Web Engineering

Unit-1

History of Internet.

→ 1968 - DARPA (Defense Advanced Research Projects Agency) during cold war contracts with BBN (Bolt Beranek & Neuman) to create ARPANet (Advanced Research Projects Agency Network)

→ 1970 - First five nodes - UCLA
- Stanford
- UC Santa Barbara
- U of Utah
- BBN..

→ 1974 - TCP specification by Vint Cerf..

→ 1983 → On Jan 1 (official birthday of Internet) New communication protocol was established called TCP/IP (Transmission Control Protocol/Internet -work protocol). It allowed different networks to "talk" to each other.

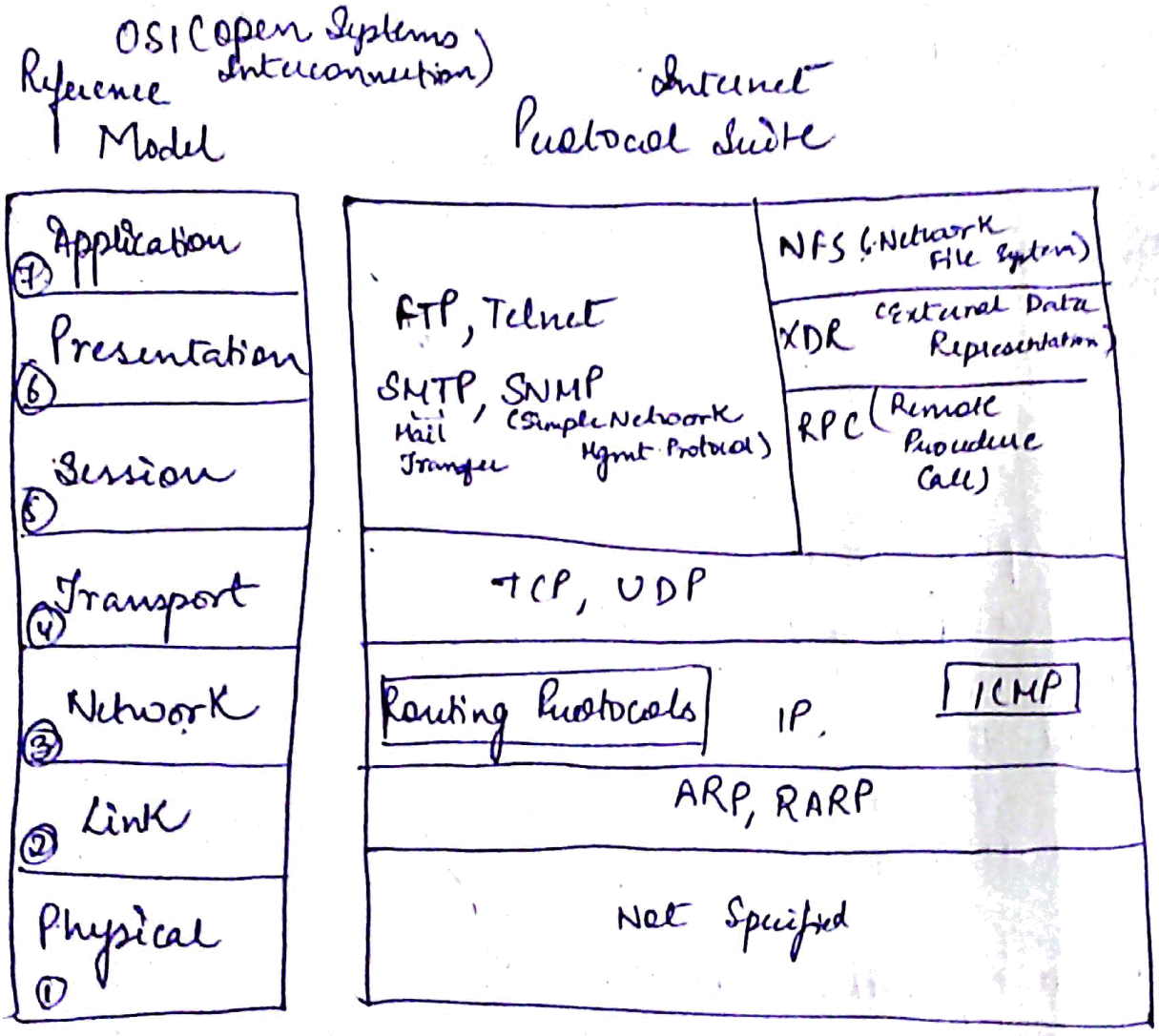
Packet Switching Invented 1964 → Department Invented 1965 → ARPANET 1969 → TCP/IP created 1972 → Internet Named & Accepted TCP/IP 1984 → WWW created 1989 → Mosaic Created 1993 → Age of eCommerce Begins 1995

BASIC INTERNET PROTOCOLS → Rules & conventions explaining how something is to be done
→ Tells how to communicate
→ Defines format of Data (being exchanged)

→ The Internet Protocols are the world's most popular open system protocol suite, because they can be used to communicate across any set of interconnected networks & are equally well suited for LAN and WAN comm.

Best Known Comm^{n} Protocol are - TCP
- IP

- The Internet Protocol suite not only includes lower layer (TCP/IP) but it also specifies common applications such as electronic mail, terminal emulation & file transf



⇒ IP (Internet Protocol)

- Specifies the format of packets, also called datagrams & the addressing scheme.
- Most n/w's combine IP with a higher level protocol called Transmission Control Protocol (TCP) which establishes a virtual connection b/w a destination & a source.
- It is a Network (layer 3) layer protocol.

IP allows you to address a package & drop it in a system, (like a postal system), but there's no direct link b/w you & the recipent.

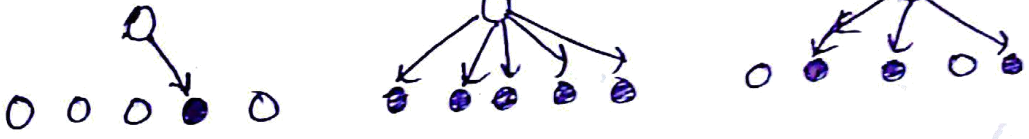
IP includes a set of rules that process the idea of unreliable packet delivery.
- how hosts & routers should process packets?
- how & when error messages should be generated?
- the conditions under which packets can be discarded?

IP Service
→ provides an unreliable
↓
IP doesn't make an attempt to recover lost packets

connectionless
↓
each packet is handled independently

best effort service.
↓
IP doesn't make guarantees on the service.

→ Supports: One to One (unicast), One to all (Broadcast), one to several (multicast)



⇒ OSI Model Layers.

) Physical Layer
- deals with Hardware of network
- This Layer's H/w: Cables, Connectors, Hubs, Repeaters etc.
- Function: Manages signaling to & from physical network connections
- Physical Layer protocols & Standards
  - Ethernet (802.3)
  - Token ring (802.5)
  - WiFi (802.11)

(2) Data Link Layer

- This layer deals with MAC addresses of devices
- Responsible for Physical Addressing, Error correction preparing the info. for the media frames
- Devices: Switches, Bridges, Wireless Access Points, NICs etc
- Data Link Layer Protocols & Standards
  - L2PP
  - PPP
  - SLIP etc

(3) Network Layer

- Layer deals with Packets (Data Bundles)
- Responsible for Logical addressing & routing
- Devices: Routers, Layer 3 Switches, Firewalls etc
- Network Layer Protocol
  - ARP
  - IP
  - RIP
  - IGRP

(4) Transport Layer.

- deals with segments
- Breaks info into segments & is responsible for connection & connectionless comm^{n}.
- Hardware: Proxy Server, Gateways, Firewall etc
- ~~TCP Lay~~ Transport layer Protocols - TCP
  - UDP

(5) Session Layer.

- Responsible for establishing, managing & terminating user connections
- Acknowledgements of data received during a session.
- Retransmission of data if it is not received by a device
- Protocols - STP, SIP, NetBIOS etc

(6) Presentation Layer.

- allows hosts & applications to use a common language
- performs
  - → Data formatting
  - → Encryption & Decryption for security
  - → Compression & Expansion
- Examples - eg MP3, JPEG, Mpeg etc

(7) Application Layer.

- This layer is what the user sees
- provides interface for users to communicate with applications
- ex: email, Instant messengers, http, smtp, Telnet, ping etc.

TCP/IP Model

It was developed earlier than the OSI model.

It has only 4 layers.

Application Layer
Transport Layer
Internet Layer
Network Access Layer

(4) Application Layer - protocols defined the rules when implementing specific n/w applications

eg: FTP (File Transfer Protocol)
Telnet (Remote Terminal Protocol)
SMTP (Simple Mail Transfer Protocol)
HTTP (Hyper Text Transfer Protocol)

(3) Transport Layer

- End to End Data Transfer.

Examples: TCP - Connection oriented. (Connection established before data exchanged)
- Reliable delivery of data.

- UDP (User Data Gram Protocol)
  - → Connectionless service
  - → Delivery is not Guaranteed (unreliable)

(3) Internet Layer
- defines the rules of how to find the routes for a packet to the destination.
- only gives best effort delivery (packets can be delayed corrupted, lost)

Examples: - IP
- ARP (Address Resolution Protocol) defines the procedure of network address / mac address translation.
- ICMP (Internet Control Message Protocol) defines the procedure of error message transfe

(4) Network Access Layer.
→ Also known as N/w Interface Layer
It is the layer in the TCP/IP model at which data is transmitted and received across the physical n/w
- mostly in H/w

Examples → ethernet
→ Token ring
→ ATM (Asyn. Transfer Mode)

★ IP Address
- a unique global address for a network interface.
- 32 bit long identifier.
- encodes a network number & a host number.

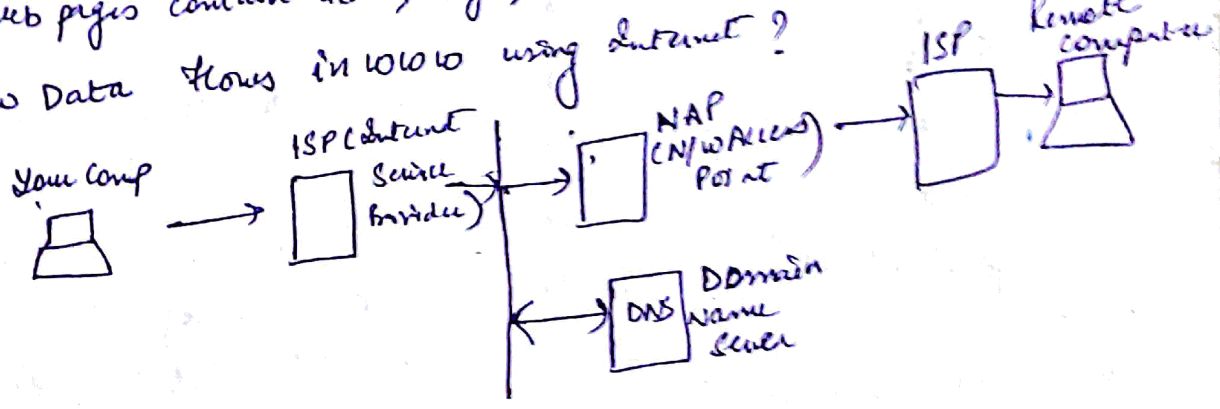
Class A	1.0.0.0	- 126.0.0.0
Class B	128.0.0.0	- 191.255.0.0
Class C	192.0.0.0	- 223.255.255.0
Class D	224.0.0.0	- 239.255.255.255 } Experimental address for future use
Class E	240.0.0.0	- 255.255.255.255 }

*. Subnetting - It enables the network administrator to further divide the host part of the address into 2 or more subnets.

World wide web

- invented by Tim Berners Lee in 1989
- It is an open source information space where documents and other web resources are identified by URLs
  - interlinked by hypertext links
  - & can be accessed via the Internet.
- web pages contain text, images, videos & other multimedia

How Data flows in WWW using Internet?



Internet

- it is a global system of interconnected computer networks.
- Its access is provided by ISPs
- It runs applications like www, ftp, html etc

World wide web -

- web is a collection of text documents & other resources, linked by hyperlinks & URLs
- Usually accessed by web browsers.
- Its an applications running on Internet.

★ Advantages

- Mainly free info
- Low cost of initial connection
- Accessible from anywhere
- Exchange of huge volumes of data

Disadvantages.

- Danger of overload & excess info
- Difficult to filter & prioritize info
- No Regulation
- No quality control over data.

W3C World Wide Web Consortium

- Created in October 1994
- "To lead www to its full potential by developing common protocols that promote its evolution & ensure its interoperability."

W3C Goals
- Universal Access
- Semantic Web
- Web of Trust

HTTP (HyperText Transfer Protocol)

- Application protocol for distributed, collaborative, hypermedia information systems.
- foundation of data comm^n for www.
- HyperText → Structured text that uses logical links (Hyperlinks) b/w nodes containing text.
- Comm^n protocol - to send & receive web pages & files on internet.
- HTTP version 1.1 most commonly used today.
- works by using a USER AGENT (Webbrowser) to connect to a server.
  server must be located using URL.
- https → secure version of http → encrypts all info
  → prevents from malicious codes / hackers.
  → usually seen in payment websites.

XHTML

eXtensible Hypertext Markup Language

- is a markup used to design websites & web pages
- much more strict than ordinary html & is based on both XML & HTML.

Why XHTML?

Due to the lax (non strict) nature of HTML, different web browsers can sometimes display websites incorrectly since there is more than one way to interpret poorly written html.

- Strict & formal nature of html will allow for web browsers to display the same page correctly.

Eg.

Incorrect html version

Correct; xhtml version

→ Elements must be properly nested

 Hello

→ Elements must always be closed.

 ═══

Empty elements too!
 , <hr />

→ Elements must be in lower case

<P> wrong , ✓.

→ XHTML attributes names must be in lower case & quoted.

<table width="100%">

→ In XHTML, <!DOCTYPE> line is mandatory.

HTML

- HyperText Markup Language → for describing web documents.
- It is described by HTML Tags.

HTML Tags

- They are Keywords surrounded by angle brackets.
- comes in pairs.
  - → <tagname> content </tagname>
    - start tag
    - end tag (always with '/')
- → They are of 2 types
  - Standalone -
 <hr>
  - Embedded <p> <body> ... etc

- All html documents begin with <html> tag & end with </html>.
- visible part of the webpage is written in <body> tag.

→ html documents are made up by html elements

- everything from the start tag to the end tag.

<h1> My first Head </h1>

- element content

→ Attributes provide additional information about html elements

<table width="100" height="50">

- tag
- attribute

• HTML Headings.

→ defined with <h1> to <h6> tags.

<h1> (most imp heading) ... <h6> (least imp heading)

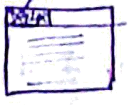
<h1> My Page </h1>

- <hr> → Horizontal Rule.
- Creates a horizontal line
- used to separate content

• HTML Head tag <head>

- placed b/w <html> tag & body tag.
- contains metadata (meta data → about data)
  ↳ (data not displayed on web page)

1) → Title element → defines the title of web page.
2) meta element <meta> → define the info about webpage
3) Style element <style> → used to define internal CSS style sheet
4) <link> → used to link external CSS style sheet.
5) <Script> <Base>



<Body> Tag

- <p> paragraph
-
 break line
- <pre> → preformatted text
  - text inside <pre> element is displayed in a fixed-width font & it preserves spaces & line breaks.

→ HTML Styles.

→ Use Style attribute.

Syntax: Style = "property: value;"

eg <body style = "background-color: lightgrey;" >
</body>

eg. This is WE

eg. <h1 style = "font-family: courier;" > Heading XYZ </h1>

eg. <h1 style = "text-align: center;" > Centre </h1>

→ — → bold text

→ — → italics

→ <u> — </u> → underlined.

→ <mark> — </mark> → highlighted text

→ <del> XYZ </del> → ~~XYZ~~ deleted / striked

→ → subscripted text

→ → superscripted text

- HTML Comments

<!-- comment -->

- Hyperlinks.

↓ It is a text or an image you can click on, jump to another document.

→ uses <a> anchor tag.

Syntax: link text

(attribute)

eg. Open Google

eg. Local Link

 Next Page

• HTML Images Syntax.

→ <img> tag, (src attribute specifies the URL)

Syntax: <img src = "url" alt = "some text">

(address of the image)

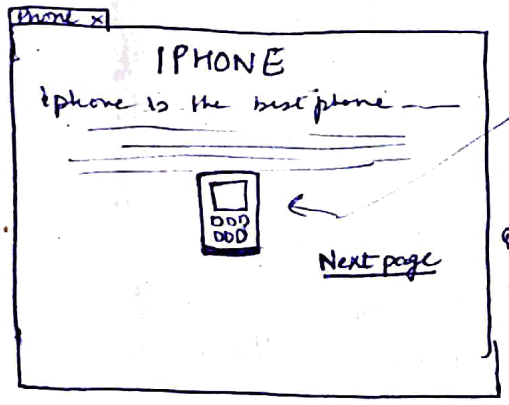
(alt attribute - specifies alternate text for image if image doesn't get displayed)

eg. <img src = "Sunset.jpg" alt = "sunset pic" style = "width: 128px; height: 128px;">

(Specify the extension)

we can specify the size of the image.

Q.



create two web page, background = blue on click of the image, a new page should get opened.

Ans <html>
<head> <title> Phone </title>
</head>
<body style = "background-color: blue;">
<h1 style = "text-align: centre;">
IPHONE </h1>
 Iphone is in ———
 <img src = "phone.jpg">
 Next Page
</body> </html>

HTML Lists

→ Unordered List
- Item1
- Item2
- Item3

→ tag.
Each item starts with <li> tag

eg.
```
<ul>
<li> Pagal </li>
<li> Idiot </li>
<li> Stupid </li>
</ul>
```

Output ←
- Pagal
- Idiot
- Stupid

• type attribute - defines the type of ul list
- disc (by default)
- circle
- square

eg.
```
<ul type="square">
<li> xyz </li>
<li> ypq </li>
</ul>
```
o/p →
- xyz
- ypq

→ Ordered List
1. —
2. —
3. —

```
<ol>
<li> — </li>
<li> — </li>
<li> — </li>
</ol>
```

• type attribute →
- "1" → numbers
- "a" → small letters
- "A" → capital "
- "i" → small roman letters
- "I" → big " "

eg.
```
<ol type="i">
<li> Hie </li>
<li> Hello </li>
</ol>
```
→
i. Hie
ii. Hello

• start

eg.

• Desc
- L

- start attribute : If we want to specify the position the list should start from.

eg. <ol type = "A" start = "C" >

 cat
 dog → C. cat D. dog E. elb
 elb

+ Description Lists.
- ↳ list of terms, with a description of each term

<dl> tag → defines the desc. list
<dt> tag → defines the term (name)
<dd> tag → def. describes each term

<dl>
<dt> coffee </dt>
<dd> - black hot drink </dd>
<dt> Milk </dt>
<dd> - white cold drink </dd>
</dl>

Nested HTML Lists.
List inside Lists

 coffee
 Tea

 Black tea
 Green tea

Milk

HTML Tables

- <table> tag,
- Tables are divided into → table rows <tr> tag
  → table data <td> tag
  → table headings <th> tag (optional)

```
<table border = 1 >
<caption> marks </caption>
<tr> <th> Name </th>
     <th> Roll no </th>
     <th> Marks </th> </tr>
<tr> <td> AB </td>
     <td> 123 </td>
     <td> 90 </td></tr>
<tr> <td> XY2 </td>
     <td> 231 </td>
     <td> 30 </td> </tr>
</table>
```

Attributes of Table: (Not supported in html5)

- align - left, center, right (alignment of table according to surrounding text)
- bgcolor - background color of table
- border - 0, 1 (0 - no border, 1 - border)
- cellpadding - specify the space b/w cell wall & cell conten (in pixels)
- frame - (specifies which part outside borders should be visib above, below, lhs, rhs, box, void.

Attributes with <tr> <td> - only works in one cell

align, bgcolor, border, color, height, width, valign

HTML forms → used to collect user input

<form> tag.

elements of form tag:

1. <input> element.

- name
- value
- size
- Type attribute
- maxlength

- time
- date
- checkbox
- password
- text (textbox)
- radio (Male, Female)
- submit (Submit Button)

<form>
input type = "text" >
input type = "radio" name = "gender" value = "male" > (used in groups)
put type = "submit" value = "Submit" >

attributes of form
- action attribute - defines the action to be performed after form is submitted
- method attribute
  - Specifies the HTTP method (GET or POST) to be used
  - form submission is passive
    - like a search engine query
    - avoid sensitive info as data will be visible in page address
  - form (updating data)
    - like password change
    - includes sensitive info
    - better security, data not visible on page address

<select> element
drop down list
<option> element defines options to select.
→ The list will normally show the 1st item as selected.
→ we can add a selected attribute to define a predefined option.

```
<select name = "Course" >
<option name value = "MBA" > MBA </option>
</select>
```

- <textarea> element - multiline input field
- <button> → defines a clickable button

Form example:

```
<html>
<head> <title> My form </title> </head>
<body>
<h1> Registration form </h1>
<form action = "Nextpage.jsp" method = "post" >
<b> Name: </b>
<input type = "text" name = "name" > </input>
<br>
<b> Password: </b>
<input type = "password" name = "pass" > </input>
<br> Select Your Gender:
<input type = "radio" name = "Gender" value = "male" >
<br>
<input type = "radio" name = "Gender" value = "female" > female
Select your Hobbies: <br>
<input type = "checkbox" name = "Reading" value = "yes" > Reading
<br> <input type = "checkbox" name = "Playing" value = "yes" > Playing
Select your City:
<select name = "City" >
<option value = "Delhi" > Delhi </option>
```

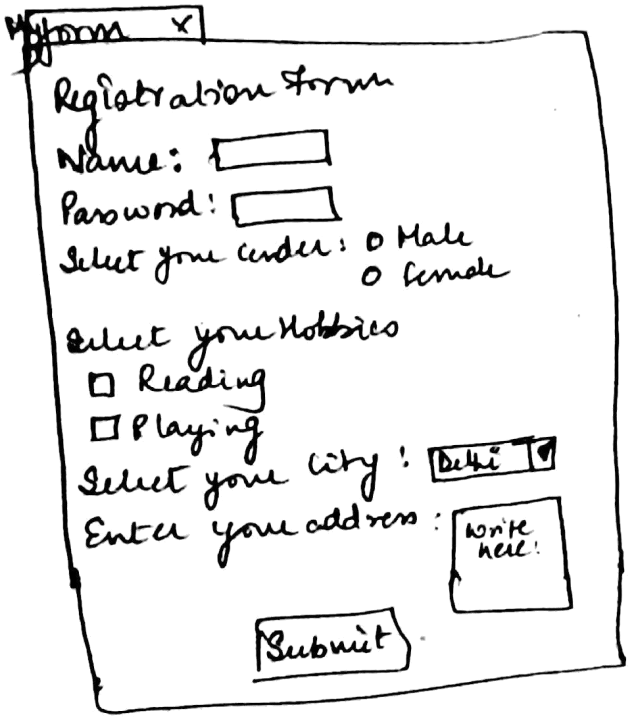
<option value="Mumbai">Mumbai</option>
<option value="Ghaziabad">Ghaziabad</option>
</select>

Enter your address:

<textarea name="address" rows="10" cols="5">Write here:
</textarea>

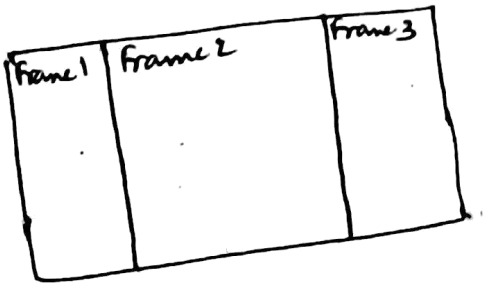
<input type="submit" value="Submit">

</form>
</body>
</html>.



HTML Frames → Web page gets divided into fr.

```
<frameset cols = "25%, *, 25%">
<frame src = "frame1a.html">
<frame src = "frame2.html">
<frame src = "frame3.html">
</frameset>
```



html Documents Definition

<!DOCTYPE> declaration before html tag.

↑
it is not a html tag,
it is an instruction to the web browser about what version of html the page is written in.

eg. <!DOCTYPE html>

CSS

Cascading Style Sheet

- describes how HTML elements are to be displayed on screen, paper or in other media.
- Saves a lot of work as it can control the layout of multiple web pages all at once.
- CSS removed the style formatting from the HTML page.

CSS Syntax

Selector { Property : value },

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

Selectors in CSS

1) ~~Strict~~ Simple html selector - all html tags <p>, <h1>, <h2>, <table> <td> etc.

2) Class → '.' represented as dot,

<body>
<p class = "A1" > —

CSS → .A1 { font : xyz
color : blue }

↑ works on all the tags under P tag

3) Id Selector → '#'.

<p id = "A1" > CSS → #A1 { font : —
color : — }

↑ works on only P tag

Types of CSS:

1. Internal
2. External
3. Inline
4. Imported.

→ Inline → (Highest Priority) ← (1).

<tag style = "property : value" >

eg: <h1 Style = "color : red" > .

→ Internal / Embedded (2).

<html > <head>
<style>
h1 {color : "red" }.
P { color : pink }.
</style>
</head>

→ External CSS (3).

Referred as: (stylesheet / alternate stylesheet)

Types : 1. Persistence - only one style sheet
2. Alternative - more than 1 cssheet
3. Preferred - out of 2, we prefer to use one style sh only

→ we have to create a text file with & save it with .css extension.

<html>
<head>
<style>
<link rel = "stylesheet" type = "text/css" href = "mystyle.css">

Imported → (4)

<html>
<head>
<style>
@importURL(" ")

→ imported from net.

→ Default Browser's Style sheet → (least Priority)

★ → CSS Background properties.

- background-color
- background-image
- background-repeat (:repeat-x (horizontally), :repeat-y (vertically), :repeat)
- background-attachment
- background-position (right top, bottom, etc)

fixed

★ Border styles: dotted, dashed, solid, double, none, hidden, groove, ridge

eg. p.dotted { border-style: (3d) dotted; }.

→ CSS Margin properties → used to ~~create~~ generate space around elements.

```
p {
    margin-top: 100px;
    margin-bottom: 100px;
    margin: right: 150px;
    margin: left: 150px;
}
```

Shorthand.

p { margin: 100px 150px 100px 80px; }
(top, right, bottom, left)

→ CSS Padding Properties.

↓ space around content
i.e space b/w the element content & the element border.

padding-top, right, bottom, left

p { padding: 80px 20px 30px 50px }.

→ Text Property

- color: blue;
- h1 { text-align: centre / left / right; }
- Text Decoration.

a { text-decoration: none; }
or h1 { text-decoration: overline / linethrough / underl

xt- Transformation.

eg:→ p.uppercase { text-transform: uppercase; XYZABC
}

p.lowercase { text-transform: lowercase; xyz abc
}

p.capitalize { text-transform: capitalize; Xyz Abc
}.

Letter Spacing → used to specify the space b/w characters in a text.

h1 { letter-spacing: 3px; H i e
}

h2 { letter-spacing: -3px; He
}

line-height - space b/w lines

word spacing - space b/w words

→ FONTS

- p {font-family: " Times New Roman ", Times, serif; }
  if these 2 are not found → this will run
- p { font-style: normal / italic / oblique; }
- font-size: 14px }.

CSS Links

eg. a {color: hotpink;
}

→ a:link → normal, unvisited link
→ a:visited → a link user has visited.
→ a:hover → a link when the user mouses over it
→ a:active → a link the moment it is clicked

eg. a:hover {
color: blue
background-color: cyan;
text-decoration: underline;
}

CSS Lists

• ul.
{
1) list-style-type: circle;
}
2) list-style-image: url('abc.gif');

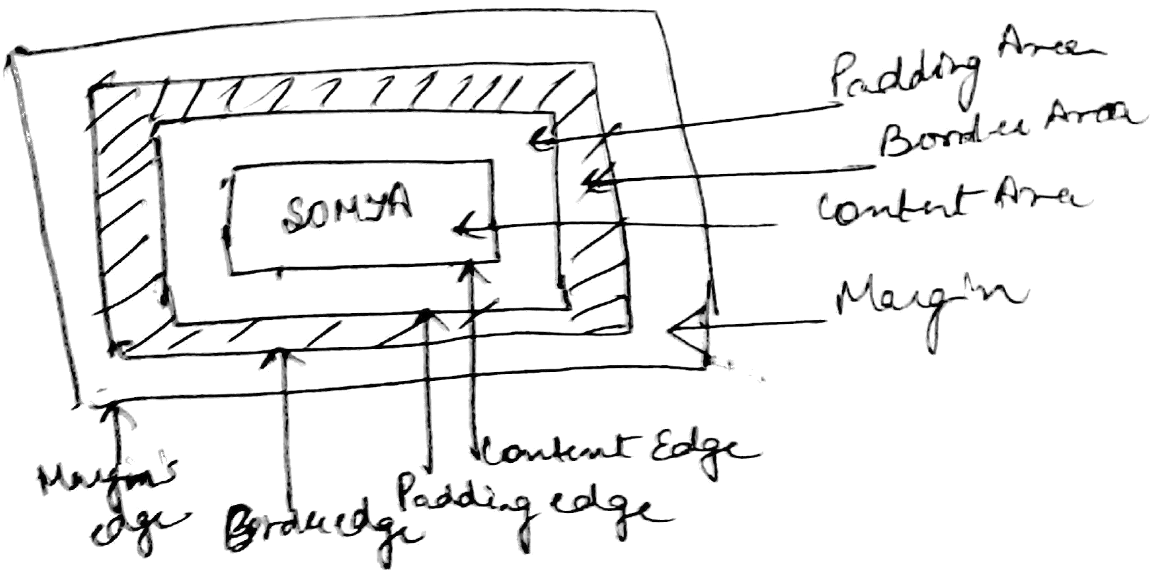
ol {
background: red;
padding '20px';
}

[illegible] padding
[illegible] { [illegible]
1) border: 1px solid black;
}
2) height: [illegible];
3) width: [illegible]px;
4) text-align: left;
5) vertical-align: bottom;
6) border-bottom: 1px solid blue;
}

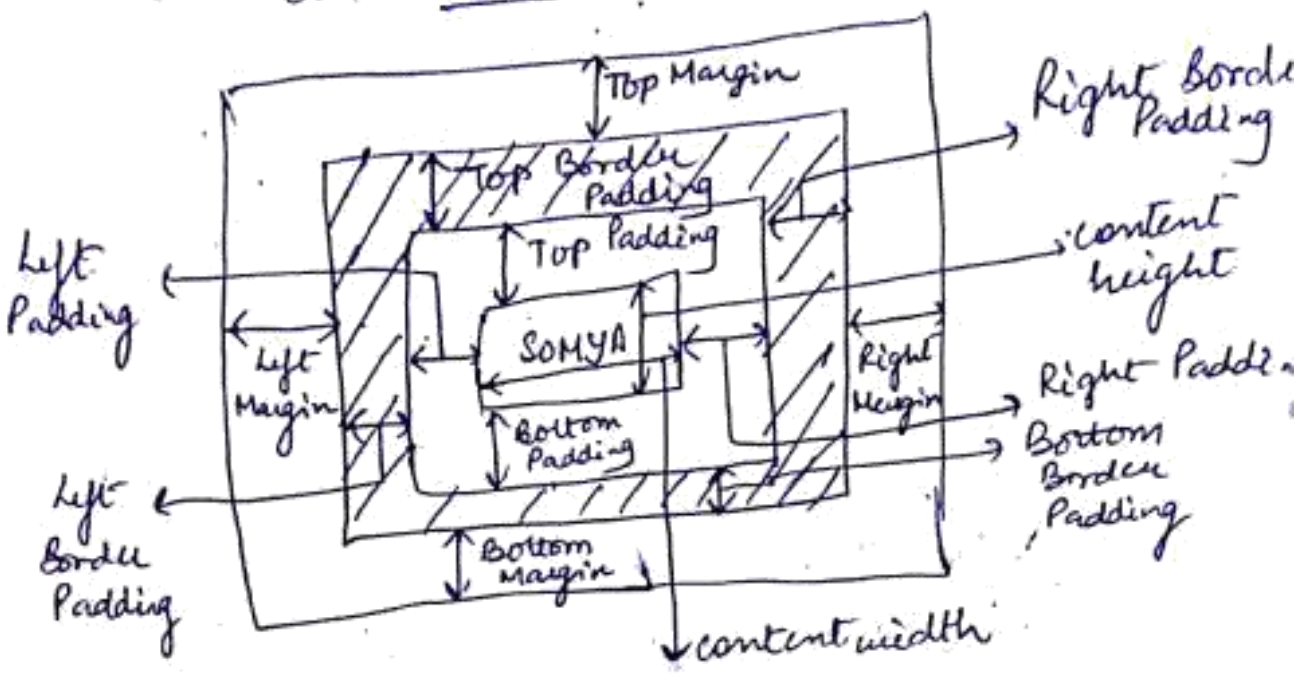
- tr:hover { background-color: pink }

CSS Box Model

It is a box that wraps around every [illegible]. Consists of margins, borders, padding & the actual content



CSS Box Flow Control



DHTML → Dynamic HTML

- It is the art of combining HTML, JavaScript, DOM and CSS to create dyn web pa
  (Document Object Model)
- DHTML is not a language or a web standard
- According to W3C: Dynamic HTML is a term used by some vendors to describe the combination of HTML, Stylesheets and Scripts that allows documents to be animated.

WML → Wireless Markup Language.

- Topmost layer in the WAP (Wireless Application Protocol) architecture is made up of WAE (Wireless Application Environment) which consists of WML & WML Scripting language
- It is an application of XML, which is defined in a document-type definition (dtd) for WAP devices eg Phone PDA

- It takes care of the small screen & low bandwidth of transmission.
- It is based on HDML & is modified so that it can be compared with HTML.
- Similar to html, uses tags & plain text.
- WML is the Markup language defined in the WAP specification. WAP files are written in WML.
- files are saved with ".wml" file extension
- WAP 2.0 (latest version)
  - uses XHTML mobile Profile (MP) (or WML (MP)) as markup language
  - uses CSS (WAP CSS) as style sheet.

XML → Extensible Markup language.

- It is a text based markup language derived from SGML (Standard Generalized Markup lang.)
- It is designed to transport & store data.
- or To represent structured data on webpage.
- XML was designed to carry data, not to display data.
- XML tags are not predefined, we have to define our own tags.
- It is designed to be self descriptive
- It is a W3C Recommendation.

Eg. <note>
<to> ABC </to>
<from> XYZ </from>
<heading> Reminder </heading>
<body> Don't forget my Birthday Gift! </body>
</note>

- It is just information wrapped in text.
- <to>, <from> → we invented / defined these tags on our own.

- XML is a software and hardware independent tool for carrying information.
- It simplifies data storage & sharing
- used for ~~exchange~~ offloading & reloading of databases.
- used to store & arrange the data

Syntax of XML DOCUMENT

<?xml version="1.0"?>] Document Prolog

<contact-info>
<name> ABC </name>
<company> TP cell </company>
<phone> 9050432178 </phone>
</contact-info>] Document Elements.

Syntax Rules

- XML Declaration
- Tags & Elements
- Attributes
- References - allows us to add additional text or markup using &amp
- Text

L Declaration → contains details that prepare an XML processor to parse the XML document.

<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

↑ Specifies the character encoding used in the document.

- It is case sensitive & must begin begin with <?xml>
- always the first statement of XML doc.
- standalone informs the parser whether the document relies on info from external source or not - by default → set to no
  - If yes → tells the processor that no external files.

Tags & Elements ↳ building blocks of an XML ↳ container to hold text

<element> . . . </element>.

Attributes.

Specify the property for the element, using a name value pair

- here attributes unlike HTML, are case sensitive.

eg. <a href="www.——.com">
(attribute)

How to add comment in XML:

<!--- write here --->