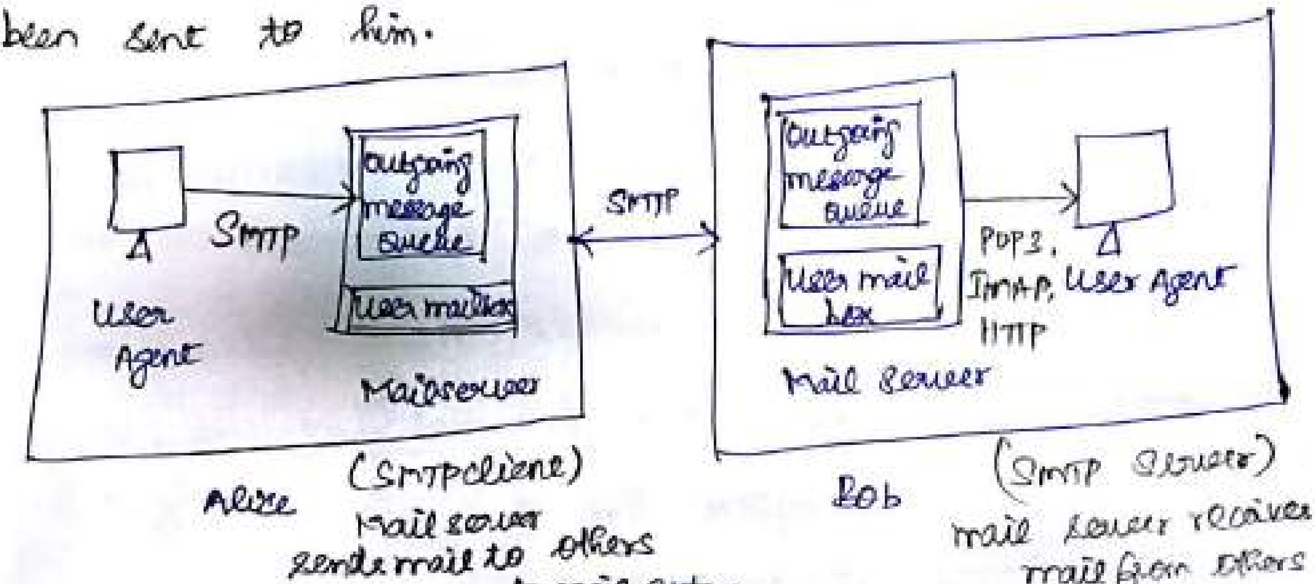
* When Alice is finished composing her message, her user agent sends the message to her mail server, where the message is placed in the mail server's outgoing message queue.

* When Bob wants to read a message, his user agent retrieves the message from his mailbox in his mail server.

* Mail servers form the core of the e-mail infrastructure.

* Each recipient such as Bob, has a mailbox located in one of the mail servers.

* Bob's mailbox messages & maintains the messages that have been sent to him.



E-mail system

* When Bob wants to access the messages in his mailbox, the mail server containing his mailbox authenticates Bob (with username & password).

* Alice's mail server deals with failures in Bob's mail server.

* If Alice's server cannot deliver mail to Bob's server, Alice's server holds the message in a message queue and attempts to transfer the message later after every 30 minutes, if there is no success after several days, the server removes the message and notifies the sender (Alice) with an e-mail message.

* SMTP is the application layer protocol for e-mail and it uses TCP to transfer mail from sender to receiver. It has two sides SMTP client and SMTP server.