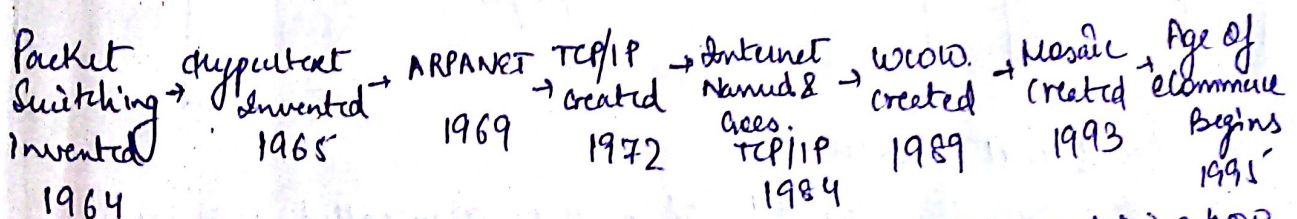


Web Engineering

Unit 1

History of Internet

- 1968 - DARPA (Defense Advanced Research Projects Agency) during cold war contracts with BBN (Bolt Beranek & Newman) 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.



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's.

'Best Known Commⁿ Protocol area - 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 trans.

OSI (Open Systems)
Reference Interconnection
Model

Internet
Protocol Suite

Application ⑦	FTP, Telnet		NFS (Network File System)
Presentation ⑥	SMTP, SNMP Mail (Simple Network Mgmt Protocol) Transfer		XDR (External Data Representation)
Session ⑤			RPC (Remote Procedure Call)
Transport ④	TCP, UDP		
Network ③	Routing Protocols	IP	ICMP
Link ②	ARP, RARP		
Physical ①	Not Specified		

⇒ 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 recipient.

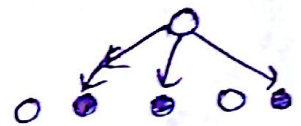
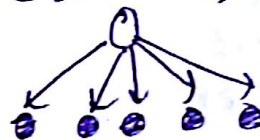
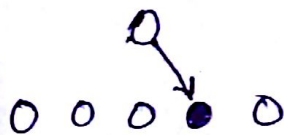
IP includes a set of rules that process the idea of unreliable packet delivery.

- how hosts & routers should process packets?
- how & when error messages should be generated?
- the conditions under which packets can be discarded?

P Service

→ provides an unreliable connectionless best effort service.
↓
IP doesn't make an attempt to recover lost packets
↓
each packet is handled independently
↓
IP doesn't make guarantees on the service.

→ Supports: One to One (Unicast), One to all (Broadcast), One to several (Multicast)



⇒ OSI Model Layers

1) 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
 - SDP 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.
- Hardware: Proxy Server, Gateways, Firewall etc
- ~~TCP~~ 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

TC
=
2
3

2
J

a

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

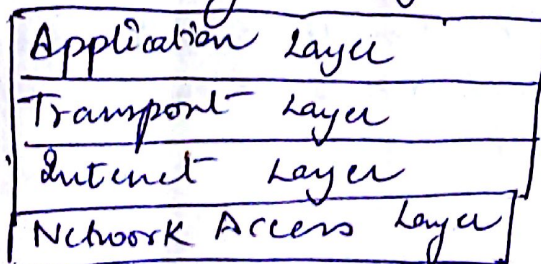
7) Application Layer.

- 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.



- 1) Application Layer - protocols defined the rules when implementing specific network applications
- eg: FTP (File Transfer Protocol)
 - Telnet (Remote Terminal Protocol)
 - SMTP (Simple Mail Transfer Protocol)
 - HTTP (Hyper Text Transfer Protocol)

2) Transport Layer

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

• UDP (User Datagram 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 transfer

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 n/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

Class E 240.0.0.0 - 255.0.0.0

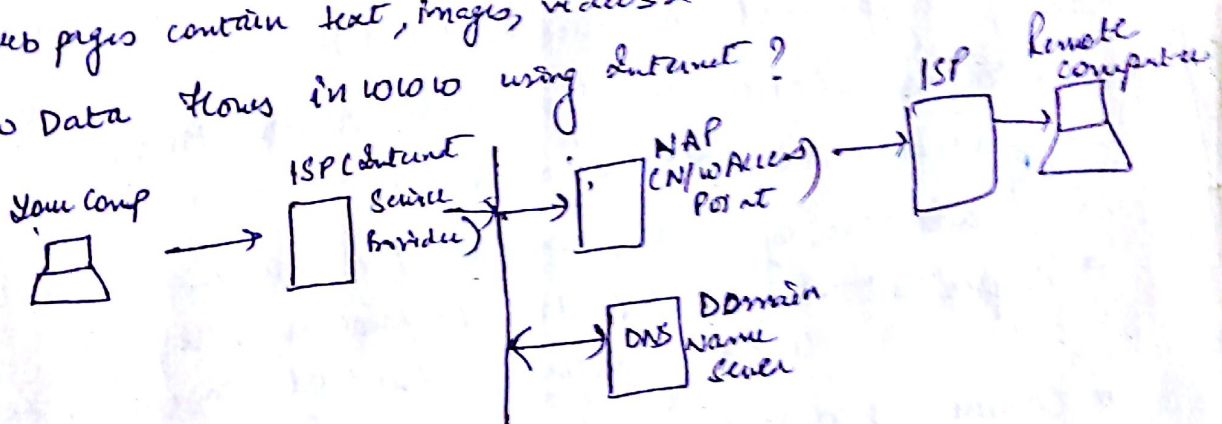
} Experimental addresses for future use

* 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 hypertexts & URLs
- Usidably 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, hypertext information systems.
- Foundation of data commⁿ for www.
- Hypertext → Structured text that uses logical links (hyperlinks) b/w nodes containing text.
- Commⁿ protocol - to send & receive web pages & files on internet.
- HTTP version 1.1 most commonly used today.
- Works by using a USER AGENT to connect to a server.
(web browser)
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 `<br>`

Correct, XHTML version `<br />`

→ Elements must be properly nested

`<div><b> Hello </b> </div>`

→ Elements must always be closed.

`<p> == </p>`

Empty elements too! `<br />`, `<hr />`

→ Elements must be in lower case

`<P>` wrong, `<p>` ✓.

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

`<table width = "100%">`

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

HTML

- 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 pair.

$\rightarrow \langle \text{tagname} \rangle$ content $\langle / \text{tagname} \rangle$

↑ ← end tag

start tag

← end tag (always with ').

→ They are of 2 types

- Stand alone - `<br>` `<br />`
- 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.

$\chi h \propto$ my first head $\langle 1/h \rangle$
 $\uparrow$
 element content

→ attributes provide additional information about html elements

tag $\rightarrow$ table width = "100" height = "50" $\rightarrow$
 attribute

members. • HTML Headings.

→ defined with $\langle h1 \rangle$ to $\langle h6 \rangle$ tags.

↑
most imp
heading

↑
least imp
heading

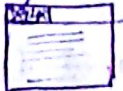
$\langle h1 \rangle$ My Page $\langle /h1 \rangle$

• $\langle hr \rangle$ → Horizontal Rule.

- Creates a horizontal line.
- used to separate content.

h'/' • HTML Head tag $\langle head \rangle$

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

- 1) $\langle title \rangle$ element → defines the title of web page.
- 2) meta element $\langle meta \rangle$ → define the info about webpage 
- 3) style element $\langle style \rangle$ → used to define internal CSS style sheet.
- 4) $\langle link \rangle$ → used to link external CSS style sheet.
- 5) $\langle script \rangle$ & $\langle base \rangle$

Body Tag

- $\langle p \rangle$ paragraph
- $\langle br \rangle$ break line
- $\langle pre \rangle$ → preformatted text
 - text inside $\langle pre \rangle$ 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. `<p style = "color: red;"> This is WE </p>`

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

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

→ `<b>` — `</b>` → bold text

→ `<i>` — `</i>` → italics

→ `<u>` — `</u>` → underlined.

→ `<mark>` — `</mark>` → highlighted text

→ `<del>` XYZ `</del>` → XYZ deleted / Striked

→ `<sub>` — `</sub>` → subscripted text

→ `<sup>` — `</sup>` → 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: `<a href = "url"> link text </a>`
(attribute)

eg. `<a href = "http://www.google.com"> Open Google </a>`

eg. Local link

`<a href = "Secondpage.html"> Next Page </a>`

• HTML Images Syntax.

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

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

(address of the image)

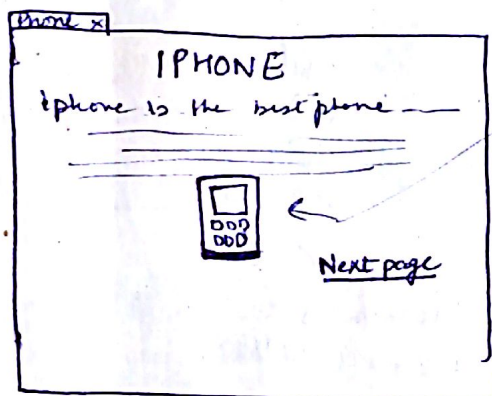
(alt attribute - specifies alternate text for image if image doesn't get displayed)

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 this 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: center;">
    IPHONE </h1>
```

`<p> iPhone is m ——— </p>`

`<a href = "newpage.html"> <img src = "phone.jpg"> </a>`

`<a href = "Nextpage.html"> Next Page </a>`

`</body> </html>`

HTML Lists

• start

- Unordered List • Item1
• Item2
• Item3

eg.

→ tag.

Each item starts with tag

- Pagal
- Idirot
- Shyid

← Output

```
<ul>
  <li> Pagal </li>
  <li> Idirot </li>
  <li> Shyid </li>
</ul>
```

- type attribute - defines the type of ul list
- ~~disc~~ disc (by default)

eg. <ul type="square">
 xyz
 yps

o/p → □ xyz
□ yps

→ 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">
 Hie
 Hello

→ i. Hie
ii. Hello

Wk

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

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

<code></code>	<code> cat </code>	<code>C. cat</code>
<code></code>	<code> dog </code>	<code>D. dog</code>
<code></code>	<code> eli </code>	<code>E. eli</code>
	<code></code>	

→ Description Lists.

default → 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

eg. `<dl>`

`<dt> coffee </dt>`

`<dd> - black hot drink </dd>`

`<dt> milk </dt>`

`<dd> - white cold drink </dd>`

`</dl>`

→ Nested HTML Lists.

eg List inside lists

```

<ul>
  <li> coffee </li>
  <li> Tea
    <ul>
      <li> Black tea </li>
      <li> Green tea </li>
    </ul>
  </li>
</ul>

```

1000
2

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 (no border), 1 (border)
- `cellpadding` - specify the space b/w cell wall & cell content (In Pixels)
- `frame` - (specifies which part outside borders should be visible above, below, lhs, rhs, box, void)

Attributes with `<tr>` `<td>` - only works in one cell
`align`, `bgcolor`, `border`, `color`, `height`, `width`, `valign`

HTML Forming → used to collect user input

<form> tag

elements of form tag:

• <input> element

checkboxes

attributes:
- name
- value
- size
- type attribute
- maxlength

time
date

checkbox

password

text

textbox

radio

Male Female

submit

Submit

Submit

Button

</form>

input type = "text"

(used in group)

input type = "radio"

name = "gender" value = "male"

input 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 seek

→ The list will normally show the 1st item as selected.

→ we can add a selected attribute to define a predefined option.

```

<select name = "Country" >
  <option name value = "MBA" > MBA </option>
</select>

```

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

Form example:

```

<html>
<head> <title> Myform </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: <br>
    <select name = "City">
      <option value = "Delhi" > Delhi </option>

```



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

Enter your address:

```
<textarea name="address" rows="4" cols="5">Write here:
</textarea>
```

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

```
</form>
```

```
</body>
```

```
</html>
```

Registration Form

Name:

Password:

Select your gender: ☐ Male
☐ Female

Select your hobbies

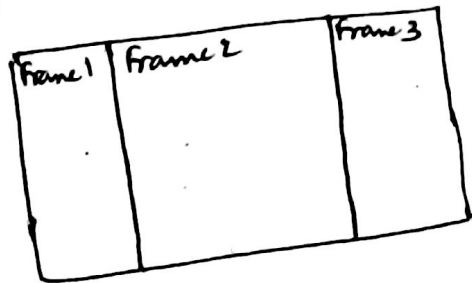
☐ Reading
☐ Playing

Select your city:

Enter your address:

HTML Frames → Web page gets divided into frames

```
<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) ~~Select~~ Simple html selector - all html tags `<p>`, `<h1>`, `<h2>`, `<table>`, `<td>` etc.
- 2) Class → '.' represented as dot.

`<body>`
`<p class = "A1" >` — CSS → `A1 { font : xss ; 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:

- Types:
1. Persistence - only one style sheet; <style>
 2. Alternative - more than 1 CSS sheet; <link>
 3. Preferred - out of 2, we prefer to use one style sheet 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 a (4)

<html>

<head>

<style>

→ Imported from net.

@import URL(4_4)

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

provided
as:

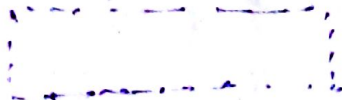
list

styles

is she
very
h

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

eg. p.dotted {border-style: 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 Properties

- color: blue;

- h1 {

text-align: centre / left / right; }

- Text Decoration.

a { text-decoration: none; }

or h1 { text-decoration: overline / line-through / underline }

text 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; H e }

line-height - space b/w lines

word spacing - space b/w words

→ FONTS

• p { font-family: "Times New Roman", ^{Times, serif} ~~serif, Times~~; }

if these 2 are not found → this will run

• p { font-style: normal/italic/oblique; }

• font-size: 14px }

class

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;

}

border. 1/2 in. solid black;

1) height 1000,

9) admissible copy;

9) text-align: left;

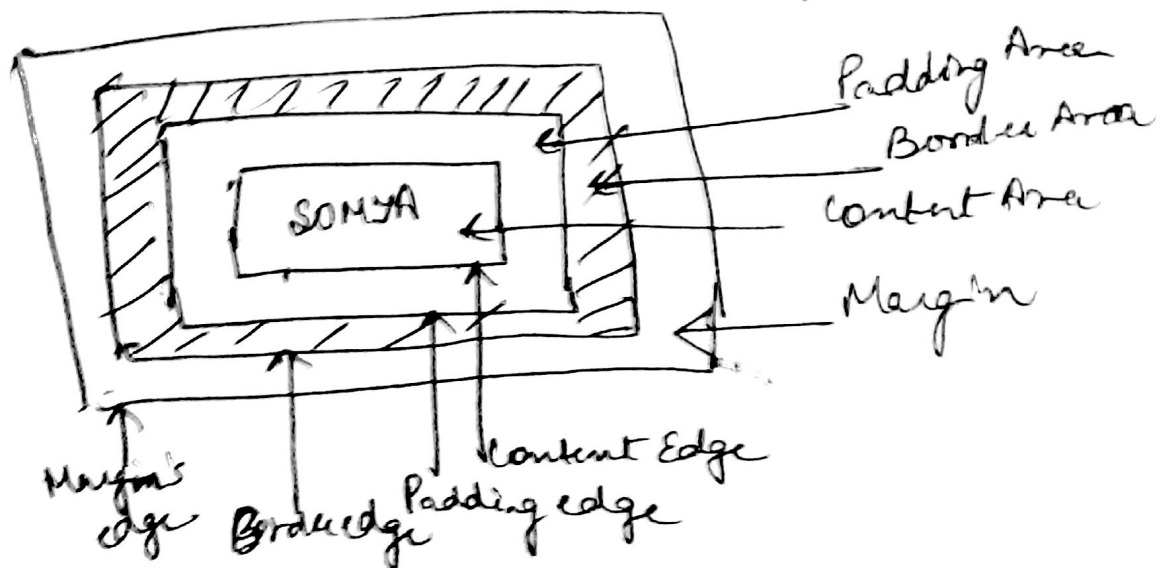
s) vertical-align: bottom;

6) border-bottom: 1px solid blue;

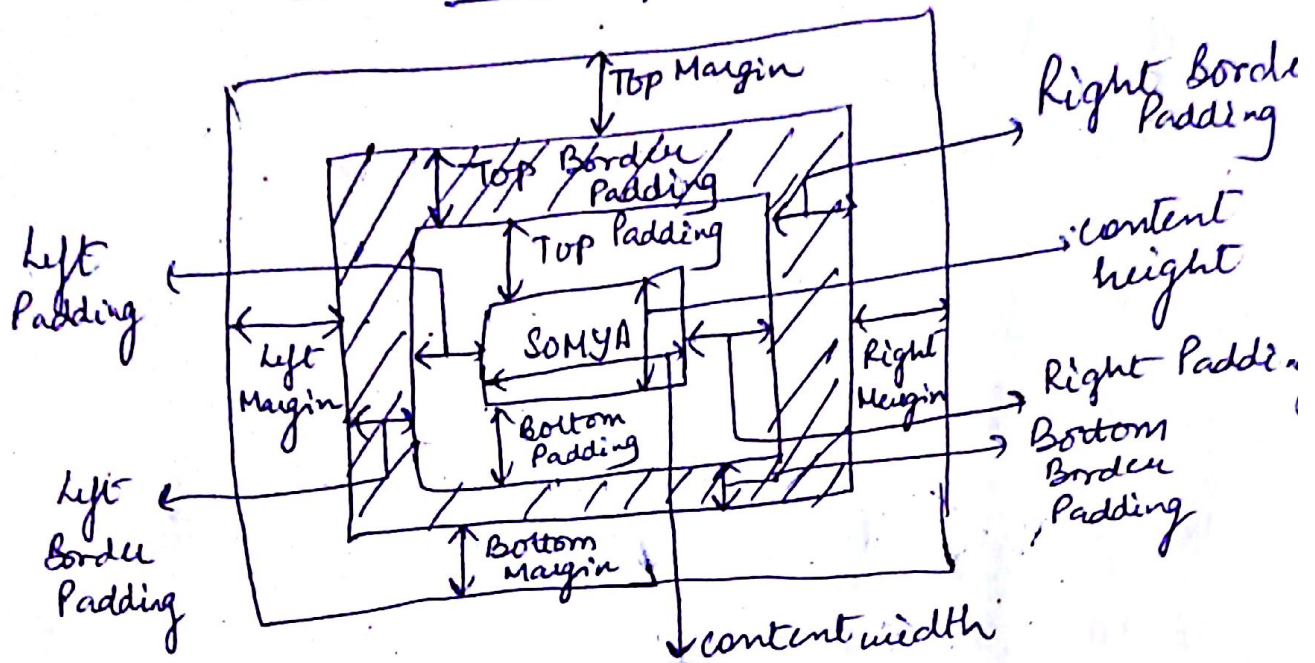
- ²¹br: hover & background-color: pink;

✓ CSS Box Model

21- Is a box that crops around everything.
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 dynamic web pages (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 HTML & is modified so that it can be compared to with HTML.
- Similar to HTML, uses tags & plain text.
- WML is the Markup Language defined in the WAP specification. WAP sites are written in WML.
- files are saved with ".wml" file extension
- WAP 2.0 (latest version)
 - └ uses XHTML mobile phone (MP) ^{or WML (MP)} as markup language
 - └ WCSS (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> TPCell </company>  
  <phone> 9650 432178 </phone>  
</contact-info> ] Document Elements.
```


Syntax Rules

- XML Declaration

- Tags & Elements

- Attributes

- References - allows us to add additional text or markup using &

- Text

XML 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.

or

It is case sensitive & must 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 -

Tags & Elements
building blocks of an XML
containers to hold text
element > ... </element>

by default → set to no
if yes → tells the processor that no external files.

Attributes

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

• these attributes unlike HTML, are case sensitive.

• eg. `<a href="www. --- .com">`
↑
(attribute)

How to add comment in XML?
`<!-- write here -->`