

HTML - JAVASCRIPT

3. Internet

1. Network

2. type of Networks.

4. Web

- Tim Berners Lee

- IETF

- W3C Group | WHATWG Group.

By this Group

Some languages

like Java, PHP, .NET

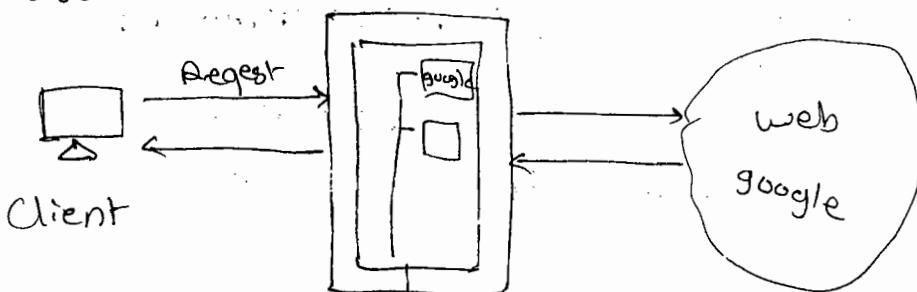
are following HTML 5

designed by W3C Group

By this Group Some languages
are like Python, Perl etc
are following HTML 5
designed by WHATWG
group.

www.w3.org

5. Webserver



Server (Combination of S/w & H/w)

Web Servers →

- ① Microsoft IIS
- ② Apache Tomcat
- ③ JBOSS
- ④ Light PGI
- ⑤ IBM Lotus

MNC's use ~~Ticketing~~ Tools → Lotus Notes, Clarify, Remedy
It is used to communicate b/w clients & Developers,
Developers & etc.

6. Website → It is a virtual directory on Web Server.

Open IIS → right click expand Servername →

Expand sites → rightclick on Default website

→ Add virtual directory → Give alias name as

BigCinemas → Give physical path → D:\BigCinemas\website

website (Just an folder → virtual directory)

8.

9.

10.

11.

12.

13.

→

eg.

Virtual Path (Web)

→ http://localhost/Bigcinemas

Physical path.

D:\BigCinemas\website (Folder)

→ This is for clients to access

This is for developer to make changes.

Use → uniform resource locators

↳ It is a virtual path

→ User should not have access to physical path. They should not disturb the contents.

→ We can try this in network (2 PC's etc)

In command prompt "ipconfig" → will get the IP address of m/c.

→ Uniform resource locator is nothing but virtual path.

Qn Ans: Difference bet web application & website

- web app' is individual app'.

- web app' will allow interaction with user.

→ eg →

10.

11.

12.

13.

① website is a collection of several websites & web applications.

② website can interact with website.

e.g. → yahoo is website & 'chat' is an application in yahoo website

IIS
↳ Default site

└ Add Application
└ Add Virtual directory

14.

15.

8. Blog (Web-Log) → keep info about History (ie Journals)
↳ Blogs are Journal on websites. Internet.
9. microblog → many pages people will post their personal info on single page

10. Wiki

11. Podcasting

12. Widget

13. RSS

→ Microblog →

Eg. Twitter is a microblog.

MVP, MVC, (MVVM) ← architecture.

→ MVVM is used for Twitter - ie everything is opened in single page. ie SPA.
(ie Single Page Application)

→ It is developed using knockout js.

⑩ Wiki → (ie Quick) Eg. Wikipedia

It allows anonymous users to edit its content.

Eg. Type of Wiki is Wikipedia. we can edit the pages in wikipedia having "Edit Page".

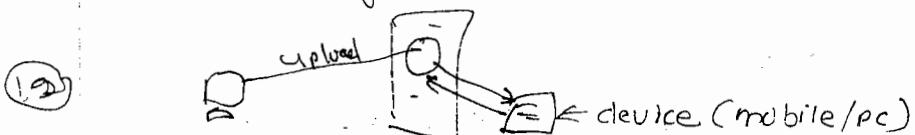
→ for developing Web parts which is part of SharePoint which is used in Wiki,

Eg. www.imdb.com (movie related DB)

In this we are able to give rating, certificates etc.

⑪ Podcasting → iPod is Pod catcher. & Pod means media.
Eg. iTunes (apple), YouTube (Google), Windows Media Player (MS)

This is toughest work in web development.



We are connecting to some Podcasting

⑫ Widget →

Eg. BMI calculator of hospital, Gpa calculator in Bank.

(13) RSS - Real Simple Syndicate.

→ Synchronizing all

RSS small set of code which is updated on browser and we will get notification about any new updates related to website to which we are subscribed. In mobile also we have RSS.

If we delete fb RSS in mobile then we will not get notifications automatically, if we want to get notifications we have to refresh. PQ.

(14) Web Pages →

→ Static (htm, html)

→ Dynamic (asp, jsp, php)

① Static Pages → one predefined pages which need not to compile & process they just need to parse (ie. converting) in understandable language.

② Dynamic Pages → need to be compiled, processed & rendered the dynamic page.

Rendering is done by View Engines.

Rendering server side code into browser understandable code.
Eg -> 1) Spock

2) Nhamel

3) ASP.NET

4) Riangles

5) Razor

Compile → understandable code

process → calculating result / generated o/p.

rendered → sending the generated o/p.

Q? Difference b/w htm & html?

→ If we are exporting from 3rd party then it extension will be htm, if we are giving extension

manually then if given as "htme" then the extension will be htm.

e.g.) If we save our word document & 'Save type' selected as "web page" at bottom.

6-1-16

Web Terminology

- ① Network → A computer Network Comprises of group of computers connecting with each other for sharing of information & Resources.
- ② Types of Networks → Computer Networks are classified into 3 major Types based on their range & capacity. They are.
<1> LAN <2> MANI <3> WAN.
Local Area ↓ metropolitan Area → wide Area
- ③ Internet → It resembles a wide area network that connects computers all over the world.
- ④ Web → ① web is a portion of internet with restricted access ability. ② It works with the mechanism 'request & response'.
③ Tim Berners Lee introduced the concept of web.
④ W3C (World Wide Web Consortium) maintains the Standards of web..
- ⑤ Web Server → A web server resembles both hardware & software, it satisfies the request of clients by sending & receiving the data. The popular webserver Softwares are →
① Microsoft IIS ② Apache Tomcat ③ JBoss ④ IBM Lotus
- ⑥ Website → A website is a virtual directory in the web server. It contains a collection of applications & websites. A typical website will not allow interaction with user.
- ⑦ Web Application → A web application is configured in a website, which allows interactions with user.
- ⑧ Web Page → A website or application Comprises of information stored in the form of hypertext documents known as "WebPages". They are classified into 2 types.
① Static Pages ② Dynamic Pages.
① Static Pages → static Pages are predefined in the server they are just parsed into understandable format of browser

& rendered as output at step 4 i.e. ~~at~~

eg:- 1) Home.html 2) About.htm

(2) Dynamic Pages → The pages that are generated as a response to the client request are known as dynamic pages. They are created, compiled & processed on server. However browser can't understand server-side technologies. These dynamic pages are converted into "html response".

Note → View Engines are responsible for converting the server side output into html response.

eg:- asp.net Engine, node.js E.g. SparkEngine, django E.g. Tornado

eg:- for Dynamic Pages → ① results.asp

② mail.php
③ shopping.jsp.

(3) Blog (Web-Log) → blogs are journals on internet, usually published by individual users and updated periodically.

(4) Microblog → A microblog allows multiple users to post their personal information on single page. It is also known as "SPA (Single Page Applications)".

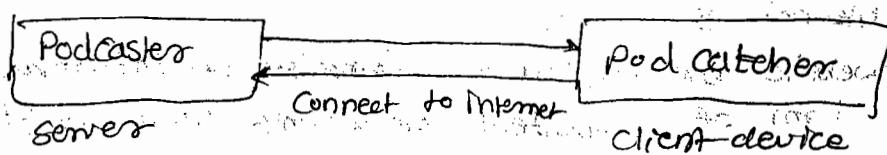
It uses a framework MVVM (Model-View-View-Model).

eg → Twitter (It's a microblog not a website/webapp)

(5) Wiki (Quick) → A wiki allows anonymous users to edit its content.

eg:- wikipedia, www.imdb.com, wiki maps,

(6) Podcasting → A podcasting allows users to upload media files so that they can synchronise their devices with the Podcasting & get all updates.



egs 1) iPod is Pod catcher, & Itune is Podcasting

2) youtube

3) windows media player

- (13) Widget → It is a small set of code embedded into any appl' to perform any specific task.
eg → Loan Calculator, BMI Calculator, World Clock.
- (14) RSS (Real Simple Syndicate) → It is a feed installed on your device or appl' that gets updates from the synchronised services

Locating Web Server On your machine

- 1) open Windows Control Panel (Run → Control)
- 2) switch to large icon view.
- 3) Go to Administrative Tools
- 4) Look for Internet Information Services Manager (IIS)
→ Adding IIS from Programs →
 - ① Open Control Panel.
 - ② Go to Programs & Features.
 - ③ click on the link "Turn windows features on or off".
 - ④ select "Internet Information Services".
 - ⑤ Click OK.

- * Fiddler is world popular web debugger.
- * F12 key to see debugger of browser.
- * In firefox we have firebug web debugger we have to download.

Creating A new Website On IIS.

- ① Open IIS (Run → inetmgr)
- ② Expand Local Computer (<http://localhost>)
- ③ Expand the folder "Sites"
- ④ ↳ right click on Default website.
- ⑤ ↳ select "Add Virtual Directory"
- ⑥ Specify the following options.
 - Alias Name : Amazon
(website name)
 - Physical path : D:\Amazon\Website
- ⑦ Click OK

Use \rightarrow `http://localhost/Amazon`
[use case not case sensitive]

Markup Language

① First browser is "Mosaic" designed by "Brendan Eich"

He developed the "Java Script". The earlier name was ~~mosaic~~ → "live script" → Sun Microsystems gave name as "JavaScript".

→ Netscape was designed by Brendan Eich & all our browsers belongs to Netscape family.

→ First Markup Language is \rightarrow SGML \rightarrow HTML \rightarrow Tim Berners Lee

IETF Group start developing HTML \rightarrow

W3C Group again started developing it \rightarrow HTML5

① W3C - HTML5

www.w3.org

markup word is derived from marking up

HTML Element & Tag \rightarrow

→ Tags are used to build Element.

Classification of Elements in HTML \rightarrow

① Normal Element

② Void Element

③ RC Data Element

④ Rich Text Element

⑤ Foreign Elements

① All normal elements which determines the value

then they are these elements will have starting tag & closing tag

Eg:- `<h1> </h1>`

- ② Void elements will not return value & they are self ending tag which will not have ending tag.
 eg -
 (or)

- ③ <hr>
- ④ RC Data Elements will not allow any elements inside it.
 eg <textarea> </textarea>
- ⑤ Rich Text elements → <html> will not print display on browser
 ∵ <html> → will print display
 eg © It can be used as individual or as it can
- ⑥ Foreign Elements → Not native to html but we can use it in html by importing library
 eg MathML

Web debugger → It is the SW tool used to [7/11/16] track the performance of the page. It also provides the options to debug the application.

The popular web debuggers are "Telerik Fiddler", "firebug" (belongs to firefox), "Internet Explorer Debugger" (By pressing F12)

HTML

- ① A markup language is used to present the data on web. The 1st markup language is GML (Generic Markup Language).
- ② The word markup is derived from computer terminology where marking up is done in any software in order to present the data.
- ③ Tim Berners Lee introduced HTML in 1990.
- ④ IETF (Internet Engineering Task Force) developed several versions of HTML.
- ⑤ W3C & WHATWG (WHAT work group) are developing HTML5 individually (web hypertext App) Technology work group.

Elements and Tags →

① HTML Comprises of elements that present the data & the elements are marked by using tags.

② In HTML elements are classified into 5 types.

④ Types of HTML Elements →

① Normal Elements.

② Void

③ Row Text

④ RC Data

⑤ Foreign elements.

① Normal Elements → The normal elements always returns a value, & they comprises of start-end tags. Separated with the attributes & Delimited with angle braces. "<>"

eg → <h1> Welcome </h1>
 ↑ ↑
 Start Tag end tag.

② Void Elements → The term void element is used to designate elements that must be empty. usually they are "self ending elements". i.e. they don't have an end tag.

eg →
, &

③ RC Data Elements → The term RC Data elements refers to the elements within which character references are supported but all their content is treated as row text & not elements.

eg → <textarea>

 Bold

</textarea>

or

 Bold

④ Raw Text Elements → The term raw Text elements refers to elements that treat their text as raw text & they also append the functionality when used with other elements.

e.g. < > &

⑤ Foreign elements → The term foreign element refers to the elements that are not native to html. They need a library to be imported in order to use with html.

e.g. SVG, MathML.

<!DOCTYPE < !^{comment to tell parser to display using html5} DOCTYPE html>

<html> (or) <xaml> (or) <xme> (or) <xhtml> (or) <wsdl>

These are also used to different elements so we have to tell which should be used.

<head>
<link>
* <meta>
<title>
<script>
<style>
</head>

<title> → is used to give title will display in tab of browser.

<links> → favicon (16x16) i.e favorite icon & extension .ico.

<meta name="keywords" content=" " />

<meta name="Description" content=" " for SearchEngineOptimization

<meta http-equiv="refresh" content="4" → Refreshes the page in every 4 sec.

<meta name="author" description=" " To

<meta charset="utf"

What is HTML encoding? → utf-8

↳ unicod information format.

Basic HTML Element

8-2-18

| <u>Element</u> | <u>Description</u> |
|-------------------|---|
| ① <!DOCTYPE html> | It specifies HTML Version 5 It tells the parser to use HTML 5 Encoding in order to render the page. |
| ② <html> | It specifies the language used is html. |
| ③ <head> | It describes the head section of page which includes <title>, <link>, <meta>, <script> & <style> |
| ④ <link> | links the external files to a web page which includes: favorite icons, style sheets etc. |
| ⑤ <title> | It specifies the title to be displayed in the browser title bar or tab. |
| ⑥ <meta> | It describes metadata, which contains information about your application like → keywords, description, author name etc. [meta is used by Search Engine Optimizer] |
| ⑦ <script> | It is used to import scripts into a web page like Javascript, Jquery etc. |
| ⑧ <style> | It is used to import and configure styles for a web page. |

Styles can be

- ① Embedded
- ② Internal
- ③ CSS (i.e External)

① Ex → index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title> Amazon | India </title>
```

```

ctrl - > zoom in   ctrl - > zoom out
<link rel="Shortcut Icon" href="icons/favicon.ico">
<meta charset="utf-8">
<meta name="keywords"
      Content="Shopping Online, Shopping in India">
<meta name="description"
      Content="India's most used online shopping site">
<meta name="author" Content="Amazon Admin">
<meta http-equiv="refresh" Content="4">
</head>
</html>

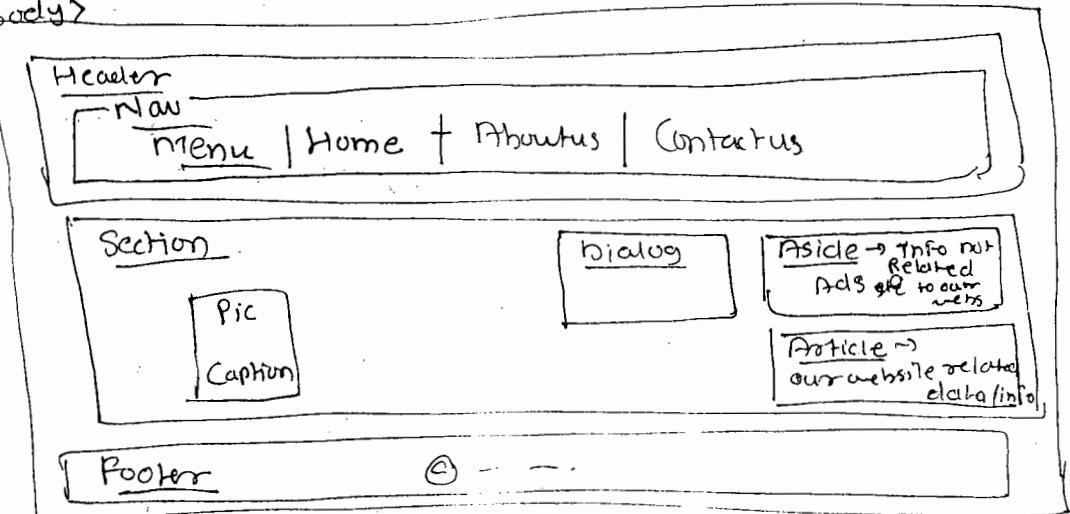
```

HTML 5 (Body) contains all ~~of~~ these new elements

```

<body>
  ① Aside ② Article ③ Header ④ Footer ⑤ Section
  ⑥ menu ⑦ Nav ⑧ Dialog ⑨ figure ⑩ FigCaption ⑪ Div
  ⑫ Span.
</body>

```



- websites are designed using table to organise content.
- The problem is with it "Kiss of Death" -
- If 1st table has to load, 3 rows if table is having all need to load so we use <div> or to solve it.
- <div> have linebreak above & below. & will not have linebreak above & below.
- <div> & are container.
 - ↓
 - comes on newline absolute (in that place only)

HTML Body Elements

| Element | Description |
|--------------|--|
| ① body | Body element specifies information to be displayed in the browser workspace (ie white space where we displays content). HTML introduces several new elements inside body to make it SEO (Search engine optimization) friendly. |
| ② Aside | It contains information that is ^{not} relevant to your website. |
| ③ Article | It contains/publishes information about your website. |
| ④ Header | It specifies the header section that displays information at the top margin of page. |
| ⑤ Footer | It specifies information to be published at the bottom margin of page. |
| ⑥ Section | It defines the area between header & footer. |
| ⑦ Menu | It contains a collection of items that are used for navigation in a website. |
| ⑧ Nav | It describes the navigation area of your website. |
| ⑨ Dialog | It is an area from where user can interact with your website (chatbox/shortcode). |
| ⑩ figure | It specifies any graphic or image area with a caption. |
| ⑪ figCaption | It is used inside the "figure" element to set a <u>caption</u> for image (or) graphic. |

⑫ **div** It specifies a division, which represents a large area with line break before & after

⑬ **span** It specifies a small chunk of code within a large area with no line break before/after

Ex(2) Home.html

```
<!DOCTYPE html>
<html>
  <head>
    <title> Amazon | India </title>
  </head>
  <body>
    <header style="background-color: red; color: white;
      text-align: center; border-style: groove">
      <nav>
        <menu>
          Home <span> | </span>
          About <span> | </span>
          Contact
        </menu>
      </nav>
    </header>
    <section style="background-color: yellow; color: black;
      text-align: center; border-style: groove">
      <dialog open>
        chat with our HR
      </dialog>
      <figure>
        
        <figcaption> Range Rover </figcaption>
      </figure>
    </section>
  </body>
</html>
```

```

<div> Welcome to Amazon
    <div style="background-color: red; color: white">
        Amazon
    </div>
</div>
<aside>
    Ads come here...
</aside>
<article>
    Friday Special Offers...
</article>
</section>
<footer style="background-color: red; color: white;
    text-align: center; border-style: groove">
    &copy Copyright Amazon 2016
</footer>
</body>
</html>

```

[↑]
nbsp → non breakable space (spaces b/w words)

 → line break

<acronym> → Tool Tip / ScreenTips.

[9-1-16]

me → In webpage we need to manage space

<acronym title="HyperText Markup Language">
HTML</acronym>

Note → HTML ignores the line breaks & word spacing defined in the editor (notepad) so in order to add them manually use the following elements.

①
 → for a line break.

② nbsp → for word or character space.

Eg → <div>

First Line
 Second Line

Word1 nbsp Word2

</div>

O/p →

First Line
Second Line
Word1 Word2

- 1)

- 2)
- 3) <acronym>
- 4) <pre>

Note ② ⇒ ScreenTips in HTML :-

You can describe or summarize any content in a page by showing screen tips, instead of defining in the browser window. The element <acronym> is used to set ScreenTips (or Tooltip).

eg → <div>

```
<acronym title = "Hyper Text Markup Language">  
    HTML  
</acronym>  
</div>
```

Preformatted Text → <pre>

HTML provides the element <pre> which is used to display information exactly as entered in the editor. i.e alongwith line breaks & words spacing

eg → <div>

```
    Basic C program:  
</div>  
<div>  
    <pre>  
        #include <stdio.h>  
        main()  
        {  
            printf ("welcome to HTML");  
        }  
</pre>  
</div>
```

→ body attributes → ① bgcolor ② background (image/media) ③ colors
(foreground)
↳ jpg, gif, png

Body elements attributes →

img → sizeless, resolution high
gif → 1 color, small icons.

png → light-weight, size height high, resolution low (Portable n/w graphic)

Body Element Attribute →

- ① bgcolor → sets a background color to the page body section.
- ② background → It sets a background image or any graphic (media)
- ③ color → It sets color for text in body.

Eg → <body bgcolor="#red" color="white">
color</body>

Eg → Image

<body background="images/4.jpg">
</body>

Note → If a page is hosted on web server then it will not support physical path for any resource.

* Eg → <body background="D:\images\4.jpg"></body>
This is not valid.

Headings in HTML →

- ① Headings are required in a web page so that web spiders & web crawlers can make a summary of contents in your website.
- ② You can't use headings to highlight any word or sentence as they have a default line break before and after.
- ③ The tag <hⁿ> is used to designate headings where "n" refers to a number from 1 to 6.

Eg. <div>

<h1> Web Terminology </h1>

The terms used in web development

<h2> Web Server </h2>

Resembles hardware and software, the popular server is

<h3> IIS </h3>

</div>

Heading Attribute →

The heading element supports "align" attribute, which can align the text left, center or right.

Ex:- `<h1 align = "center"> Welcome to HTML </h1>`

Paragraphs and BlockQuotes

① HTML ignores the paragraph marks in a page hence you have to manually designate the paragraphs by using the element `<p>`

② This `<p>` element will have line break before & after the paragraph & it uses an attribute "align", which aligns the text left, center or right or justify.

Ex:- `<p align = "justify">`

paragraph -1

`</p>`

| Paragraph | blockquote |
|-----------|------------|
| align | |
| - left | |
| - right | |
| - center | |
| justify | |

blockquote → A blockquote is similar to paragraph `(p)` but contains left & right indents, an indent will set space between page margin & paragraph.

Ex:- `<blockquote align = "justify">`

blockquote 1

`</blockquote>`

* Text Formatting in HTML

11 Jan 2016

① font ② color ③ fontStyle ④ fontEffects.

① font → attributes →

1) `face` (or `font-family`) we can copy these name from font window of notepad.

2) `size` → (1-7) ← these are levels default is 3.

3) `color`

"small"
"large"
"x-large" etc

2) Color

3) font-style

→ Bold → **b** or **strong**

- italic → *i* or **em**

3) FontEffects

- underline ⇒ u or inserted

- Strikeout ⇒ ~~strike~~ or ~~del~~ deleted

- Superscript ⇒ ^{sup} eg ^{10th}

- Subscript ⇒ _{sub} eg _{H₂O}

* Text Formatting In HTML

① The text formatting includes changing of font style, font style & font effects to control the character appearance.

| Element | Description |
|-------------------------|---|
| ① | Changes the character appearance which includes font <u>face</u> , <u>size</u> & <u>color</u> . |
| ② | Bold |
| ③ | Similar to bold. |
| ④ <i> | Italic |
| ⑤ | similar to italic. |
| ⑥ <u> | underline |
| ⑦ <ins> | Inserted → similar to underline. |
| ⑧ <strike> | Strikeout. |
| ⑨ | deleted - similar to strike |
| ⑩ <sup> | Superscript - Raised from the base line. |
| ⑪ <sub> | lowered from the base line. |

Ex. ① **<!DOCTYPE html>**

<html>

<head> </head>

<body>

<div>

```

H <sub> 2 </sub> O
</div>
<div>
    Nanesh IT 10 <sup>th</sup> Anniversary new offers :
        Java Old fee <del> 4000/- </del> New fee
        <del> 3500/- </del>
    </div> <div>
        <font face="Blackadder ITC" size="7" color="red">
            Welcome to HTML
        </font>
    </div>
    <div>
        <font face="Arial">
            Text formatting in <b><i><u><u> HTML </i></u><b> <br>
            <strong> <em><ins> Javascript </ins> </em>
            </strong> and CSS Basics .
        </font>
    </div>
    </body>
</html>

```

Font Attributes →

- ① size → Specifies the font size in levels from 1 to 7.
- ② face → Specifies the font family name like Arial, Times, New Roman.
- ③ color → Sets a color for Text.

Ex - <div>

 Welcome to HTML

</div>

* Marquee Element

- ① scrollamount → Speed .
- ② ~~anti direction~~
- ③ bgcolor → background
- ④ width → px (or) %
- ⑤ height → px (or) %
- ⑥ loop → No. of time scrolling.
- ⑦ behaviour →

difference b/w px & %?

- % will adjust the text in browser even if we resize the window.
- px will not adjust the

- ① The marquee element is used to set scrolling of the Sliding titles in a web page.
- ② You can manipulate the marquee by using CSS & Javascript.

Attributes Description

- ① scrollamount Specifies marqueses speed.
- ② bgcolor Sets a background color.
- ③ direction Specifies scrolling direction, ie. left, right, up and down.
- ④ width Sets width for marquee area in pixels or percentile.
- ⑤ height Sets height for marquee area in pixels or percentile.
- ⑥ loop Specifies the total no. of times marquee content element must appear.
- ⑦ behaviour Sets alternative behaviour for marquee.

Eg:- ① Alternate behaviour

```
<div>
  <marquee behaviour="alternate" scrollamount="10">
    <h1>Amazon | India </h1>
  </marquee>
</div>
```

Eg:- ② Marquee Redirection and other attributes

```
<marquee scrollamount="10" bgcolor="red"
  direction="left" width="50%" height="50%
  loop="5">
```

```
<font face="arial" color="white">
```

Amazon offers 50% on HIFC and Axis credit

Images In HTML

→ HTML 5 introduces some few new elements to display images in a web page they are →

| Element | Description |
|----------------|---|
| ① <figure> | → Encapsulates any image or graphic with caption. |
| ② <figCaption> | → Set a caption for image or graphic. |
| ③ | → allocates a location for image in webpage. |

Image Attributes :-

| Attribute | Description |
|-----------|--|
| ① src | → Specifies name & path of image to be displayed. |
| ② width | → Set image width. |
| ③ Height | → Sets image height. |
| ④ border | → sets a border for image. |
| ⑤ align | → align the image left or right. Text will wrap around image only when align is specified. |
| ⑥ alt | → Alternative text, to be displayed when image fails to load. |

<div>

<figure>

<img

src = "images/car.jpg" align = "left"

width = "200" height = "100"

alt = "Audi-Q7" border = "1">

<figCaption> Audi - Q7 </figCaption>

</figure>

</div>

LIST In HTML

① Ordered List

② Unordered List

③ Data List

→ A list Comprises of collection of items with bullets and numbering. However it will not update automatically when new item is added or removed from list.

→ HTML provide following list →

① Ordered List →

→ This a collection of items with autonumbering

→ The tag is used to encapsulate all items and every item in the list is designed with the tag

Attributes →

① ~~style~~ type → indicates the numbering style for list items like a, b, c or i, ii, iii or 1, 2, 3, A, B, C.

② Start → Indicates the level no. from where numbering have to start.

Ex:- <h2> Web Terminology </h2>
<ol type="A" start="1">
 Web Server
 Web Site
 Web Application

O/P → A. Web Server
 B. web Site
 C. web Application

type can be →

① type = "A"

② type = "a"

③ type = "i"

④ type = "I"

⑤ type = "l"

↓
no need to mention it bcz default type is numbers only.

② Unordered List → It is similar to ordered list
contains collection of list items with a bullet symbol
like circle, disc, square

Ex:- <h2> Web Technology </h2>
<ul type="square">
 Web Server
 Website

O/P →
■ Web Server
■ Website.

③ Data List → A data list contains collection of data items and data definitions. This is used to represent a collection of terms and definition.

| Element | Description |
|---------|------------------------------|
| ① <dl> | Collection of data terms. |
| ② <dt> | Specifies data term. |
| ③ <dd> | Definition for any data term |

Ex:- <h2> web Terminology </h2>

<dl>
 <dt> Web Server </dt>
 <dd> It resembles hardware & Software </dd>
 <dt> Web Site </dt>
 <dd> It is Virtual directory on server </dd>
</dl>
</dl>

→ HTML 5 introduces a new data list with forms element which is collection of options linked with any form element so that it will auto-complete the terms. you are typing in the textarea.

d) Ex:- <form>
<input type="text" size="40"
list="terms">
<datalist id="terms">
<option> Sachin </option>
<option> David </option>
<option> Sourav </option>
</datalist>
</form>

1>

2>

3>

Card - cash Back

Every Scrolling & Sliding is
not a Marquee; we can do it
using Flash or Animation.

</marquee>

</div>

| | | | |
|-----------------------------------|---|--------|---------|
| Google & search in Adsense Google | Notes Written in ending Topic - List, Image | Absent | 18-1-16 |
|-----------------------------------|---|--------|---------|

→ it is a reference operator.

e.g. → How to remove the underline of Hyperlink?

Ans → using styles → i.e. style="text-decoration: none;"

Hyperlink →

→ A hyperlink is clickable text, picture or graphic.

that links to any named location within the same document or any another document.

→ Hyperlinks are one of the navigation techniques used in web pages to navigate the user from one location to another.

→ Hyperlinks are classified into 2 types:

① Intra document links.

② Inter document links.

→ HTML provides "Anchor Element" to define hyperlinks & the anchor is specified by using the tag <a>

Attributes →

① href : Specifies the target url to navigate when link clicked.

② target: specifies the target location where the linked content should open. i.e. in the same window, new window, frame etc.

Syntax:-

 Text

Eg:- Intra Document links Intra document links It specifies links that navigate the user to any named location within the same page.

This includes following Steps →
Step①: Specify anchor for any location in the Page.

Eg: ① <h2 id="html" > HTML Basics </h2>

② <figure id="fig1">

</figure>

Step②: Link the element to anchor by referring to its Id.

Eg: HTML Basics

 Car

Eg: Tutorial.html

```
<!DOCTYPE html>
<html>
<head> </head>
<body>
<div>
<h1 align="center" id="toc">Table of Contents </h1>
<ol>
  <li><a href="#html"> HTML Basics </a></li>
  <li><a href="#Javascript"> JavaScript Examples </a></li>
</ol>
</div>
<div>
<h2 id="html"> HTML Basics </h2>
<a href="#toc">
  
</a>
<p> </p>
</div>
```

MIME types.

<h2 id="javascript"> JavaScript Examples </h2>

```
<a href="#toc">  
    
</a>  
<p> - - - - -  
</p>  
</div> </body>  
</html>
```

Inter Document Links

- It refers to link that navigate the user to any another document or specified url.
- If server supports the MIME Type of linked file then it can read or download file by using a hyperlink.

Eg → links.html

```
<!DOCTYPE html>  
<html>  
<head></head>  
<body>  
<div>  
<ol>  
  <li><a href="http://www.google.com" target="_blank">  
    Google ! </a> </li>  
  <li><a href="tutorial.html"> HTML Tutorial </a></li>  
  <li><a href="Content/adobe.exe" > Download Adobe  
    Reader </a></li>  
  <li><a href="Content/adobe.asp.pdf"> Read Asp PDF </a>  
  <li><a href="images/1.jpg"> Audi-Q7 </a></li>  
</ol>  
</div>  
</body>  
</html>
```

Note → How to remove underline for a hyperlink?
 ⇒ By using style

```
<a href = "url" style = "text-decoration : none;">  
</a>
```

Frames in HTML.

- frames are used to split the browser window into horizontal & vertical panes. So that every frame represents a window & allows to access & display multiple pages in a single browser.
- The following elements are used to create & manipulate frames.

| Element | Description: |
|------------|---|
| <frameset> | It is a collection of frames organized into rows and columns. |
| <frame> | It represents individual frame, which can display any document as source. |

Syntax:-

```
<frameset rows/cols = "percentile">  
  <frame name = "frame1" src = page1.html>  
</frameset>
```

Note → A HTML Document with frames can't use body elements. you have to display any content using another HTML Document.

Eg:-

- Add following HTML Pages into your website.
 - index.html
 - menu.html
 - footer.html
 - home.html
 - about.html
 - Contact.html.

- ② index.html (source code)

```
<!DOCTYPE html>  
<html>  
<head><title>Amazon</title></head>
```

```

<frameset rows="15%, 70%, 15%" noresize>
  <frame src="menu.html" name="framehead">
  </frame>
  <frame src="home.html" name="framebody">
  </frame>
  <frame src="footer.html" name="framefooter">
  </frame>
</frameset>
</html>

```

③ menu.html (Source code)

```

<!DOCTYPE html>
<html>
  <head> </head>
  <body>
    <header>
      <nav>
        <menu style="text-align: center">
          <a href="home.html" target="framebody">
            
          </a> <span> | </span>
          <a href="about.html" target="framebody">
            
          </a> <span> | </span>
          <a href="contact.html" target="framebody">
            
          </a>
        </menu>
      </header>
      <nav>
        <menu style="text-align: center">
          <a href="#"> Home </a>
          <a href="#"> About </a>
          <a href="#"> Contact </a>
        </menu>
      </nav>
    </body>
  </html>

```

④ footer.html (Source code)

```
<!DOCTYPE html>
<html><head></head>
<body>
<h2 align="center">© Copyright 2006 </h2>
</body>
</html>
```

⑤ Add any content into the pages

"home.html", "about.html", "contact.html"

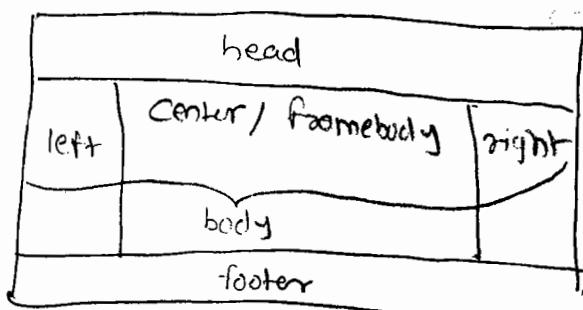
Nested frames.

→ A frameset in html can use another frameset hence it is possible to create a nested frames.

Eg: <!DOCTYPE html>

```
<html>
<head></head>
①<frameset rows="15%, 70%, 15%">
  <frame src=" " name="head" /></frame>
②<frameset cols="15%, 70%, 15%">
  <frame src="menu.html" name="left" /></frame>
  <frame src=" " name="center" /></frame>
  <frame src="ads.html" name="right" /></frame>
</frameset>
<frame src=" " name="body" /></frame>
<frame src=" " name="footer" /></frame>
</frameset>
</html>
```

O/P →



Iframes in HTML →

- Iframes are similar to frames but they are embedded into the document body. We can access & display any page or domain as a source in iframe.

Syntax:-

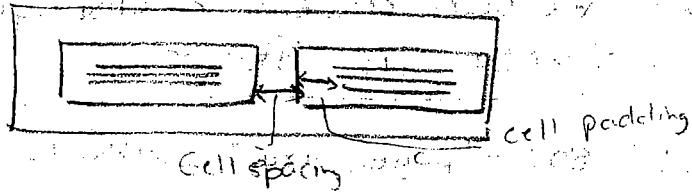
```
<iframe name="frame1" src="page/url" width="400"
        height="100">
</iframe>
```

2+1/16

Ex:- for iframe [Iframeindex.html]

```
<!DOCTYPE html>
<html>
<head> <title> Amazon </title> </head>
<body>
<header>
<nau>
<menu style="text-align:center">
  <a href = "home.html" target="framebody" > Home </a>
  <span> | </span>
  <a href = "about.html" target="framebody" > ABOUT </a>
  <span> | </span>
  <a href = "Contact.html" target="framebody" > Contact </a>
  <span style="float:right"> CONTACT </span>
</menu>
</nau>
</header>
<section>
  <iframe src="home.html" name="framebody"
          width="550" height="250"> </iframe>
</section>
<footer style="text-align:center"> © Copy Right 2006
</footer>
</body>
</html>
```

Web debugger → F12 → Network → click on it.
See the time.



Tables in HTML

- ① Tables are used in web applications to organize information into rows & columns.
- ② The following elements are used to create & organize data in tables.

| Element | Description |
|---------|---|
| <table> | Specifies the start & end of the table. |
| <th> | Specifies the table header. |
| <tr> | Specifies the table row. |
| <td> | Specifies the table cell. |

| Attribute | Description |
|---------------|--|
| ① border | Sets border for table. |
| ② align | Aligns table, tr and td to left, center or right. |
| ③ valign | Aligns tr cell content vertically top, center or bottom. |
| ④ width | Sets width for table. |
| ⑤ height | Sets height for table and table row. |
| ⑥ bgcolor | Sets background color for table, tr and td. |
| ⑦ background | Sets background image for table, tr and td. |
| ⑧ cellpadding | Sets spacing between text and cell border. |
| ⑨ cellspacing | Sets spacing between cells. |
| ⑩ colspan | Merges columns into single cell. |
| ⑪ rowspan | Merges rows into single cell. |

Eze → **Products.html**

web Spider
web scroller

```
<div>
<table border="1" width="300" height="100"
       bgcolor="yellow" align="center"
       cellspacing="10" cellpadding="10">
<tr align="center">
<td colspan="2"> Name </td>
<td colspan="2"> Address </td>
<td rowspan="3"> Student Info </td>
</tr>
<tr>
<td> first name </td>
<td> Last name </td>
<td> City </td>
<td> Pincode </td>
</tr>
<tr>
<td> Raj </td>
<td> Kumar </td>
<td> Hyd </td>
<td> 500079 </td>
</tr>
</table>
</div>
```

O/P →

| Name | Address | StudentInfo | | |
|------------|-----------|-------------|---------|--|
| first name | Last name | City | Pincode | |
| Raj | Kumar | Hyd | 500079 | |

New HTML 5 Table Elements.

Elements

- ① <thead>
- ② <tbody>
- ③ <tfoot>
- ④ <caption>
- ⑤ <colgroup>

New HTML 5 Table attributes.

| Attribute | Description |
|-----------|--|
| ① rules | Sets rules for table which includes all, rows, cols, groups. |
| ② frame | Sets a frame for table which includes border, rows, lhs, above, below, void. |

Ex.

html

<table rules = "groups" frame = "border">

<caption> Table - 2.1 </caption>

<colgroup span = "2" style = "background-color: blue">

<thead> if we apply style then only we can see

<th> Product ID </th> the effect of colgroup

<th> Name </th>

<th> Price </th>

<thead>

<tbody>

<tr>

<td> 1 </td>

<td> Mobile </td>

<td> 13000 </td>

</tr>

<tr>

<td> 2 </td>

<td> LED TV </td>

<td> 43000 </td>

</tr>

</tbody>

<tfoot>

<tr>

<td colspan = "2"> Total Amount </td>

<td> 45000 </td>

</tr>

</tfoot>

<table> Use → <http://localhost/Amazon/shopping.html>
</div>

22nd Jan

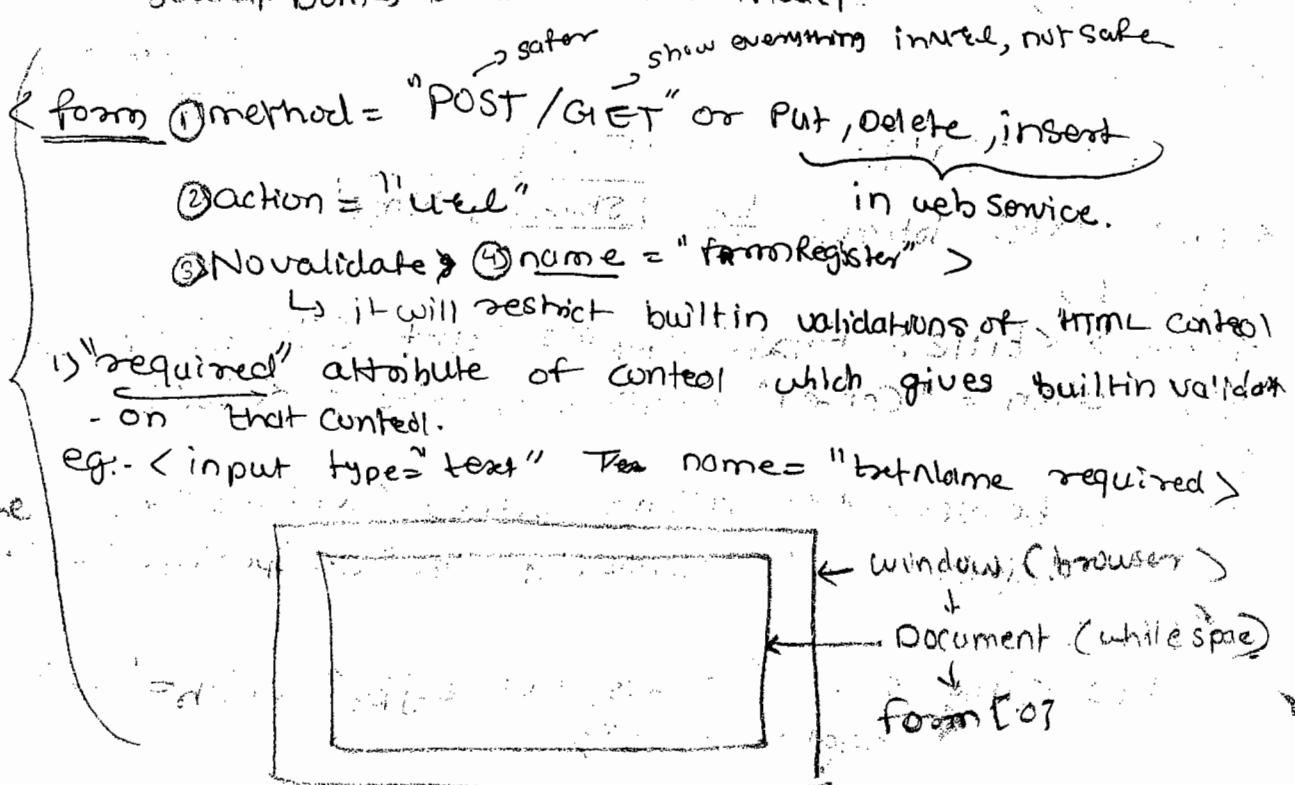
Nested Tables in HTML
<!DOCTYPE html> Ex: [shopping.html](#)

```
<html>
<head> <title> Amazon Shopping Cart </title> </head>
<h1 align = "center"> Amazon Offers </h1>
<div>
    <table border = "1" align = "center" width = "300"
           height = "200" cellpadding = "20" cellspacing = "10">
        <tr>
            <td> <img src = "images/mobile.jpg" width = "200"
                   height = "200">
            </td>
            <td>
                <table border = "1" width = "200" bgcolor = "yellow">
                    <tr>
                        <td> Product ID </td>
                        <td> 101 </td>
                    <tr>
                        <td> Product Name </td>
                        <td> LG Mobile </td>
                    <tr>
                        <td> Product Price </td>
                        <td> ₹ 13000 </td>
                    <tr>
                        <td colspan = "2" style = "text-align: center;">Amazon offers
                </table>
            </td>
        </tr>
    </table>
</div>
</body>
</html>
```

| | |
|--------------|-----------|
| img | mobile |
| Product ID | 101 |
| Product Name | LG mobile |

row → top outer table inner table

JavaScript DOM → Document Object Model



Forms in HTML

- A form is collection of controls that allows interaction with the user every web application allows users to interact with it by using a webform.
- A Typical web form contains Controls like Buttons, TextBox, CheckBox, RadioButton, Listbox etc.
- HTML provides the following elements to design a form →

Element Description

`<form>` Specifies the start and end of a form in a web page. It encapsulates all the controls in page.

`<input>` Adds an element into the form like Button, TextBox, checkbox etc.

`<select>` Adds a Listbox or Dropdownlistbox

`<textarea>` Adds a textarea with multiline property.

`<datalist>` Adds a datalist with collection of options displayed as autocomplete.

metacharacters & Quantifier → Pattern

Registration.html

The <form> tag → It is used to encapsulate all the controls in a webpage and manage the page on various request types (ie Request headers) (those are get, post, put, delete).

Attributes of <form> tag →

Attribute Description

① Method : Specifies the request header GET
GET, POST, PUT, DELETE, INSERT etc

② Action : Specifies the target URL to navigate when form is submitted.

③ novalidate : Disables the Default HTML Validation on Page.

④ Name : Sets the Reference name for form.

Syntax:-

```
<form name="formHome" method="POST" default method is GET  
action="home.html" novalidate>  
</form>
```

Pattern

Meta characters

[a-z]
[A-Z]
[a-zA-Z] or [a-Z]
[0-9]
[a-zA-Z0-9_] or \w
[a,s,z]
[^a,s,z]
+ . - . @ \\$
[@ # \$. &]
(?=.*[A-Z])
(?=.*[0-9])
(?=.*[!@#\$%^&])
[a-zA-M4-9]

Quantifiers.

{4} → exactly 4
{4,10} → min 4 max 10
{4, } → min 4 max n

① Pattern
Metacharacters
[a-z]
[A-Z]
[a-zA-Z] [a-Z]
[0-9]
[a-zA-Z0-9_] or
[a,s,z]
[^a,s,z]
+ . - . @ \\$
[@ # \$. &]
(?=.*[A-Z])
(?=.*[0-9])
(?=.*[!@#\$%^&])
[a-zA-M4-9]
② Pattern
min 4 max 10
min 4 max 10
min 4 max 10
min 4 max 10
(+) 1 [0-9] {1,10}
+ 982345100

Ex → Pattern

+91[0-9]{10} → +919876543210

Form Controls

[1] TextBox → The TextBox Control enables the user to input, view and edit the text on page dynamically during runtime.

Syntax:-

```
<input type="text">
```

Attributes :-

| Attribute | Description |
|-----------|-------------|
|-----------|-------------|

- ① name : Specifies a reference name for TextBox.
- ② value : Specifies the value present in TextBox.
- ③ size : Sets width for TextBox.
- ④ maxlength : Specifies the maximum no. of characters allowed.
- ⑤ readonly : Disables editing, hence readonly.
- ⑥ placeholder : sets a watermark text for TextBox.
- ⑦ autofocus : Sets focus on TextBox automatically when page loaded.
- ⑧ required : Specifies that TextBox must have a value and can't be null or empty.
 - will not work if we set "novalidate" for forms
- ⑨ pattern : Validates the input value by matching it against the pattern.
 - Patterns are build by using metacharacters and quantifiers.
- ⑩ list : Attaches a datalist to TextBox.

Note → "required" and "Pattern" will not work if we set "novalidate" for <form> tag or form → "required" & "Pattern" will only work when we click Submit button or we submit the page

```

<div>
  <input type = "text"
    name = "factUsername"
    autofocus
    required
    pattern = "(?=.*[A-Z])\w{4,15}"
    placeholder = "Name up to 15 characters with
      atleast one uppercase letter">
</div>

```

* Creating Patterns using Metacharacters & Quantifiers

| Meta Characters | Description |
|------------------------|--|
| 1) [a-z] | - Only lower case letters |
| 2) [A-Z] | - Only upper case letters |
| 3) [0-9] | - Only numeric |
| 4) \d | - Only numeric (0-9) digits. |
| 5) [a-zA-Z] @ [a-zA-Z] | - Both upper & lower case letters. |
| 6) [a-zA-Z0-9_] | - Alpha numeric with an underscore |
| 7) [a-zA-M4-9] | - Characters in specified range. are allowed |
| 8) [a,d,s] | - Only Specified characters allowed. |
| 9) [^a,d,s] | - Excluding Specified Characters all other characters allowed |
| 10) !@#\$%^& | - Individual special characters |
| 11) [!@#\$%^&] | - Special characters allowed. |
| 12) (?=.*[A-Z]) | - Atleast one uppercase letter |
| (?=.*[a-z]) | - Atleast 1 lower case letter. |
| (?=.*[0-9]) | - Atleast 1 numeric digit. |
| (?=.*[!@#\$%^&]) | - Atleast 1 special characters. |

| Quantifier | Description |
|------------|---|
| ① {n} | - Exactly n - numbers of chars |
| ② {n,m} | - minimum \Rightarrow n and maximum \Rightarrow m |
| ③ {n,} | - minimum \Rightarrow n and maximum \Rightarrow any |

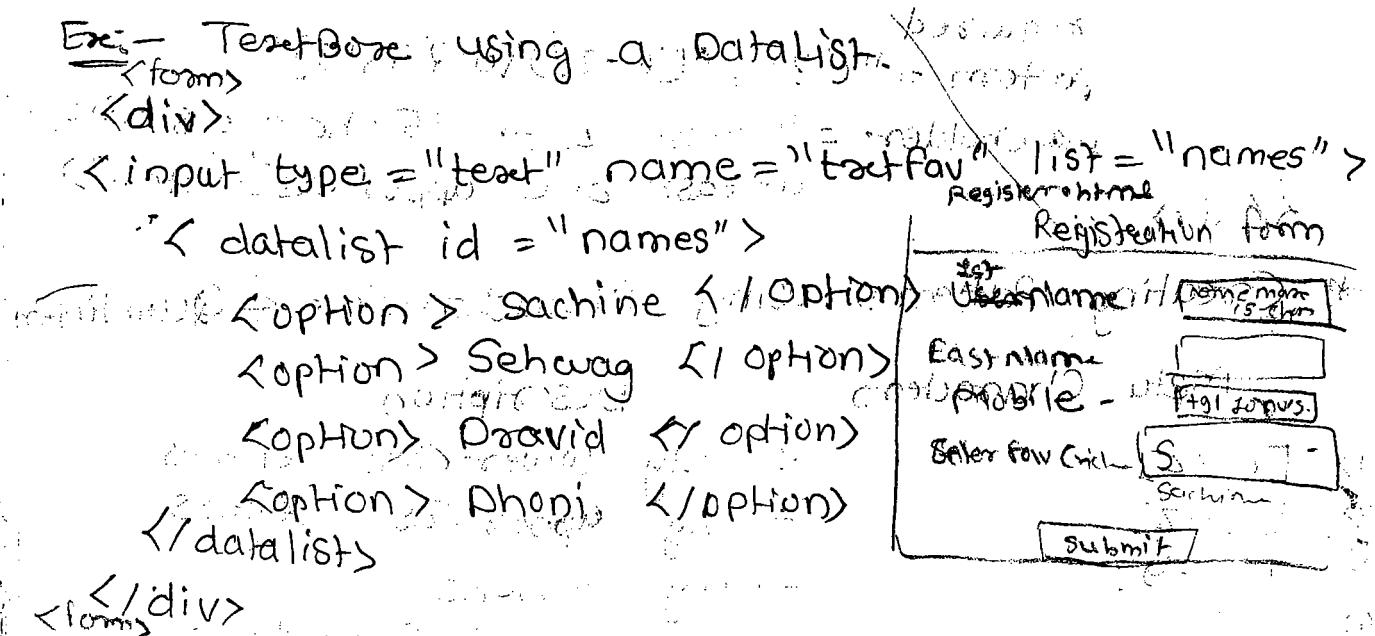
Ex:- Validating 10 digit mobile number starting with +91.

⇒ <input type="text" name="tactmobile" pattern="^+91[0-9]{10}">

Ex:- Textbox using a Datalist.

```
<form>
  <div>
    <input type="text" name="tactfav" list="names" />
    <datalist id="names">
      <option> Sachin
      <option> Sehwag
      <option> David
      <option> Rohit
    </datalist>
  </div>
</form>
```

Registration form



* HTML input Controls that belongs to Textbox family

Input Type Description

- ① Password - Marks the input value with password char "*" or "•"
- ② Date - Shows the textbox with date picker, so that user can select a date.
- ③ Week - Shows a week Picker.
- ④ Year - Shows a year picker.
- ⑤ Url - sets URL validation for textbox.
- ⑥ Email - sets Email validation for textbox.
- ⑦ Color - Shows a color Picker.
- ⑧ Number - Sets a numeric Up/ Down for textbox with min and max values.
- ⑨ Range - Shows a range Slider so that user can select any value from specified range.
- ⑩ File - Shows Open file dialog control that allows the user to select a file for uploading.

① Thread ② Mutex ③ Semaphore ← multi-threading
Ex → <div> ↗ only 1 task can use a process at a time
 Date of Birth : process ↗ 2 tasks can use a process at a time
 ↗ eg. windows media player ↗ eg. youtube downloader (TubeMate)
 ↗ <input type = "date" name = "txtDOB" /> 3 videos
 ↗ we can download

</div>
 <div>
 Your Email :
 <input type = "email" name = "txtEmail" />
 ↗ mutual exclusive

</div>
 <div>
 Password:
 <input type = "password" name = "txtPwd" />

</div>
 <div>
 Age :
 <input type = "number" name = "txtAge" />
 ↗ min = "15" max = "30"

</div>
 <div>
 Upload photo:
 <input type = "file" name = "txtPhoto" />

↗ note → All new browser modes are not available with
 browsers like "Internet Explorer". (Best Recommended
 browser by w3 is "Safari").

[2] Radio Button Control → Mutex
 → Radio Buttons enables the user to select any one option from a group of choices.
 → The radio buttons exhibits exhibit "MUTEX"
 (mutual Exclusion), which allows you to Deselect any option only by selecting another option.

Syntax → <div>
 Select Gender :
 <input type = "radio" name = "gender" value = "male" />
 ↗ Male ← in
 <input type = "radio" name = "gender" value = "female" />
 ↗ Female

Note →

In order to group the radio buttons into one category you have to define a common name.

[3] CheckBox Control

25-Jan-16

The CheckBox Control enables the user to select multiple option from a group of choices.

Eg:- <div>

Your Hobbies :

```
<input type="checkbox" name="music"  
      value="Music"> Music  
<input type="checkbox" name="movies"  
      value="Movies"> Movies.
```

Note → The "checked" properties specifies if the radio button or checkbox are selected or not.

The Default property for "checked" = true.

* CheckBoxList is not available in HTML. So we can achieve this by say using CSS.

[4] DropDownList Control

→ A dropdownList or a comboBox displays a drop-down menu and enables the user to select any one option from a group of choices.

Element

Description

<select>

Adds a dropdownList.

<option>

Specifies items in dropdownList.

Eg → <div> Select your City:

```
<select> → don't have size & multiple attribute  
<option name="SelectCity" value="-1">Select a City  
</option>  
<option name="Delhi" value="115005"> Delhi</option>  
<option name="Hyd" value="590082"> Hyderabad</option>  
</select>  
</div>
```

[5] Listbox Control →

- A Listbox is similar to dropdown but enables the user to select more than 1 option from a group of choices.
- To select multiple options you have to use "ctrl" or "shift" keys. Ctrl will select random and Shift will select in sequence. series.

Ex:- <div> Select your City :

```
<select size="2" multiple>
  <option name="delhi" value="110003"> Delhi</option>
  <option name="hyd" value="500093"> Hyderabad</option>
</select>
</div>
```

Note → If <select> tag is having "size" & "multiple" attribute then it is Listbox & if we don't have those attribute then it is DropDownList/combobox.

[6] TextArea Control →

- TextArea is similar to TextBox Control Except that it shows the text in multiple lines.
- It is an RCOntrol Element that treats the elements inside it as normal text.

① Ex :- <div> Address :

```
<textarea rows="10" cols="40"></textarea>
</div>
```

② <div> License Terms :

```
<textarea readonly rows="10" cols="40">
  - - - License Terms - - -
</textarea>
</div>
```

[7] Meter →

- A meter inserts gauge meter into a web page that can display changing values dynamically during runtime in a meter form. (ie Strong p, weak p in green, Red, Range)

(e) `<div>` `<meter min="1" max="100" value="50">/meter</div>`

[8] Progress →

A progress Bar is used in webforms to show the status of any task performed like downloading, uploading, installing, copying etc.

`<div>` `<progress min="1" max="100" value="50">/progress</div>`

[9] Button → Buttons are used to perform Record Actions
Record Navigation (previous, next, etc.)
& Miscellaneous Actions (open, print, choose, etc.)

→ HTML provides 3 types of buttons

Button Type Description

① Submit :- Submit the form data to server.
for data is passed into the address bar as query string.
★ Max. no. we can pass the no. of values ie in query string is 2048 (Estandard)
- for Mozilla: 65535
- for Opera: 1,90,000

② Reset :- It resets the form data by Configuring all controls to their Default values.

③ Button :- It is used for user defined actions

foreign Element → Svg, MathML, RTF

Ex:- <input type="submit" value="Submit" name="Submit">
<input type="reset" value="Reset" name="Reset">
<input type="button" value="Print" name="btnPrint">

Note → Every Control should have name. because when we submit or post the data then the values are not seen in Query String in URL. bcz in query string value displayed as name value.

26-1-16

Styles in HTML

→ Styles are set of attributes defined to the elements in HTML to make them more interactive and responsive. In web pages styles can be defined by using following methods ⇒

- ① Inline Styles.
- ② Embedded Styles
- ③ CSS (Cascade Style Sheets)

① Inline Styles → The styles are defined within the tag by using "style" attribute. These are individual to every element and are not accessible from any another element.

Ex:- <h1 style="background-color: red; color: white; text-align: center; border-style: dashed;">

Styles in HTML

</h1>

② Embedded styles → The styles are defined in the head section of a page by using a style element so that these styles are accessible to any element within the same page. but are not available to any element on other pages.

Ex:- <head>
 <style>

```
h1
{
    background-color: green;
    color: white;
    text-align: center;
}

</style>
</head>
<body>
    <h1> HTML </h1>
    <h1> CSS </h1>
</body>
```

③ CSS (Cascade Style Sheets) → The styles are defined in a separate style sheet so that they are accessible from any page within the website or from any other website.

Ex: Creating a new Style sheet.

- ① Open notepad.
- ② Type the following code.

```
h1
{
    background-color: blue;
    color: white;
    text-align: center;
}
```

③ Save the file into your website Styles folder by name "headings.css".

- link StyleSheet to your page
- ① Go To your HTML Page "tutorial.html"
 - ② Link the style sheet.

```
<head>
    <link rel="stylesheet" href="styles/headings.css">
</head>
```

online minification Tools → To minify the code CSS / JS.
eg. Refresh.js, smalify. → to reduce the size of file CSS / JS.

If our site contain CSS file & if we send the request for a webpage then request will go for webpage & also for CSS file. means when we send request to 1 page, 2 request will go to the server. If our page is linked with 5 CSS files then 6 request will go to the server. So drawback with CSS is for every CSS with 1 request will be send to the server. So see this by using CSS for a web page, start debugger & refresh webpage & see in the debugger.

• net → VS → we have minification plugins
we can use them to minify CSS / JS files

We can see 2 requests has been send to the server.

Minification → It is the process of reducing size of CSS & Javascript files. not by compressing them but by removing blank spaces, linebreaks, shortening the lengthy variable names, and using various programming logics. (eg ternary operators in the place of if...else) → $(? :)$

→ minification can be handled by using minification tools like Smalify, WebEssentials, AjaxMinify etc.

Ex → Step to Minify CSS / JS.

- ① Create a new StyleSheet by name "headings.css" and save in Styles folder of your website.
- ② Install any minification Tool on your Computer.
eg Smalify.
- ③ Drag & drop the CSS file folder into "Smalify".
- ④ Click "Minify now" button.
- ⑤ This will generate the minified file
ie "headings.min.css" here min is not extension just a part of name.

⑥ Link the "minified" file to your page.

<head>

```
<link rel="stylesheet" href="styles/headings.min.css">
```

</head>

→ Bootstrap → Largest Repository of CSS and JavaScript.

→ Same as Bootstrap we have 3rd party plugin's.

- NPM { These plugins are used in AngularJS } AngularJS → Google
- BOWER { Used in AngularJS } jQuery → Mozilla
- GRUNT { Javascript minification } Javascript → Sun Microsystems
- GROWLIE { Alert box } Growl

→ How to install Bootstrap?

www.getbootstrap.com

→ Bootstrap CDN ⇒ It is online library.

27-1-10

Bootstrap

① Bootstrap is one of the largest Repositories of CSS and JavaScript. It provides several Templates that you can implement in your website.

Step ① Enable Bootstrap for your website.

- ① Visit "http://www.getbootstrap.com"
- ② Click "download Bootstrap".
- ③ This will download a bootstrap.zip, which contains the files - bootstrap.min.css
- bootstrap.min.js.
- ④ Copy these files and paste them into your website "physical path" (Create a folder by name bootstrap).
ie. D:\Amazon\website\bootstrap

bootstrap.min.css bootstrap.min.js

Step-② Use Bootstrap template for your website.

- ① Visit "getbootstrap.com".
- ② Click on CSS category.
- ③ Click on any desired Example like Buttons, Menu, GridSystem etc...
- ④ For Every Example it provides
 - a) CSS Code
 - b) HTML Code
- ⑤ Copy the CSS Code into a CSS file in your website.
- ⑥ Copy and copy the HTML code into your web page.
- ⑦ Link the following files to web page.

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="bootstrap/bootstrap.min.css">
  <script src="bootstrap/bootstrap.min.js">
</head>
</html>
```

* Syntax For Styles [Embedded / CSS]

Selector
 {
 attribute 1 : value ;
 attribute 2 : value ;
 }

Various Types of Selectors →

- ① Type Selector
- ② ID Selector
- ③ Class Selector
- ④ Descendent Selector
- ⑤ Attribute Selector

① Type Selector → It refers to the name of the element to which you want to apply the styles. All similar elements in Page will acquire the styles.

Ex → <head>

{style} → grouping of selectors

h1, p
}

background-color: red;

color: white;

}

</style>

</head>

<body>

<h1> HTML Basics </h1>

<p> Paragraph - 1 </p>

</body>

② ID Selector → An ID Selector uses "# symbol" for reference for styles so that any element can use the styles by referring to an ID. However every element can use only 1 ID, hence multiple styles can not be applied through ID reference.

Ex → <head>

<style>

#headings

{

background-color: red;

color: white;

}

</style>

</head>

<body>

<h1 id="headings"> HTML Basics </h1>

<p id="headings"> Paragraph - 1 </p>

</body>

③ Class Selector → The class selectors are defined by using a ":" symbol / reference, and they are referred by using the keyword "class".
→ One of the major advantage of class selector is every element can access multiple classes.

Eg:- <head>

<style>

• backstyle

{ background-color: red;

}

• textstyle

{

color: white;

text-align: center;

}

</style>

</head>

<body>

<h1 class = "backstyle textstyle"> HTML Basics </h1>

</body>

④ Descendent Selector → It refers to a child element of any specific parent element.

→ It is used mostly when same child element is being used by several elements in page.

Eg:- <head>

<style>

ol > li

{ color: red;

}

</style>

</head>

<body>

 Item-1

 Item-2

</body>

Note → If we write also in <body> then style will be applied only to list items not list items. because it is parent specific,
ie ol > li
{ color: red.
}

⑤ Attribute Selector → Styles can be applied to any element based on a specific attribute so that any element using that attribute can ^{inherit} the styles.

```
Eg → <head>
    <style>
        input [type = text]
        {
            background-color: yellow;
        }
    </style>
<head>
<body>
    <form name = "form1">
        Name : <input type = "text" > <br> input [type = text]
        Email : <input type = "text" >
    </form>
</body>
```

Details in depth about Styles.

- ① Background Styles ↗
color
image
position
repeat etc.
- ② Text Styles ↗
color
font-family
font-size
font-weight
text-decoration
text-align
text-indent
text-decoration
text-transform
- ③ Back-Border Styles ↗
border
border-color
border-width
border-style
border-radius
border-collapse

<http://Sharma-nareshit.blogspot.in>

↳ Downloads.

↳ All our Classroom Demos.

Note → Even if we write
2 buttons in <form> as
those are also <input>
attribute, style will not be
applied to buttons bcz.

JavaScript

28/1/2016

- Javascript [Popular]
 - ↳ Jquery
 - ↳ Angular Js → client side
 - ↳ Node Js → mvc.
 - ↳ Backbone Js
 - ↳ JavaScript Spines.
 - ↳ Knockout Js.

- Javascript is used in many places
 - ↳ web developments
 - ↳ Animation / Graphics / special effects
- Javascript
 - 1. Client Side Validation
 - 2. Client Side Interaction

www.brendaneich.com → Brendan Eich → JavaScript Designer.

mocha → LiveScript → Javascript (named by JavaSun microsystems)
↳ No Relation with Java
Programming Systems →

- ① POPS ② OOPS ③ COPS.

① Process oriented Programming System
advantages → they will directly interact with hardware.
↳ less memory & faster, eg. embedded system, C.

- ② Object based programming System -

advantages → Readability & code separation
disadvantages → not support inheritance & dynamic polymorphism.
↳ extensibility & customization is not possible.

eg → VB, Javascript, VB Script.

- ③ Object oriented programming system.
↳ Code is complex & need more memory.

features of JavaScript

- ① Code Reusability

it will not have any compiler

- ② Code Separation

- ③ Light weight

Interpreted Programming

- ④ Reduce Complexity

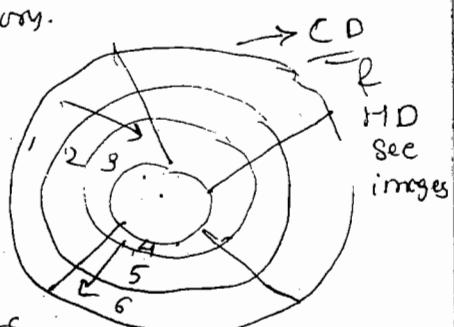
bcoz Javascript treats every element like an array

- ⑤ Reduce Overhead

to reduce overhead

→ array is used to store values in sequential manner

How → by storing multiple values under same name



disc defragmentation

Client Side Script

- A client side script is used in web applications to control client side interactions and client side validations.
- Validations are required in web application to ensure that Contadictionary and unauthorized data is not get stored into the database.
- A client side script is a script that runs on the client machine and native to the client browser.
- The 1st Client Side Script was introduced in 1995 by Brendan Eich, which was named as "mocha". Later it adopted several new features and was renamed as "LiveScript". Sun Microsystems purchased the rights of LiveScript and named it as "JavaScript". The other client side scripts are
 - ① → VBScript
 - ② → Jquery
 - ③ → Angularjs
 - ④ → Knockoutjs
 - ⑤ → Backbone JS
 - ⑥ → JavaScript Spines
 - ⑦ → Nodejs

Key Features of Javascript

- Javascript is an object based programming system (OBPS). It supports code reusability & code separation, however any object OBPS will not support inheritance & dynamic polymorphism.
- Javascript is an interpreted programming language, which is integrated with into other application environment. Some of the key features of Javascript are →
 - ① Code Reusability
 - ② Code Separation
 - ③ Light weight
 - ④ Reduce Complexity
 - ⑤ Reduce Overhead

Note →
→ Primary goal of inheritance is Extensibility not reusability. However we can reuse the code.
→ By just creating object we can reuse the code.

Drawback of JavaScript

- ① Not Secured Cuz present in Client Browser
- ② Can't interact with backend database → SOLR AngularJS Backbone knockout JS
- ③ It can be disabled on any browser. & client side script can be bypassed.

Note → How to check whether JavaScript is enabled or not Client Browser? → `<noscript>` element msg

Ans → Using `<noscript>` element → will display only if client browser's JavaScript is disabled

Ex :- `<body>`

```

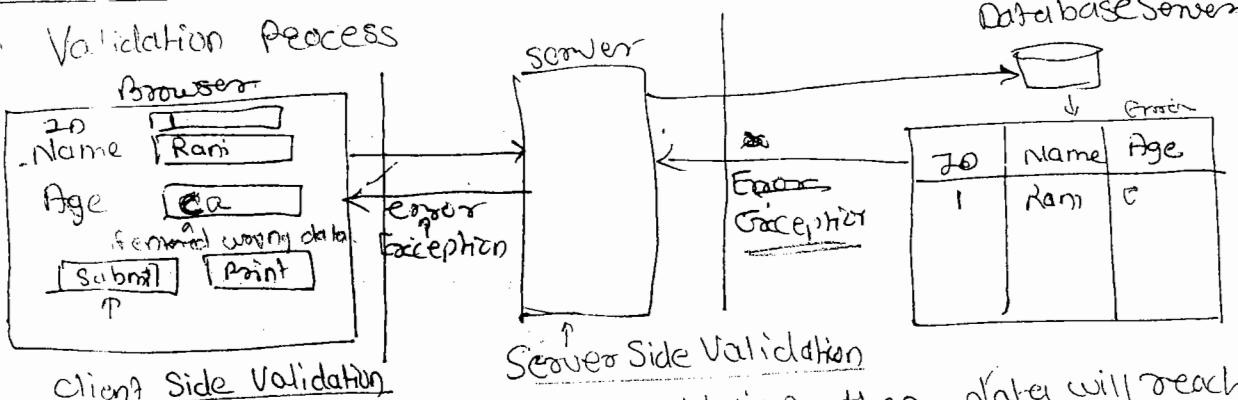
<h1> JavaScript Demo </h1>
<noscript> Please enable JavaScript on your
           browser
</noscript>

```

<body>

Note → Chrome → Settings → `Javascript` → Content Setting →
 ⚪ Do not allow → Manage exception → localhost → `block` →
 [Done] → `Done` In Internet Explorer → Internet option etc.
 How To Disable JavaScript in Browser Chrome

Validation Process



- ① If we didn't write any validation then data will reach to the Database Server & Exception is directly sent to client which is not understandable by client only programmer can understand it.
- ② So to avoid this we write Server Side Validation, i.e. Exception sent by Database is caught to Server & proper message is displayed to client.
- ③ Again In above case we are doing round trip to the Database & it will be burden on Server so to solve this

we can write ^{client} ~~server~~ side validation. which will execute on client browser only & server roundtrips can be saved.

Q) Then Should we only do all validations only by using client Side Validation? no need of server side validation?

⇒ No, We have to use combination of Server Side & Client Side validation. bcz. Client Side validation is not secured and it can be disabled on any browser & client side script can be bypassed. & also

29 JavaScript DOM (Document Object Model)

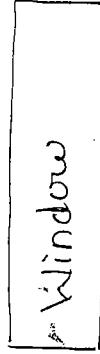
(28-1-16)

Object → should be in small letters

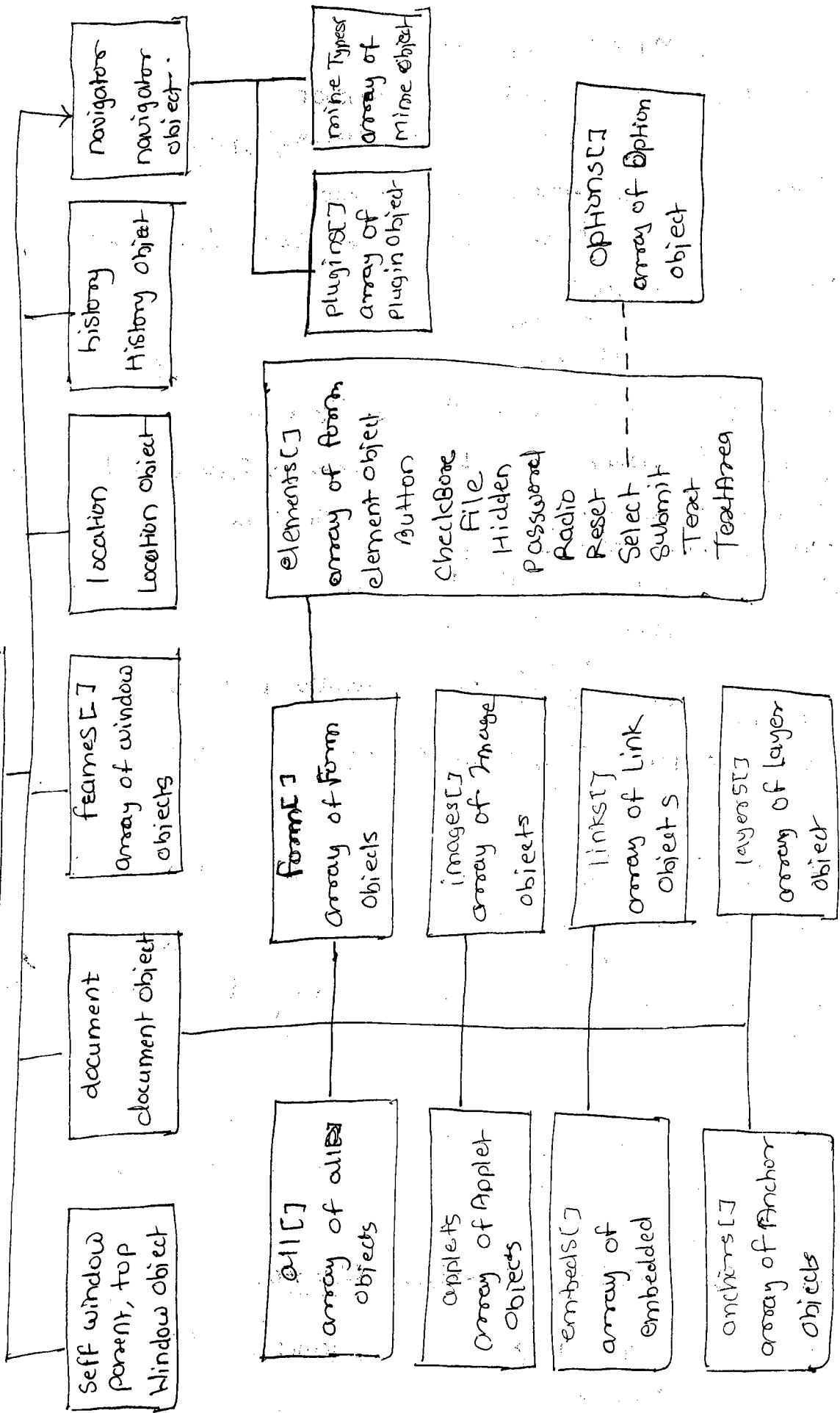
- | Object | Description |
|-------------|---|
| ① window | : Provides set of properties & methods to manipulate the browser window. |
| ② document | : It provides set of properties and methods that are used to manipulate the document elements. like forms, links, anchors, images etc... |
| ③ location | : It provides set of properties and methods that can access the client location like Port Number, protocol, Server name etc.... |
| ④ history | : It provides set of properties and methods that are used to control the browser history. i.e. the pages visited recently, total count of pages in history etc... |
| ⑤ navigator | : It provides set of properties and methods that are used to know the details of client browser like browser name, version number, plugins installed etc... |

Note → JavaScript is case sensitive. all objects should write in small letters

DOM



Document Object Model.



MIME Type →

Car.jpg → .jpg is extension

→ file type ⇒ image/jpeg

• Exec → .exe is extension

→ file type ⇒ application/octet-stream

Excel

→ .xls

.xlsx

.CSV.

→ Every file will have MIME Type.

→ Go To inetmgr → MIME Type → Extension | MIME Type

→ "text/javascript" is a MIME Type of Javascript.

Q) How to put Javascript?

⇒ <Script type="text/javascript"

language="javascript"

src = "">

</script> → if script is not locally present.

Q) Where to put Javascript Code? ⇒ in <head> generally.

⇒ ① head

clean separation (JS)

② body

③ External file → minify & use file.

→ whenever we want to run our script on specific event

then put the script in head.

→ whenever we want to run our script along with the body then put the script in body. bcz some situations

* Placing JavaScript in a Web Page

There are 2 different methods of placing JavaScript in a HTML Page.

Method 1 : Using <script> Element

Syntax:-

| | |
|-----------------|--|
| <u>Syntax:-</u> | <script type="text/javascript"> </script> |
|-----------------|--|

Syntax 2 :- <script language="javascript">
</script>

Syntax 3 :- <script type="text/javascript"
src = "ScriptFile.js">
</script>

Method - 2 :- Place Javascript within the start tag of
any element.

Syntax :-
<input type="button"
value="Print"
onclick="window.print()">

* Where to Place JavaScript In Web Page.

- ① Head Section
- ② Body Section
- ③ External File

① Script In Head Section :- Must Be define when you
want the script To Execute on any specific event.
like → Button click, Selection changed, body load etc...

Eg → <head>
<script type="text/javascript">

</script>
</head>

② Script In Body Section :- Must be define when you
want to execute during Body Load event.
→ If you want to manipulate the document elements
while loading the page then configure using script
in body section

Eg → <body>
<script type="text/javascript">

</script>
</body>

Drawback of JavaScript → Some browsers will not run JavaScript? old version
→ `<!-- -->` will script in commented lines.

- ③ Script in External File:→ allows to Reuse the code and separates the code from presentation

Ex:-

- ① Create a new Javascript files "demo.js".

- ② Link the file to your web page/^{↳ This file should be minified file.}

```
<script type="text/javascript" src="demo.min.js">  
  or (i.e. demo.min.js)  
  src = "Demo.js" >  
</script>  
  by using minification  
  Tools. eg simplify.
```

Note → How to make JavaScript available to primitive types of browsers? (ie old version browser)

⇒ Enclosed the script within Comments. /

Ex → `<script type="text/javascript">`

<!-- your script -->

- ① By DOM Hierarchy
- ② By Id Name
- ③ By Id
- ④ By TagName
- ⑤ By className
- ⑥ By class Name

30-1-16

Referring DOM elements in JavaScript

JavaScript supports several methods used to refer the elements in a document they are →

- ① Refer by using DOM Hierarchy.
- ② Refer by Using Name
- ③ `getElementsById()`
- ④ `getElementsByName()`
- ⑤ `getElementsByTagName()`
- ⑥ `getElementsByClassName()`

- ① Referring elements By Using DOM Hierarchy.

→ you have to follow the document object model (DOM) of JavaScript. in order to refer any element in HTML Page.

```

<html>
  <head>
    <script type = "text/javascript">
      function bodyLoad()
        { it is optional as it is same page
          window.document.forms[0].elements[0].value = "John";
          window.document.forms[0].elements[1].value = "Submit",
        }
      </script>
    </head>
    <body> onload = "bodyLoad()">
      <form name = "frmHome">
        <div>
          Enter Name:
          <input type = "text" name = "txtName">
        </div>
        <div>
          <input type = "button" name = "btnSubmit">
        </div>
      </form>
    </body>
  </html>

```

② Referring elements By using the reference Name

given to them → Every element in HTML document can be identified by using a reference name given with the attribute "name".

```

Ex:- <head>
  <script type = "text/javascript">
    function bodyLoad()
      { window.document.frmHome.txtName.value = "John";
        window.document.frmHome.btnSubmit.value = "Submit",
      }
    </script>
  </head>
  <body>
    <!-- Same as above -->
  </body>

```

③ Using getElementById() method →

- Every element in HTML document can be defined with a unique Id, which you can use to refer the element in code.
- In this method you need not to follow the DOM Hierarchy.

Ex:-

```
<head>
  <script> type = "text/javascript">
    function bodyLoad()
    {
      document.getElementById("txtName").value = "David";
      document.getElementById("btnSubmit").value = "Submit";
    }
  </script>
</head>
<body> <form> <div>
  <!--
  <input type = "text" Id = "txtName" >
  <input type = "button" Id = "btnSubmit" >
  </div> </form>
</body>
```

④ Using getElementsByTagName →

- A HTML document may contain several elements using the same name like all Radio Buttons in a specific category or group (ie gender/ payment) you can access all elements by using their name.

⑤ Using getElementsByTagName →

- It allows to access any element directly by referring to its Tag Name.

Ex:- All images can be accessed by using the Tag name "`img`".

⑥ Using getElementsByClassName → This method allows to access elements by using the CSS class name.

i.e. a single CSS class can be defined to several elements in a page. Hence in order to access all elements using that CSS class, you can use the `Classname()` method.

Ex → For all the above 3 accessing ways →

```
<!DOCTYPE html>
<html>
<head>
<style>
    .backstyle {
        background-color: yellow;
    }
</style>
<script type="text/javascript">
    function bodyload() {
        var p = document.getElementsByName("payment");
        var d = document.getElementsByTagName("div");
        var s = document.getElementsByClassName("backstyle");
        alert("Total No. of Payment Modes:" + p.length);
        alert("Total No. of div's:" + d.length);
        alert("Total No. of elements using backstyle :" + s.length);
    }
</script>
</head>
<body onload="bodyload()">
<form name="frmHome">
    <div>
        <h1 class="backstyle">Register</h1>
    </div>
    <div>
        Payment Modes: <br>
        <input type="radio" name="payment" value="Gift Card">
        Gift Card <br>
    </div>
</body>
```

```

<input type="radio" name="payment" value="CreditCard">
    Credit Card <br>
</div>
<div class="backstyle">
    © copyright 2016
</div>
<form>
</body>
</html>

```

1. alert() → user can't skip.
 2. document.write() → Round Trip
 3. innerHTML → use HTML
 4. innerText → RC Data
 5. console.log() → its only for developer & don't see it in browser in debugging mode.
- Console** Tab in debugger

* Handling Output In JavaScript

- ① alert()
- ② document.write() → 302 (Round Trip)
- ③ innerHTML
- ④ innerText
- ⑤ console.log()

1-1-16

① alert() → Alert is a method of window object that shows an alert dialog with specified message and seeks confirmation from the user.

Ex:- <!DOCTYPE html>

```

<html>
<head>
<script type="text/JavaScript">
function showAlert()
{
    alert ("Welcome to Javascript");
}
</script>
</head>
<body onload="showAlert()">
<h1> Alert Demo </h1>
</body>
</html>

```

② document.write →

write() is a method of document object that uses a round trip to print specified output. i.e. output is displayed in a new html document.

Ex:- <!DOCTYPE html>

```
<html>
  <head>
    <script type="text/javascript">
      function showMessage()
        {document.write("Welcome to Javascript");}
    </script>
  </head>
  <body onload="showMessage()">
    even if we put something in body it will not display.
    </body>   bcz it will display only in new document.
</html>
```

③ innerHTML →

It is a document property that can point the specific content in the document elements like h1, p, div, span etc. You can control the output by using HTML elements.

Ex:- <!DOCTYPE html>

```
<html>
  <head>
    <script type="text/javascript">
      function showResult()
        {
          var str = "Welcome to Javascript";
          document.getElementById("msg").innerHTML =
            "<marquee>" + str + "</marquee>";
        }
    </script>
  </head>
  <body> onload = "showResult()">
```

```
<h1 id="msg"></h1>  
</body>  
</html>
```

- ④ `innerText` → It is an RCDATA Type property that treats the content as `RawText`. hence it will not allow to use any HTML elements to modify the output.

Ex:- Same as `innerHTML` → Remaining Complete Exp.

```
var str = "<b> Welcome </b>";
```

```
document.getElementById("msg").innerText = str;
```

O/p → Welcome

- ⑤ `Console.log()` → It is a function that prints specified Content into the console of web debugger like fiddler, firebug, Chrome Debugger (F12)

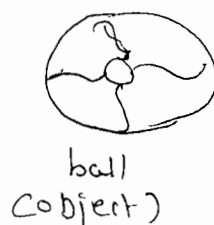
[Firefox →  → Addons → ] →

Install →  → all checkboxes. → Install] in Helpsee options

Ex:- var str = "Welcome to JavaScript";
console.log(str);

Note → To view the console →

- ① open your page in chrome
- ② press "F12" function key
- ③ Goto "Console" Category



ball
object

Red [→ Properties of ball
10mm]

kick → fly() } methods of ball object
(pin → Explode) }
events of ball object

Class is blue point to create an objects.

JavaScript BOM (Browser Object Model) Objects

- ① window
- ② navigator
- ③ location
- ④ history
- ⑤ document

① window Object →

The JavaScript window object provides set of properties and methods to control browser window

[1] Methods →

| Method | Description |
|------------------------|---|
| open() | : opens a new window with specified URL & properties. i.e. parameters |
| close() | : closes the current window (it will not work in chrome) |
| print() | : Invokes printer on your computer and prints the current page contents. |
| blur() | : blurs the current window. i.e. it is out of focus. |
| focus() | : gets focus to specified window so that it is displayed on the top of all other windows. |
| alert() | : displays an alert box with specified message. |
| confirm() | : displays a confirm box that allows the user to accept or cancel the actions. |
| prompt() | : Displays an input control that allows the user to input a value during runtime |
| resizeTo() moveTo() | feature: mouse |

Ex: A simple HTML document using open, close and print methods.

Note → Close method is supported only on browser like IE, firefox.

```
window object html  
<!DOCTYPE html>  
<html>  
<head>  
<script type = "text/javascript">  
function openWindow ()  
{  
    window.open("http://www.google.com", "Google",  
    "width=400, height=400");  
    window.open("http://www.yahoo.com", "Yahoo",  
    "width=400, height=400");  
}  
function closeWindow() // Semicolon is not mandatory  
{ window.close(); }  
// in Javascript but it is  
// recommended.  
function printPage()  
{  
    window.print();  
}  
</script>  
</head>  
<body>  
<form name = "formHome" >  
    <input type = "button" name = "btnOpen" value = "  
        Open Window" onclick = "openWindow()" >  
    <input type = "button" name = "btnClose" value = "Close  
        Window" onclick = "closeWindow()" >  
    <input type = "button" name = "btnPrint" value = "Print  
        Page" onclick = "printPage()" >  
</form>  
</body>  
</html>
```

Note → Syntax for window.open() →
window.open("url", "title", "params / properties").
Ex → window.open("http://www.google.com", "Google",
"height=400, width=400");

21/2/16

alert() → It shows a message box with **OK** button. So that user can continue with his actions by clicking **OK**.

Syntax:-

```
alert("message");
```

confirm() → It shows an alert with **OK** & **Cancel** buttons. User can continue by clicking **OK** or can cancel the actions by clicking **Cancel**.
→ It returns "Boolean true" on **OK** click and "Boolean false" on **Cancel** click.

Syntax:-

```
confirm("message");
```

prompt() → It is an input Control that allows the user to input values dynamically during runtime.
It Shows a prompt dialog with **OK** & **Cancel** buttons. and it Returns "Null" when clicked **Cancel** & returns empty string if input is not provided & **OK** clicked.

Syntax:-

```
prompt("message", "value");
```

Ex → **Dialogs.html** Ex. for alert(), prompt(), confirm()

```
<!DOCTYPE html>
<html>
<head>
```

```
<script type="text/javascript">
    function showAlert() {
        alert("welcome to Javascript");
    }

    function showConfirm() {
        var xc = confirm("Are you sure you want to delete?");
        if (xc == true) {
            document.getElementById("msg").innerHTML =
                "You clicked OK";
        } else {
            document.getElementById("msg").innerHTML =
                "You clicked Cancel";
        }
    }

    function showPrompt() {
        var name = prompt("Enter Name", "Name min 4 chars");
        document.getElementById("msg").innerHTML =
            "Hello!" + name;

        if (name == null) → ie user clicked [cancel] button
            document.getElementById("msg").innerHTML =
                "Please Enter Name";
    }
</script>
</head>
<body>
    <form name="fromHome">
        <div id="msg"> </div>
    </form>
</body>
```

```

<input type="button" value="ShowAlert" name="btnAlert"
       onclick="ShowAlertc()">
<input type="button" value="Confirm" name="btnConfirm"
       onclick="showConfirmc()">
<input type="button" value="EnterName" name="btnPrompt"
       onclick="showPromptc()">
</form>    moveTo(), resizeTo().
</body>
</html>

```

~~Ex → for moveTo() & resizeTo()~~

Note → *Resizing the window & Specifying window position on X & Y axis.:*

- The method "moveTo()" is used to change window position on x & y axis.
- The method "resizeTo()" is used to resize the window by specified height & width.

Syntax:

```

var win = window.open("url", "title", "width=100,
                      height=100");
win.moveTo(100, 100);
win.resizeTo(400, 400);

```

* JavaScript window Object Properties

| Property | Description |
|-----------------------------------|---|
| ① closed | : Returns boolean true if window is in closed state |
| ② locationbar (ie address bar) | : Specifies whether to display the locationbar in browser window. (Can be set to true/false) |
| ③ menubar | : Show/hide menu bar. (Can be set to true/false) |

④ Toolbar

: show/hide the toolbar
(true/false)

⑤ Scrollbars

: Show/hide the scrollbars
(true/false)

⑥ Statusbars

(at the bottom
left corner)

: refers to the status bar of
browser window.

⑦ Status

: Shows information in status bar.

Syntax:-

```
window.open("http://www.nareshit.in","_blank",  
"toolbar=yes,location=yes,directories=no,  
status=no,menubar=yes,scrollbars=yes,  
resizable=no,copyhistory=yes,width=400,  
height=400");
```

Note → The boolean true/false are referred using
"yes/no"

* Interview Questions on this
for Server Side Navigator Object (Imp) for Developers
to know about client browser.
programmers also.

Q) How we can display or hide div using JS?

⇒ <div style="display:none">

Q) How to disable Cookies?

⇒ Set Browser → Settings → Cookies → Content settings →

block 3rd party → management tab →

Localhost block → done.

JavaScript Navigator Object

→ The navigator object provides set of properties
that are responsible for returning the details
of client browser like browser name, family,

Cookie enabled, Plugins, mime types, language, System language etc.

| Property | Description |
|---------------------|--|
| ① appCodeName | : It returns the browser name. |
| ② appName | : It returns the browser family name like netscape |
| ③ appVersion | : It returns the browser Version |
| ④ userAgent | : It returns the user agent header that describes the compatibility of your page on various platforms. Like is it suitable for Safari / Android / mozilla / chrome |
| ⑤ mimeTypes[0].type | : It is an array of mime Types Supported on your machine you can access individual mime type by referring to their index. |
| ⑥ plugins[0].name | : It returns an array of plugins supported on your browser. |
| ⑦ language | : It returns the browser language. |
| ⑧ platform | : It returns the client platform like win32, win64, etc. |
| ⑨ online / offline | : It returns boolean true if browser is online, you can set your browser to offline to access offline pages. (Opera & FF supports offline) |
| ⑩ systemLanguage | : It returns the client culture information |

(11) cookieEnabled : It returns the status of cookies that is enabled or not.

* Cookies are small text files which stored on client machine memory.

Ex:- <!DOCTYPE html>

```
<html>
<head>
<script type="text/javascript">
function ShowDetails()
{
document.getElementById("tbl").style.display="block";
document.getElementById("browserFamily").innerHTML=
navigator.appName;
document.getElementById("platform").innerHTML=
navigator.platform;
document.getElementById("cookies").innerHTML =
navigator.cookieEnabled;
document.getElementById("lang").innerHTML =
navigator.language;
}
</script>
</head>
<body>
<div>
<form name="frmHome">
<input type="button" value="Show Client Browser Details"
onclick="ShowDetails()">
</div>
```

```

<div id="tbl" style="display:none">
  <table width="400" border="1" align="center">
    <tr>
      <td> Browser Family </td>
      <td> <span id="browserfamily"></span> </td>
    </tr>
    <tr>
      <td>client Platform </td>
      <td> <span id="platform"></span> </td>
    </tr>
    <tr>
      <td> Cookie Enabled </td>
      <td> <span id="cookies"></span> </td>
    </tr>
    <tr>
      <td> Browser Language </td>
      <td> <span id="lang"></span> </td>
    </tr>
  </table>
</div>
</form>
</body>
</html>

```

3-2-16

JavaScript Location Object

→ The location object provides set of properties and methods that are used to access the client location like servername, domainname, portno, protocol, host etc.

| Property | Description |
|----------|--|
| ① hash | : Returns the id using hash. that refers the recently viewed location. |
| ② host | : gets the Server name or IP Address of client. |

- ③ hostname : It is similar to a host which returns server name and domain name along with port no.
- ④ href : returns the complete URL of client.
- ⑤ pathname : returns the associated physical folder name of current page.
- ⑥ port : returns the port number if any specified.
- ⑦ protocol : returns the client protocol ie http, https, file, ftp etc.
- ⑧ search : returns the search string, ie that is passed as query string in the URL.

Method

- ① reload() : Reloads the current page.
- ② replace() : Replaces the current URL with a new URL.

Ex → ① ⇒ Get the client location Details.

location.html

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function ShowDetails()
{
    document.getElementById("tbl").style.display = "block";
    document.getElementById("url").innerHTML = location.href;
    document.getElementById("port").innerHTML = location.port;
    document.getElementById("protocol").innerHTML = location.protocol;
    document.getElementById("host").innerHTML = location.host;
}

```

```
document.getElementById("path").innerHTML = location.pathname;
}

</script>
</head>
<body>
<div>
<form name="formHome">
<input type="button" value="Show Client Details".
onclick="ShowDetail()">

<br>
<div id="tbl" style="display:none">
<table border="1" width="400">
<tr>
<td> Your URL </td>
<td> <span id="url"></span> </td>
</tr>
<tr>
<td> Your Port No. </td>
<td> <span id="port"></span> </td>
</tr>
<tr>
<td> Protocol </td>
<td> <span id="protocol"></span> </td>
</tr>
<tr>
<td> Your Host </td>
<td> <span id="host"></span> </td>
</tr>
<tr>
<td> Page Path </td>
<td> <span id="path"></span> </td>
</tr>
</table>
</div>
</form>
</body>
</html>
```

→ In debugger, to step wise execution by putting break point → click ⏪ or F10. To check next step.

→ Sources Debugger → sources → click on webpage → (ie location1.html)

Ex ② → Get Recently viewed location of client on current page. and control Redirection from 1 page to another page.

location.html

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
    function GetLocation()
    {
        var str = location.hash;
        switch(str)
        {
            case "#javascript":
                alert("You viewed Javascript Basics - view more tutorials");
                location.href = "http://localhost/amazon/tutorial.html";
                break;
            case "#mobile":
                alert("You saw mobile - You will be redirected to product page");
                location.href = "http://localhost/American/Shopping.html";
                break;
        }
    }
</script>
</head>
<body>
```

```

<div>
  <h2 align="center">Table of Contents </h2>
  <ol>
    <li><a href="#javascript"> JavaScript Basics </a>
    <li><a href="#mobile"> LG Mobile </a> </li>
  </ol>
<div>
  <form name="formHome">
    <input type="button" value="Recently Viewed"
      onclick="GetLocation()" >
  </form>
<div>
  <h2 id="javascript"> JavaScript Basics </h2>
  <p> --- your paragraph --- </p>
  <figure id="mobile">
    
    <figcaption> LG Mobile </figcaption>
  </figure>
  <p> --- your paragraph --- </p>
  </div>
</body>
</html>

```

[4-2-16]

Location methods Example.

```

<form name="formHome">
  <input type="button" value="Reload" name="btnReload"
    onclick="location.reload()" >
  <input type="button" value="Google!" name="btnRedirect"
    onclick="location.replace('http://www.google.com')" >
</form>

```

JavaScript History Object

① length

② back()

③ forward()

④ go() → go(-7), go(+2) etc. → going on specific page

Q) How can we disable the "back" button on browser?

→ window.history.forward() → This is not perfect option,
will work on few browsers in older days.
Nowadays we do that using "push event".

→ The history object provides a set of properties and methods that are used to manipulate the browser history.

Members

Descriptions

① length

→ Returns total count of pages in a browser history:

② back()

→ Navigates to the previous page in history.

③ forward()

→ Navigates to the next page in browser history if any available.

④ go()

→ Navigates to any specific page in browser history.

- ve ⇒ Back

+ ve ⇒ forward.

ie go(-3) or go(3)

→ History length property.

<!DOCTYPE html>

<html>

<head>

<script type = "text/javascript">

fn

```

function bodyload()
{
    var h = history.length;
    if (h > 3)
    {
        location.href = "http://localhost/amazon/register.html";
    }
    else
    {
        document.getElementById("msg").innerHTML = "Hai! you
        can view more 3 pages for free --";
    }
}
</script>
</head>
<body onload="bodyload()">
<h1> JavaScript History Object </h1>
<div id="msg">
</div>
</body>
</html>

```

Ex 2 → History Back(), forward() & go() method. Example

```

<form name="formHome"> history.html
    <input type="button" name="btnback" value="Back"
           onclick="history.back()"/>
    <input type="button" name="btngo" value="Go To
    page 3" onclick="history.go(-3)"/>
    <input type="button" value="Clear"/>
</form>

```

* Define/Place "forward" button in register.html. After clicking forward button in register.html, we should be redirected to history.html.

JavaScript Screen Object

- (Q) what will be the default size of a web page?
→ 800x600. (Now 1024x768) (80rows & 40columns)
↳ screen can have
favorite icon → 16x16
- (Q) How we can design webpage so that it can fit in different devices like mobile, Tablets etc?
⇒ using CSS.
- * "white paper" will contain all the info. about the software which people is going to release..

Cmd → C:\User\admin> mode

Columns → 80

we can change it →

Cmd → C:\User\admin> mode 40

Columns → 40

→ The screen object provides set of properties & methods that are used to get the resolution of client screen, as well as the bit type.

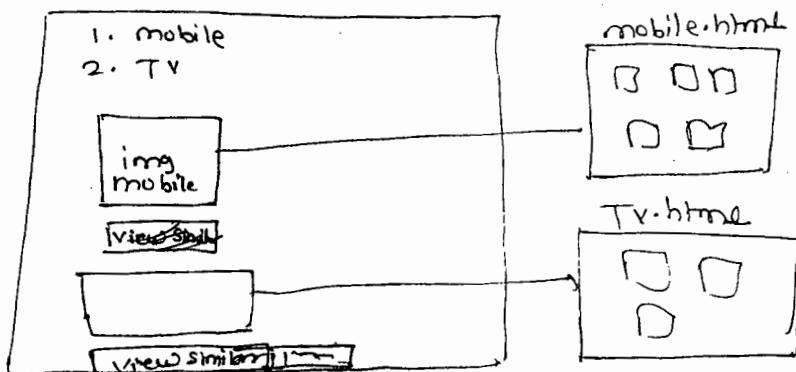
| Property | Description |
|---------------|---|
| ① width | → gets the screen width. Screen.width |
| ② height | → gets the screen height. Screen.height |
| ③ availWidth | → Gets the more screen width supported. |
| ④ availHeight | → Gets the more screen height supported. |
| ⑤ colorDepth | → Returns colors per pixel. |
| ⑥ pixelDepth | → Similar to colorDepth but not supported by many browsers. Hence use colorDepth instead. |

Ex: for Screen Object

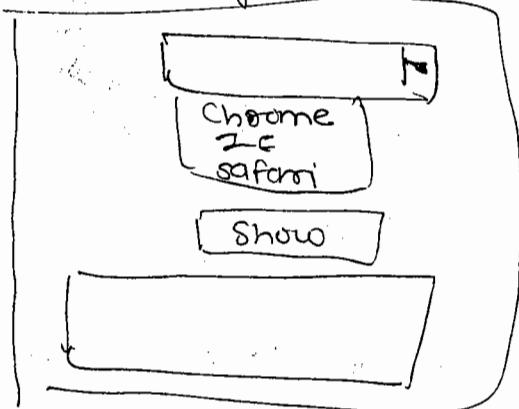
```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function bodyload()
{
    var w = screen.width;
    var h = screen.height;
    document.getElementById("msg").innerHTML = "Your Screen
Resolution is "+w+"x"+h+" pixels "+ "<br>" +
"ColorDepth "+ screen.pixelDepth ;
    // or screen.colorDepth .
}
</script>
</head>
<body onload="bodyload()">
<div id="msg"></div>
</body>
</html>
```

Ex:- ① Create a page called "Payment.html" & display
the message based on protocol used.
ie if user is using http
then show -- not secured.
else
show -- secured → you are Secured.

② Create a Shopping Cart page with product thumbnails
and description.



③ Create a dropdownlist that contains a list of browsers and shows if your browser is any one of the following, or in a given list. use switch case.



```
<div>
<form>
    Select Browser:
    <select id="lstBrowsers">
        <option value="Internet Explorer">
            Internet Explorer </option>
        <option value="Chrome">
            Chrome </option>
    </select>
```

```
<input type="button"
```

```
    onclick="getBrowser()"
```

```
function getBrowser()
```

```
    var b = document.getElementById("lstBrowsers").value;  
    yourdocument.write("Your selected browser is: " + b);
```

```
5
```

Variables in JavaScript

512116

Variables → Starts with underscore (-) or a-z.

① Variables starts with underscore (-) or a-z.
124A → invalid.

② Can contain numbers.

✓ Sales2008

✓ -Sale 2016

③ Special Characters. → = \$ -

④ Case Sensitive

X = 10 name = "john"

xe = 20

⑤ Size → Not specific ⇒ but 4mb.

⑥ Implicitly typed.

var i = 10; → i is int

var i = "A"; → i is string.

⑦ Not mandatory to declare the variable and use directly.
ie even if we don't write var keyword then also
var value3 = 30

⑧ Variable can be ① Local or ② Global. → window.value1 = 10
value3 = 30.

→ Variables are simply storage locations in memory where you can store a value & use it as a part of any expression.

→ The JavaScript variables are declared by using the keyword "Var", which represents implicitly typed variable, i.e. the datatype of variable is determined according to the value assigned.

→ Variable naming Conventions ⇒

① A variable name must start with an alphabet or underscore (-) symbol.

② It can be alphanumeric with special characters like \$ - and \$.

③ Variable names are case sensitive.

④ There is no standard limit of characters in variable name. however the default size on JS is 4MB.

and it is adjustable.

⑤ The variable scope can be 2 types →

- ⑥ Local Variable
- ⑦ Global Variable

Syntax:-

① `var value1 = 100; // Global variable`
function a()

{
 ② `var value2 = 200; // local variable`
 }

Note ③ You can directly use a variable without declaring it.
Eg → `var sales = 1000; // valid.` ✓

OR
`sales = 1000; // valid` ✓

④ You can declare a global variable in a function by using the keyword window.

Eg:- `function a()`

⑤ `window.value1 = 1000;`

~~also~~ `alert(window.value1);`

/* Global variable. but
will not support by
all browsers. Chrome
will support it.
so not recommended.

To access this variable, we
have to access using window
keyword

⑥ You can also declare a global variable just by specifying the name and value without "var" or window keywords.

3 ways to declare
Global variable.

Eg:- `function a()`

⑦ `var value1 = 10; // local`

⑧ `value2 = 20; // global;`

↑
by default without var keyword variable is
considered as windows member scope.

Data Types

① number -

② ⁴¹ Strings

✓ var i = 10;

✓ var i = 5.6;

✓ var i = "Sales";

✗ var i = sales45;

✓ var i = 2e3

$$2 \times 10^3 = 2000$$

✓ var i = 2E3

$$2 \times 10^3 = 2000.0$$

a = eval(2e3)

In JavaScript

② Strings

var n = "

✓ var n =

true/false

✓ var n =

null

✓ var n = null;

③ Boolean

var n =

true/false

✓ var n =

null

Note → ① Confirm ("R you sure?")

returns true ← [OK]

Boolean false ← [Cancel]

② Prompt ("Enter name");

returns [OK] ⇒ Entered value returns

string [Cancel] ⇒ returns Empty if no input

null ⇒ returns "null"

- The values & variables in JavaScript are implicitly typed hence the datatype is determined based on the value assigned or specified.
- JavaScript datatypes can be classified into 2 types

① Primitive Datatypes

② Non-Primitive / Reference Datatypes

→ Object

→ Array

→ Regular Expression. ie RegExp

[1] Primitive Types →

The Primitive types are

| Datatype | Description |
|-------------|--|
| ① String | → represents sequence of characters. eg "hello" |
| ② Number | → represents numeric values eg. 10 ³ |
| ③ Boolean | → represents boolean value either true or false |
| ④ Undefined | → represents undefined value |
| ⑤ Null | → represents null ie. no value at all |

Valid String \Rightarrow

Var name = "john";

Var path = "C:\windows";

Var year = "2018";

Valid Numbers \Rightarrow

Var x = 10;

Var x = 10.3;

Var x = -2E3 $\Rightarrow 2 \times 10^3 = 2000$ • 002

Var x = -2e3 $\Rightarrow 2 \times 10^{-3} = 0.002$.

Var x = 20x \Rightarrow must be parsed.

Cie Var x = parseInt("20x");

Valid Boolean \Rightarrow

Var x = true; } w3-group

Var x = false;

Var x = yes; } WHAT Group allowed valid

Var x = no;

Valid Null Type \Rightarrow

Var x = null;

Var x = null;

Note \rightarrow The dialog Results in Javascript always return either Boolean or null values.

Ex: ① Confirm()

\Rightarrow Returns true, when **OK** clicked,

\Rightarrow Returns false, when **cancel** clicked.

② Prompt()

\Rightarrow Returns not null when **OK** clicked,

\Rightarrow Returns null, when **cancel** clicked.

[2] Non- Primitive Types in JavaScript \Rightarrow

DataType Description

- ① Object \rightarrow represents instance through which we can access members.
- ② Array \rightarrow represents group of similar values.
- ③ RegExp \rightarrow represents Regular Expression.

JavaScript Operators

- \rightarrow Results are computed by using expressions.
- \rightarrow Expressions are built using operators.

Expressions are of following Type \rightarrow

- ① String Expression \Rightarrow "A" + "B" = AB
- ② Boolean Expression \Rightarrow 10 > 20 = true/false
- ③ Arithmetic " \Rightarrow 10 - 5 = 5
- ④ null expression \Rightarrow $\infty \equiv \text{null}$

- \rightarrow The results are computed in any programming language by building Expressions. And Expressions are built by combining variables and operators.
- \rightarrow one built by combining variables and operators together into a unit.

\rightarrow The JavaScript Expressions are classified into 4 types \rightarrow

- ① String Expressions :- Returns a string.
- ② Boolean Expressions :- Returns a boolean value.
- ③ Arithmetic Expressions:- Returns a number.
- ④ null Expressions :- Returns null.

\rightarrow The operators in JavaScript are classified into the following types.

- ① Arithmetic Operators
- ② Comparison (Relational) Operators
- ③ Bitwise Operators.

④ Logical Operators

⑤ Assignment Operators

⑥ Special Operators

⑦ Arithmetic Operators →

Operator Description

+

Addition

-

Subtraction

*

Multiplication

/

Division

%

Modulus (Reminder)

++

Increment

--

Decrement

Example

$$10 + 20 = 30$$

$$20 - 10 = 10$$

$$10 * 20 = 200$$

$$\rightarrow 20 / 10 = 2$$

$$20 \% 10 = 0$$

→ var a = 10; a++

Now a = 11

→ var a = 10; a--;

Now a = 9

② Comparison (Relational) operators

Operator Description

==

Is Equal to

Example

$$10 == 20 = \text{false}$$

==

Identical (Equal
and of same type)

$$10 == 20 = \text{false}$$

!=

Not equal to

$$10 != 20 = \text{true}$$

!=

not Identical

$$20 != 20 = \text{false}$$

>

Greater than

$$20 > 10 = \text{true}$$

>=

Greater than or
equal to

$$20 \geq 10 = \text{true}$$

<

Less than

$$20 < 10 = \text{false}$$

<=

Less than or equal
to

$$20 \leq 10 = \text{false}$$

⑧ Bitwise Operators

Operator Description

&

Bitwise AND

Example

$$(10 == 20 \& 20 == 30) = \text{false}$$

|

Bitwise OR

$$(10 == 20 | 20 == 30) = \text{false}$$

| | | |
|----------|----------------------------------|---|
| \wedge | Bitwise XOR | $(10 == 20 \wedge 20 == 33) = \text{false}$ |
| \sim | Bitwise NOT | $(\sim 10) = -10$ |
| $<<$ | Bitwise Left Shift | $(10 << 2) = 40$ |
| $>>$ | Bitwise Right Shift with zero | $(10 >> 2) = 2$ |
| $>>>$ | Bitwise Right Shift | $(10 >>> 2) = 2$ |

④ Logical Operators →

| <u>Operator</u> | <u>Description</u> | <u>Example</u> |
|-----------------|--------------------|--|
| $\&\&$ | Logical AND | $(10 == 20 \&\& 20 == 33) = \text{false}$ |
| $\ \ $ | Logical OR | $(10 == 20 \ \ 20 == 33) = \text{false}$ |
| ! | Logical NOT | $!(10 == 20) = \text{true.}$ |

⑤ Assignment Operators →

| <u>Operator</u> | <u>Description</u> | <u>Example</u> |
|-----------------|---------------------|---|
| $=$ | Assign | $10 + 10 = 20$ |
| $+=$ | Add and assign | $\text{var } a = 10; a += 20 ;$ Now $a = 30$ |
| $-=$ | Subtract and assign | $\text{var } a = 20; a -= 10 ;$ Now $a = 10$ |
| $*=$ | Multiply and assign | $\text{var } a = 10; a *= 20 ;$ Now $a = 200.$ |
| $/=$ | Divide and assign | $\text{var } a = 10 ; a /= 2$ Now $a = 5$ |
| $%=$ | Modulus and assign | $\text{var } a = 10; a \% = 2 ;$ Now $a = 0.$ |

⑥ Special Operators →

| <u>Operator</u> | <u>Description</u> |
|-----------------|---|
| $(? :)$ | Ternary operator : Similar to If...Else |
| , | Comma operator : allows multiple expressions to be evaluated as a single statement. |
| delete | delete operator deletes a property from the object |

in → In Operator checks if object has the given property.

instanceof → checks if the object is an instance of given type

new → Dynamic memory allocator operator creates an instance (object)

typeof → Checks the type of object.

void → it discards the expression's return value.

yield → Checks what is returned in a generator by the generators iterator.

Note → Type Casting → It is the process of converting 1 datatype to another. The values that you input through various controls in HTML are by default "string" type. In order to convert into integer or floating point types you have to use the following functions.

① parseInt()

② parseFloat()

③ isNaN() (ie not a number)

Ex:-

```
var a = "10";
var b = "20";
c = a+b;           // result: c = 2020
```

```
var a = parseInt("10");
var b = parseInt("20"); "Hello!20" → 0 in ⇒ not a num
c = a+b;           // result: c = 30
```

a = "10"

b = "

H.W.

| Description | Your Value |
|---|------------|
| Loan Amount | 5,000,00 |
| Interest | 5% |
| Months | 24 |
| Monthly Payment | |
| P.M.T = | |
| <input type="button" value="calculate"/> <input type="button" value="Reset"/> | |

green
blue
yellow

① EMI
② BMI
③ Currency Converter.

Statements In JavaScript

① Selection:

if, else, switch, case

② Iteration

for, foreach, while, do while

③ Jump

break, continue, goto, default, return

④ Exception Handling

try, catch, finally, throw

⑤ Checked & Not supported by JavaScript but Some

⑥ Faced J prog. lang. support this.

9-2-16

→ The results in programming are computed by building Expressions and a combination of expressions will form into statements that execute in a sequential order or based on any specific condition. The statements in programming are classified into following types →

Statements

Keyword

① Selection Statement

if, else, switch, case

② Iteration Statement

for, foreach, do while, while

③ Jump Statement

break, continue, goto, default, return

④ Exception Handling Stmt

try, catch, finally, throw

① Selection Statements →

① If Select? It executes a set of statements based on given condition it is used to select statements for execution with regard to the given criteria.

Syntax:-

① if (condition)

 Statement 1;
 Statement 2;

→ Only Statement 2 executes as conditional statement.

② if (condition)

 {
 Statements if true;
 }

③ if (condition)

 {
 Statements if true;
 }

 }
 else

 {
 Statements if false;
 }

}

④ if (Condition1)

 {
 Statements if Condition1 true;
 }

 {
 else if (Condition2)

 {
 Statements if condition2 true;
 }

}

 else

 {
 Statements if all conditions evaluate to false;
 }

}

⑤ if (Condition1),

 {
 if (Condition2)

 {
 if (Condition3)

 {
 Statements to execute;
 }

 }

 }

Ex:- KFC ORDER FORM

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
Function BillSummary()
{
var name=document.getElementById("txtName").value;
var burger = document.getElementById("optBurger");
var roller = document.getElementById("optRoller");
if (burger.checked == true)
{
    mcost = 100;
}
if (roller.checked == true)
{
    mcost = 120;
}
var fries = document.getElementById("optFries");
var krusher = document.getElementById("optKrusher");
if (fries.checked == true)
{
    acost = 60;
    mcost = mcost + acost;
}
if (krusher.checked == true)
{
    acost = 80;
    mcost = mcost + acost;
}
document.getElementById("summary").innerHTML = "Hello," + name + "<br>" + "Your Bill Amount:" + mcost;
}
</script>
```

```
</head>
<body>
<font face="arial" size="6">
<table>
<tr>
<td colspan="2">

</td>
</tr>
<tr>
<td>
Enter name:
</td>
<td>
<input type="text" id="taetname">
</td>
</tr>
<tr>
<td colspan="2" align="center" bgcolor="yellow">
Select a Meal <td></td>
<td align="center">
<td>

<br>
<input type="radio" name="meal" value="optBurger"
id="optBurger"> OMG Burger (200/-)
</td>
<td>

<br>
<input type="radio" name="meal" value="optRoller"
id="optRoller"> OMG Roller (120/-)
</td>
</td>
</tr>
```

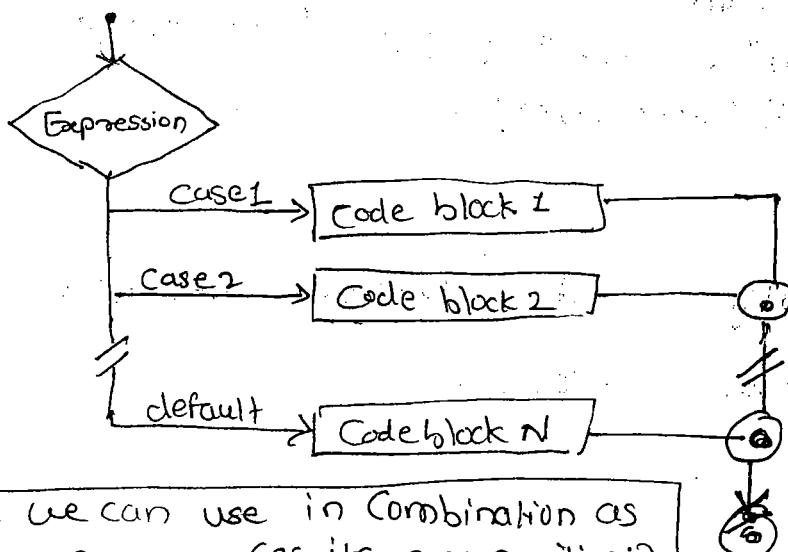
```
<td colspan="2" align="center" bgcolor="yellow">
    Select Add-On </td> </td>
<tr align="center">
    <td>
        
        <br>
        <input type="checkbox" name="OptFoies" value="optFoies" id="optFoies"> Large Foies (60/-) <del></td>
    </td>
    <td>
        
        <br>
        <input type="checkbox" name="OptKrausher" value="optKrausher" id="optKrausher"> Brownie Krausher (80/-)
    </td>
<tr>
    <td colspan="2" align="center">
        <input type="button" value="placeOrder" id="btnOrder" onclick="BillSummary()">
    </td>
<tr>
    <td colspan="2" align="center" bgcolor="yellow">
        <div id="summary" style="border: 1px solid black; padding: 5px; width: fit-content; margin: auto; font-size: small; background-color: white; border-radius: 10px; height: fit-content; min-height: 100px; max-height: 150px; overflow-y: scroll; position: relative; z-index: 1; ">
            <div style="position: absolute; top: -10px; left: -10px; width: 20px; height: 20px; background-color: black; border-radius: 50%; color: white; font-size: 1.5em; text-align: center; line-height: 20px; z-index: 2; ">X</div>
        </div>
    </td>
</tr>
</table>
</font>
</body>
</html>
```

✓ Name:
 Check-in: date $4000/-$
 RoomType
 Deluxe Suite
 $2500/-$ $6000/-$
 Emissaries
 Lockers A/C
 $1000/-$ $1500/-$
 Advance: required
 Show Summary

O/P →
 Name:
 Checkin:
 RoomType:
 Emissary:
 Advance:
 Balance:
 10-2-16

Selection Statement

Switch Statement:- It is a selection statement that enables to execute a block of code based on the specified condition. It will reduce the complexity while handling multiple conditions.
 Syntax (Representation)



* We can use it in combination as follows → (as its case sensitive)

Case "A": Note
Case "a":

Syntax →
 Switch (literal / Expression)
 {
 Case 1:
 statements;
 break;
 Case 2:
 statements;
 break;
 default:
 statements;
 break;
 }

Syntax ② →
switch (string)

```
{  
    case "String1":  
        statements to execute;  
        break;  
    case "String2":  
        statements to execute;  
        break;  
}  
3
```

Iteration Statements

- The iteration statements cause embedded statements to be executed a number of times based on the given condition.
- The keywords are ① for ② foreach ③ while ④ do while.
 - ① The 'for' Loop → It executes a statements or a block of statements repeatedly until the specified expression evaluates to false.

Syntax:-

```
for (init; condition; increment)  
{  
    conditional code;  
}
```

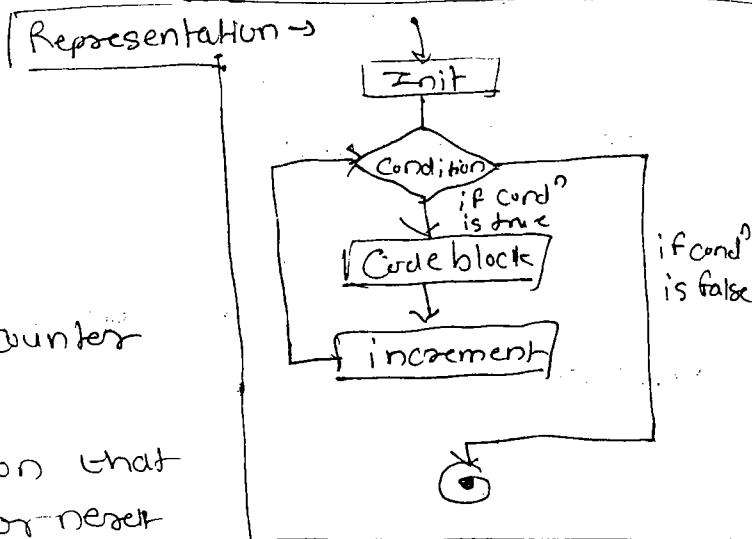
Initializer → Initialize a counter variable to start with.

Condition → Specify a condition that must evaluate to true for next iteration.

Iteration → increase or decrease counter.

Ex → for (var i=0; i<10; i++)
{
 document.alert(i);
}

Ex:- for (var i=0; i<=10; i+=2) // even no.
{
 alert(i);
}



Interview Qn

- Print Alphabet A-Z; in ascending & descending order.
- Print numbers 1-10 descending order.
- Print Even Number between 1 to 10.
- Print factorial No. of a no.
- Print a pyramid of Stars, " * ".

② The "while" Loop → It is the iteration statement

that executes a block of statements until the specified condition evaluates to false. The statements will execute only when the condition is true.

Syntax:-

```
while (Condition)
```

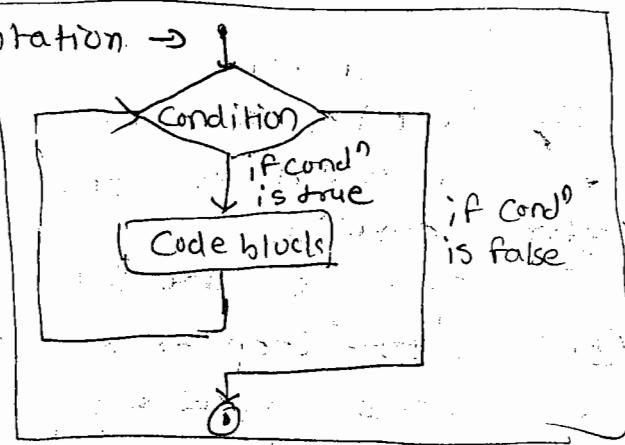
```
{
```

```
Conditional Code;
```

```
}
```

```
Ex:- var i=0;
      while (i < 10)
      {
          alert(i);
          i++;
      }
```

Representation →



③ The "do...while" Loop ⇒ It is similar to the while loop but guarantees that statements will execute at least once even the condition is false.

Syntax→

```
do
```

```
{
```

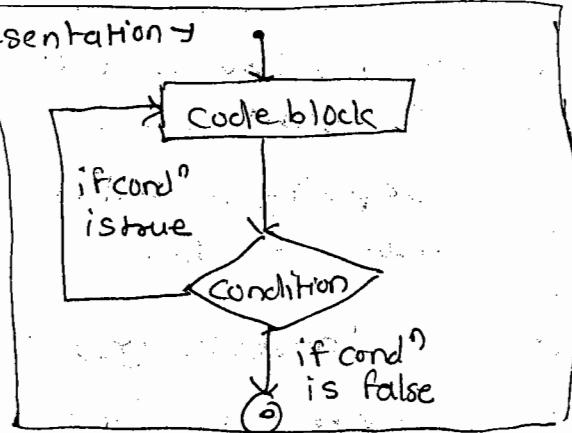
```
Conditional Code;
```

```
}
```

```
{while (Condition)}
```

```
Ex:- var i=0;
      do
      {
          alert(i);
          i++;
      } while(i<10);
      white
```

Representation →



Q) Why array index starts with 0? | RAM & H.D difference?

Array Object in JavaScript

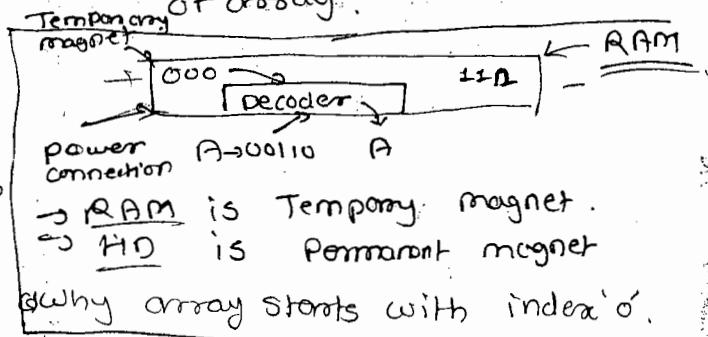
- ① Reduce Overhead (ie stored values in sequential order)
- ② Reduce Complexity (ie stored multiple values under 1 name)

Ex:- var cities = ["delhi", "mumbai"],
var sales = [5400, 56000, 7600]; } way ① to initialize
var mixed = ["delhi", 5400]; an array.

Array Constructors →

- way ② var cities = new Array ("delhi", "mum");
var cities = new Array (3); // Restricting or initializing size of array.

cities[0] = "vastak", "hyd";
cities[1] = "mumbai";
cities[2] = "pune".;



* Array Methods → Properties

- ① Reverse()
- ② Pop()
- ③ Push()
- ④ IndexOf()
- ⑤ LastIndexOf()
- ⑥ ToString()
- ⑦ split()

① length

Why for loop is used for arrays?

→ Arrays in programming are used to

- ① Reduce Overhead
 - ② Reduce Complexity
- Arrays will reduce overhead by storing multiple values in a sequential order.
- They will reduce complexity by storing all values under single name.

Declaring Array

- ① var stringArray = ["One", "two", "three"];
- ② var numericArray = [1, 2, 3, 4],
- ③ var decimalArray = [1.1, 2.2, 3.3],
- ④ var booleanArray = [true, false, false, true],
- ⑤ var mixedArray = [1, "two", "three", 4],

Array Constructor →

- The array constructor initializes size of an array & restricts the no. of values that can be stored in an array.
- It will allocate memory by using "new" operator.

Syntax →

① `Var stringArray = new Array(); // any no. of values can be stored.`

`StringArray[0] = "One";`

`StringArray[1] = "two";`

`StringArray[2] = "three";`

`StringArray[3] = "four";`

(mostly technologies supports upto 2GB)

② `Var numericArray = new Array(3); // size is fixed.`

`numericArray[0] = 1;`

`numericArray[1] = 2;`

`numericArray[2] = 3;`

* Manipulating Arrays →

→ Manipulating arrays requires sets of properties & methods, which includes. ① Searching ② Sorting array

Methods of Array →

- ① concat() ② every() ③ filter() ④ foreach() ⑤ join()
- ⑥ indexof() ⑦ lastindexof() ⑧ map() ⑨ pop() ⑩ push()
- ⑪ reduce() ⑫ reduceRight() ⑬ reverse() ⑭ shift()
- ⑮ slice() ⑯ some() ⑰ sort() ⑯ splice() ⑯ tostring()
- ⑳ unshift()

F 11 - 2 - 16

H.W Below the car display the description of the car.

→ Use the switch case & pass the count or array element.

Eg:- Image slide show using arrays. | slideshow.html

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
</head>
```

```

<script type="text/javascript">
var pics = new Array ("cars/1.jpg", "cars/2.jpg",
                      "cars/3.jpg", "cars/4.jpg");
var count = 0;
function PreviousClick()
{
    if (count > 0)
    {
        count--;
        document.getElementById("poster").src = pics[count];
        switch (count) ... For description of each image.
    }
}
function NextClick()
{
    if (count < pics.length)
    {
        count++;
        document.getElementById("poster").src = pics[count];
        switch (count) ...
    }
}

</script>
</head>
<body>
<form>
<div>
<table width="400" align="center" border="2"
       cellspacing="30" cellpadding="30">
    <tr> for button we have "click" event but for
         image hyperlink we don't have it so we need
         to write like
    <td> <a href = "javascript : PreviousClick()" > this.
    -> <img src = "images/prev.jpg" width="50" height="50" >
    </a>
    </td>
    <td>
        <img src = "cars/1.jpg" width="400" height="200"
             id = "poster">
        <div id = "description."> </div>
    </td>

```

```

<td>
<a href = "javascript: NeatClicks()">
<img src = "images/neat.jpg" width="50" height="50">
</a> </td>
</tr>
</table>
</div>
</form></body></html>

```

transitions effects by
 using CSS
 zooming, effects, animation
 transparency etc.

12-1-16

* DropDownList Cascading Exp:

- ① document.createElement()
- ② appendChild()

Creating Cascading DropDownList

① document.createElement() :- It is a method used to create any HTML element dynamically.

Ex:- document.createElement("Image"); // this will create img

② appendChild() → It is a method used to add child elements dynamically into a parent element.

Ex:- lstCategories.appendChild("option");

Ex:- OnlineShopping.html

```

<!DOCTYPE html>
<html>
<head>
<script type = "text/javascript">
function bodyLoad()
{
  categories = new Array("Select a Category", "Electronics",
                        "Shoes");
}

```

```

listOfCategories = document.getElementById("lstCategories");
for (var i = 0; i < categories.length; i++)
{
  opt = document.createElement("option");

```

```

opt.value = categories[i];
opt.text = Categories[i];
listOfCategories.appendChild(opt);
}

}

function SelectCategory()
{
    eProducts = new Array ("Select Electronic Product",
                          "LG Mobile", "LED TV");
    sProducts = new Array ("Select Shoe Product",
                          "Nike Shoe", "Lee Cooper");
    listOfProducts = document.getElementById ("lstProducts");
    choice = document.getElementById ("lstCategories").value;
    switch (choice)
    {
        case "Electronics":
            document.getElementById ("lstProducts").innerHTML = "";
            for (var i=0; i<eProducts.length; i++)
            {
                opt = document.createElement ("option");
                opt.value = eProducts[i];
                opt.text = eProducts[i];
                listOfProducts.appendChild (opt);
            }
            break;
        case "Shoes":
            document.getElementById ("lstProducts").innerHTML = "";
            for (var i=0; i<sProducts.length; i++)
            {
                opt = document.createElement ("option");
                opt.value = sProducts[i];
                opt.text = sProducts[i];
                listOfProducts.appendChild (opt);
            }
    }
}

```

```
listOfProducts.appendChild(opt);
}
break;
default:
listOfProducts.value = "Select a Product";
break;
}
}

function SelectProduct()
{
choice=document.getElementById("listProducts");
switch(choice)
{
case "LG Mobile":
document.getElementById("poster").src =
"images/mobile.jpg";
document.getElementById("description").innerHTML =
"product Name: Mobile" + "<br>" + "Price: 6000";
break;
case "LED TV":
document.getElementById("poster").src = "images/
tv.jpg";
document.getElementById("description").innerHTML =
"product Name: LED TV" + "<br>" + "Price: 56000";
break;
--- Similar for case for other products ---
}
}

</script>
</head>
```

```

<body onload = "bodyLoad ()">
<form>
<h1 align = "center"> Amazon Shopping Cart </h1>
<div>
<table width = "400" border = "1" align = "center">
<tr align = "center">
<td>
<select id = "lstCategories" onchange = "SelectCategory ()">
</select>
</td>
</tr>
<tr align = "center">
<td>
<select id = "lstProducts" onchange = "SelectProduct ()">
<option value = "selectProduct">
Select a Product </option>
</select>
</td>
</td>
<td>
<img src = "" id = "poster" width = "400" height = "200" />
<br>
<div id = "description"></div>
</td>
</tr>
</table>
</div>
</form>
</body>
</html>

```

① Hw:- Convert number into words -
500 → five hundred
⇒ To achieve by using Arrays & if condition.

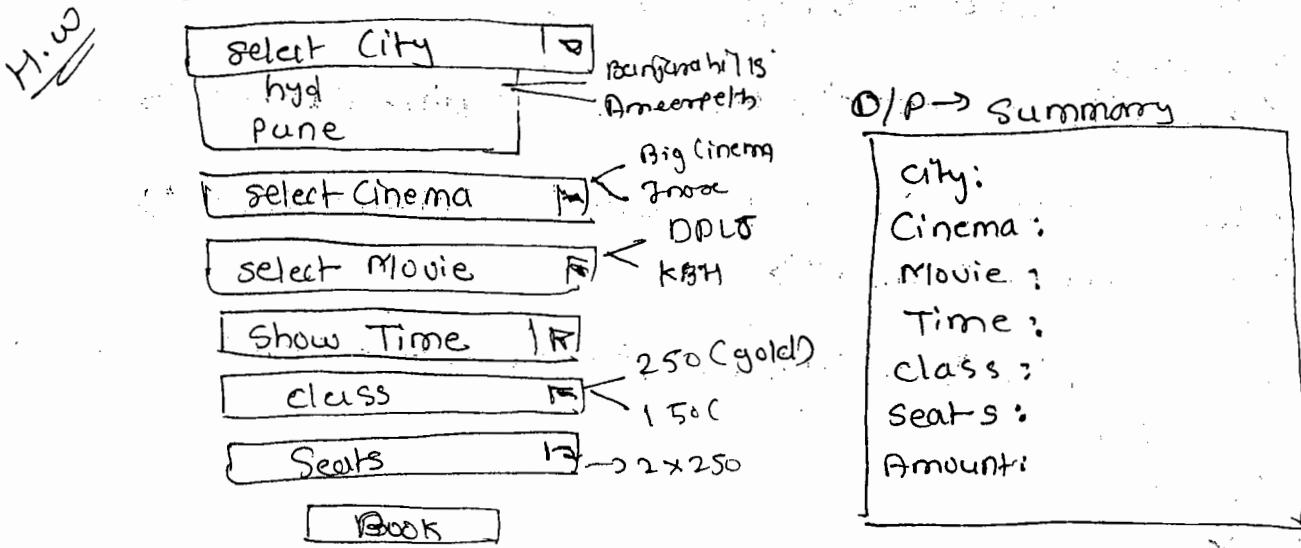
Array ("One", "Two",
Array ("Hundred", "Thousand")
Array ("eleven", "twelve", ...)

500

Submit

five hundred

Movie Ticket Booking.



→ MongoDB (online DB management)

① function getShowTimings()
 { document.getElementById("timing").innerHTML = "";
 gShowTimings = new Array();
 for (var i=0; i < gShowTiming; i++)
 {
 document.getElementById("timing").innerHTML += "

" +
 + "" +

② function AirLiftShowTimings()

{
 // same as above

3 Call this methods in respective switch cases of the
 movie cases

Google Chrome Debug → Sources → Open file → put breakpoint →
 select 1 movie from dropdownlist →
 refresh page & select 2 movie from dropdownlist →
 & press F10 → or [] → see the o/p on
 webpage at LHS & see the values at bottom

15-2-16

JavaScript In-Built Objects

- ① String
- ② Date
- ③ Math

String Functions → Manipulation fn.
Format fn.

me

| <u>Manipulation fn</u> | <u>Format fn</u> |
|------------------------|------------------|
| indexof() | bold() |
| split() | italic() |
| toUpperCase() | big() |
| toLowerCase() | fontSize() |
| charAt() | |
| lastIndexof() | |
| substring() | |
| length → Property | |
| → Match | |

- Q) In how many ways we can validate Email?
- ① use Input type="Email"
 - ② we can use pattern
 - ③ Use string fn to validate email

→ The Javascript document model supports several built-in objects that are used to manipulate strings, dates & mathematical operations.

→ The builtin objects are →

- ① String
- ② Date
- ③ Math

① String → The string object provides set of properties & methods to manipulate the strings. and also to format the string.

→ The String manipulation functions are →

Members

Descriptions

① length → Returns the total count of characters in a string

② indexof() → Returns the index number of specified character in a string.

③ lastIndexof() → Returns the last occurrence of specified character in a string.

④ substring() → Returns the characters present in specified range

- ⑤ `match()` → Returns boolean true if current string matches with the supplied string.
- ⑥ `charAt()` → Returns the "character" at supplied Index.

Ex:- Exp. for Validation of mobile using pattern & name using String functions like `match()`, `indexOf()`, `lastIndexOf()`, ~~int~~ `length` etc.

<!DOCTYPE html>

```

<html>
<head>
<script type="text/javascript">
function SubmitClick()
{
var str = document.getElementById("txtMsg").value;
if (str.length > 10)
{
document.getElementById("msg").innerHTML = "more than
    chars only";
}
else
{
document.getElementById("msg").innerHTML = "Available";
}
var mobile = document.getElementById("txtMobile").value;
var pattern = /\+\d{1}[0-9]{10}/; Note → enclose patterns
if (mobile.match(pattern))
{
document.getElementById("msgMobile").innerHTML = "mobile
    is valid";
}
else
{
document.getElementById("msgMobile").innerHTML = "Invalid
    mobile";
}

```

```

var email = document.getElementById("txtEmail").value;
var atPos = email.indexOf("@");
var dotPos = email.lastIndexOf(".");
if (atPos < 2 || dotPos - atPos < 2) {
    document.getElementById("msgEmail").innerHTML = "Invalid
    Email";
} else {
    document.getElementById("msgEmail").innerHTML = "Valid
    Email";
}

</Script>
</head>
<body>
<form>
<div>
    Enter Name:
    <input type="text" id="txtMsg">
    <span id="msg"></span>
    <br>
    Mobile:
    <input type="text" id="txtMobile">
    <span id="msgMobile"></span>
    <br>
    Email:
    <input type="text" id="txtEmail">
    <input type="button" value="Submit" id="btnSubmit"
        onclick="submitClick()">
</div>
</form>
</body>
</html>

```

Note → If specific string / characters specified in `indexOf()` & `lastIndexOf()` is not found then these methods will return "-1" so we can use it as checking / searching any char / substring in given string.

Ex → Enter Email Address: [submit]

→ Email Id : Rahul
→ Domain : gmail.com

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
email = document.getElementById("txtEmail").value;
var atPost = email.indexOf("@");
var id = email.substring(0, atPost);
var domain = email.substring(atPost+1, email.length);
document.getElementById("idspan").innerHTML = id;
document.getElementById("domainspan").innerHTML = domain;
</script>
</head>
```

Ex → Enter City: [span found.]

If Delhi or whatever city is entered if it is present in list, then display found in span tag beside TextBox.

```
<input type="text" id="txtCity" list="lstcities">
<datalist id="lstcities">
<option value="mumbai"> Mumbai </option>
```

```

var search = txtName.value;
var cities = new Array ("Delhi", "Mumbai");
on button click {
    if cities.indexOf("search") == -1
        // for not found
}

```

16-2-16

String Format Functions

→ The string format functions enables to change the string appearance dynamically, which includes font, color, size & effects.

Method Name

- (1) big
- (2) blink
- (3) bold
- (4) fixed
- (5) fontcolor
- (6) fontsize
- (7) italics
- (8) small
- (9) strike
- (10) sub
- (11) sup
- (12) toLowerCase
- (13) toUpperCase

Example:-

```

"str".big()
"str".blink()
"str".bold()
"str".fixed()
"str".fontcolor("Green")
"str".fontsize(1)
"str".italics()
"str".small()
"str".strike()
"str".sub()
"str".sup()
"str".toLowerCase()
"str".toUpperCase()

```

Ex:- Ex. for string f() (changeCase)

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function ChangeCase() {
    var name = document.getElementById("txtName").value;
    document.getElementById("txtName").value = name.toUpperCase();
}
</script>

```

```
</head>
<body>
<form>
<div>
```

Enter Name :

```
<input type="text" id="firstname" onblur="changeCase()"
       placeholder="Block Letters">
```

```
</div>
```

Password :

```
<input type="password" id="firstnamepassword">
```

```
</div>
```

```
</form>
```

```
</body>
```

```
</html>
```

problems related to Dates →

- ① Bug → Inbuilt problem in S/w.
- ② Virus → (Vital Information Under Seized)
external problem in P/S/W
- ③ Trozen
- ④ Worms.

Bigest Bug → Y2K.

12/31/99

01/01/00

Q) Interview → print 18, wednesday - 2016 as o/p. & every time day changes then I should be able to see current date.

Date Object

→ The date object provides set of properties and methods that are used to manipulate the dates however it will not allow to change or affect server dates.

Syntax →

```
var now = new Date();
```

| Method | Description |
|---------------------|--|
| ① now.getDate() | Returns the day number (eg 16) |
| ② now.getDay() | Returns weekday number (eg. Tuesday - 2) |
| ③ now.getMonth() | Returns month numbers (ie January - 0) |
| ④ now.getHours() | Returns hours in 24 hrs format |
| ⑤ now.getMinutes() | Returns minutes |
| ⑥ now.getSeconds() | Returns seconds |
| ⑦ now.getFullYear() | Returns full year (ie 2018) |

Ex-→ Print Current Date (ie 16, Tuesday - 2018)

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
    function bodyload()
    {
        var days = new Array("Sunday", "Monday", "Tuesday",
                            "Wednesday", "Thursday", "Friday", "Saturday");
        var now = new Date();
        document.getElementById("msg").innerHTML =
            now.getDate() + "," + days[now.getDay()] + "-" +
            now.getFullYear();
    }
</script>
</head>
<body onload="bodyload()">
<form>
```

```
<div id="msg"></div>  
</form>  
</body>  
</html>
```

- ⇒ Math → object to perform mathematical operations
- ⇒ MathML → It is a Library to design the mathematical equations, not used for performing mathematical operations.

Pass:

③ Math Object

- The math object in JavaScript provides functions that are used to perform various mathematical operations. → In Real world development math functions are used to design widgets like loan calculator, BMI calculator, currency converter etc.
- The commonly used mathematical functions in JavaScript are ⇒

Method Name

- ① abs
- ② acos
- ③ asin
- ④ atan
- ⑤ ceil
- ⑥ cos
- ⑦ exp
- ⑧ floor
- ⑨ log
- ⑩ max

Example

- Math.abs(-6.5)
- Math.acos(0.5)
- Math.asin(1)
- Math.atan(0.5)
- Math.ceil(7.6)
- Math.cos(0.4)
- Math.exp(8)
- Math.floor(8.9)
- Math.log(5)
- Math.max(1, 700)

- | | |
|-------------|----------------------------|
| (16) min | Math.min(2,700) |
| (17) pow | Math.pow(5,2) |
| (18) random | Math.random() |
| (19) round | Math.round(.567) |
| (20) sin | math.sin(Math.PI) |
| (21) sqrt | Math.sqrt(9801) |
| (22) tan | Math.tan(11.5 * Math.PI) |

~~Ex:-~~ → ReCAPTCHA → google's site

google → reCAPTCHA → GetCaptcha →

Signin → Label = (HTML Captcha demo)
Domain = (localhost)

=> sitekey , > Secretkey , <div> it will provide -

→ Google Developer → Guide → PHP plugin
ASP.NET plugin.

The recaptcha documentation site ← hyperlink at bottom

→ we have to download Libraries .dll for server side reCAPTCHA implementation.

Help for Math.

① EMF Calculator (Design)

| Description | Your Values |
|----------------------------|-----------------------|
| Loan Amount | 50000 |
| Loan length in months | 48 |
| Interest Rate | 3 |
| Monthly Payment | 11067.--- |
| Calculator | Reset |

Formula \rightarrow ① interest =

② payment = princ * int / (1 - (Math.pow(1 / (1 + int), term)))

(cont)

$$\text{payment} = \text{principle} * \text{monthly interest} / (1 - (1 + \text{Monthly Interest}) * \text{Months}))$$

Ex-① BMI Calculator (Hospital websites)

H.W

| | | |
|--|----------------------|--------|
| weight : | <input type="text"/> | kg. |
| Height : | <input type="text"/> | meters |
| BMI : | <input type="text"/> | |
| <input type="button" value="Calculate"/> | | |

$$\text{BMI} = \frac{\text{Weight (kg)}}{(\text{height})^2 (\text{mtr})}$$

- * also show the status by using if Condition & tell the status to the user.
- * Download Interview questions document from size's blog. 17-2-16

Events In JavaScript

- In object based programming systems, the actions are called when any event triggers. Event specifies the actions performed by the user.
- The JavaScript Events are classified into 3 major Categories :-

- ① Mouse Events
- ② key Events
- ③ miscellaneous Events.

- ① Mouse Events → The mouse events controls the actions to be performed when user handles the mouse. The commonly used events are :-

| Event | Description |
|---------------|---|
| ① onmousedown | → Indicates actions to be performed when the mouse button is press hold down. |

- ② onmouseup → Actions to be performed when mouse button is released.
- ③ onmouseover → Actions to be performed when mouse pointer is over the element
- ④ onmouseout → Actions to be performed when the pointer is moved out of the element.
- ⑤ onmousemove → Actions to be performed while user is moving the mouse pointer on the screen.
- ⑥ onclick → mouse button click.
- ⑦ ondblclick → mouse button double click.
- ⑧ oncontextmenu → mouse right button click. popup menu, context menu, dropdown menu

- me
- 1> pop-up menu
 - 2> Context menu
 - 3> dropdown menu

Mouse Properties →

- ① screenY → returns page Y axis offset co-ordinates
- ② screenX → returns page X axis offset co-ordinates
- ③ clientX → returns cursor current position on X-axis.
- ④ clientY → returns cursor current position on Y-axis.
- ⑤ altKey → returns boolean true if alt key pressed.
- ⑥ ctrlKey → returns true.
- ⑦ shiftKey → returns true (only in Mac PC's)
- ⑧ metaKey → returns true

Eg. → Eg. to demonstrate ① onmouseover ② onmouseup
 ③ onmousedown ④ onmouseout

```
<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
function Ad1()
{
document.getElementById("ads").src = "images/pepsi.jpg";
}
function Ad2()
{
document.getElementById("ads").src = "images/pepsi2.png";
}
function color1()
{
document.getElementById("head1").style.color = "red";
}
function color2()
{
document.getElementById("head1").style.color = "green";
}
function bulb1()
{
document.getElementById("bulb").src = "images/bulbon.png";
}
function bulboff()
{
document.getElementById("bulb").src = "images/bulboff.png";
}
</script>
</head>
<body>
<form>
<div>

<div>
<h1 id="head1" onmouseover="color1()" onmouseout="color2()">
    Change Color on mouse over
</h1>

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

```

Mouse events with CSS plugins to

zooming image.

imagezoom.html

* Download CSS plugins

```

<!DOCTYPE html>
<html>
<head>
<script></script>
<style>
    #gallery img {
        width: 200px;
        height: 200px;
        -webkit-transition-duration: 0.6s; // for chrome, FF
        -moz-transition-duration: 0.6s; // for firefox
        -o-transition-duration: 0.6s; // for opera
        opacity: 0.6;
        z-index: 1;
        margin: 0;
        position: relative;
    }

```

```

#gallery img:hover {
    -webkit-transform: scale(1.5);
    -moz-transform: scale(1.5);
    -o-transform: scale(1.5);
    box-shadow: 0px 0px 25px gray;
    -webkit-box-shadow: 0px 0px 25px gray;
    -moz-box-shadow: 0px 0px 25px gray;
    opacity: 1;
    z-index: 10;
}
}

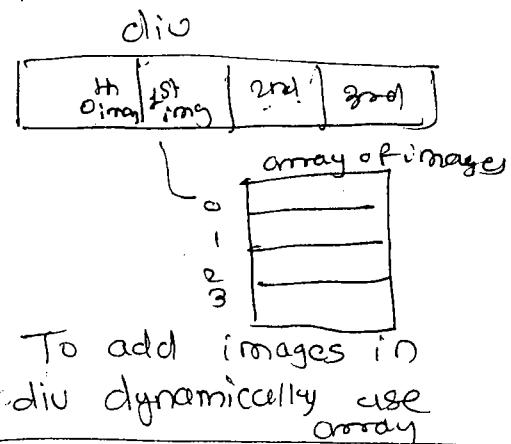
</style>
<Script type="text/javascript">
function bodyload()
{
    var pics = new Array("images/1.jpg", "images/2.jpg",
        "images/3.jpg", "images/4.jpg");
    var container = document.getElementById("images");
    for (var i=0; i<pics.length; i++)
    {
        var img = new Image(); // Create element dynamically
        img.src = pics[i];
        img.width = "200";
        img.height = "100";
        container.appendChild(img);
    }
}
</Script>
</head>
<body onload="bodyload()">
<marquee scrollamount="10"
    onmouseover="this.stop()"
    onmouseout="this.start()"
    style id="gallony" >

```

```

<div id="images"></div>
</marquee>
</body>
</html>

```



JavaScript Functions & Parameters

18-2-16

- The functions in JavaScript can be defined with parameters which are referred as formal parameters
- The values that are passed while calling the function are known as "Actual parameters"
- You can access the parameters & their values by referring to the ① parameter name or By using the keyword ② arguments.

Eg: ① function using parameters.

```

<script type="text/javascript">
function showMessage(city1, city2)
{
    alert("City1:" + city1 + " " + "City2:" + city2);
}
showMessage ("Delhi", "Chennai");
showMessage ("10", "20");
</script>

```

Eg: ② function parameters accessed using argument keyword.

```

<script>
function showMessage(fname, lname)
{
    alert("Hello!" + argument[0] + " " + arguments[1]);
}

```

```
ShowMessage ("Ajay", "Kumar");
```

```
<script>
```

Ex ③ Accessing all arguments passed into the function

```
<script>
```

```
function ShowMessage ()
```

```
{
```

```
for (var i = 0; i < arguments.length; i++)
```

```
{
```

```
alert (arguments [i]);
```

```
}
```

```
}
```

```
ShowMessage ("Delhi", "Hyd");
```

```
<script>
```

Function with Return

```
<script>
```

```
Ex: function Sum (n1, n2)
```

```
{ return n1 + n2;
```

```
}
```

```
var result = Sum (10, 20)
```

```
alert (result);
```

```
<script>
```

→ JavaScript supports functions that can return a value, the function return values are often referred by using the keyword "return".

* A function can have another function inside it.

```
function Multiply (x)
```

```
{ function fn (y)
```

```
{ return (x * y), }
```

```
return fn;
```

imagine
like
a
class

```
vara = multiply (3)
```

```
a(2) = 6
```

```
a(3) = 9
```

a is treated like a
constructor function as it is referring
multiply functions

- * function Using Another function
- JavaScript is object based programming system, that allows to encapsulate set of properties and functions.

Ex:- <Script>

```

function multiply(x)
{
    function fn(y)
    {
        return (x * y);
    }
    return fn;
}
var a = multiply(3);
alert(a(2));           // output => 6
alert(a(3));           // output => 9

```

* Functions As Expressions

- You can create a function & designate it as an expression by assigning it to any object, you can call the function with reference of the object

Ex:- <Script>

```

var add = function(a, b)
{
    return a + b;
}
alert(sum(10, 20));      // Invalid
alert(add(10, 20));     // 30

```

<script>

functions in Javascript.

- * Anonymous functions in Javascript
- functions without any specific name are designated as anonymous functions.
- These functions will allow to pass the parameters directly into the function signature, however the functions are referred by using an object.

Ex:- <script>

```

var obj = function () {
    {
        alert ("hello!");
    }
}

var object = function (fname) {
    {
        alert ("hello!" + fname);
    }
}

alert (obj());
// output → hello!
alert (object("Ajay"))
// output → hello! Ajay
</script>

```

* Calling A Function On Specific Event
 (Anonymous) <input type="button" id="btnSubmit" value="Submit" />

Ex:- <script>

```

document.getElementById ("btnSubmit").onclick =
    new function () {
        {
            alert ("Submitted");
        }
    }

```

<script>

```

var btn = document.getElementById ("btnSubmit");
btn.onclick = new function () {
    {
        alert ("Submitted..");
    }
}

```

(e)

~~Temp interview~~ Disable any event by calling a function
 → events in Javascript can be disabled when the function called by the event return to boolean
false

- ① oncopy
- ② on paste
- ③ oncut
- ④ oncontextmenu
- ⑤ onselect
- ⑥ onselectionstart

```
<body oncut = "return false">  
  body.oncut = new Function() { return false };
```

Ex → For "onmousemove" event.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
  #flying {  
    position: fixed;  
  }
```

```
</style>  
<script type = "text/javascript">  
function UpdateFlyingObj(event)  
{  
  var mouseX = Math.round(event.clientX);  
  var mouseY = Math.round(event.clientY);  
  var flyingObj = document.getElementById("flying");  
  flyingObj.style.left = mouseX + "px";  
  flyingObj.style.top = mouseY + "px";  
}
```

```
</script>
```

```
</head>  
<body onmousemove = "UpdateFlyingObj(event)">  
  <div style = "height: 2000px;"> Move the mouse  
  pointer to test </div>  
  <img id = "flying" src = "images/flag.gif"/>  
</body>  
</html>
```

KeyPress Events

① onkeyUp ② onkeydown ③ onKeyPress

Properties → 1) charCode (same as keyCode)
 2) keyCode (returns which key is pressed)
 3) which (which key is pressed)

19-2-18

- Q) write a code to check Capitalization without using pattern?
- (Q) check capslock is on/off?
- ⇒ Using keypress & events & properties.

JavaScript key Events

→ The key events are responsible for managing actions of users based on key strokes/strikes. You can control functionality by using various key events like -

| Event | Description |
|--------------|--|
| ① Onkeyup | → actions to perform when key released. |
| ② Onkeydown | → actions to perform when you hold or down the keys. |
| ③ onkeypress | → actions to perform on key press. |

| Property | Description |
|------------|--|
| ① charCode | → specifies the key code according to UTF (unicode town) standard. |
| ② keyCode | |
| ③ which | |

Ex:- Validating the password using key event using onkeyup, onkeypress, events & charCode/which/keyCode properties.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

Password.html

```

<script type="text/javascript">
function ValidatePassword()
{
    var pwd = document.getElementById("txtPwd").value;
    var pattern = /C?=[.*[A-z]]\w{4,15}/;
    if (pwd.match(pattern))
    {
        document.getElementById("status").value = "20";
        document.getElementById("msg").style.color = "green";
        document.getElementById("msg").innerHTML = "Strong password";
    }
    else
    {
        document.getElementById("status").value = "20";
        document.getElementById("msg").style.color = "red";
        document.getElementById("msg").innerHTML = "Weak password - Use atleast one uppercase letter";
    }
}
function GetCode(e)
{
    document.getElementById("result").innerHTML =
    e.charCode || e.keyCode || e.which;
}

</script>
</head>
<body>
<form>
<div>

```

↓
we can use only 1 out of 3
but its better practice to write
all 3, because it should support
all browsers.

we can say type = "password"
just for testing we kept it as text.

Enter Password: →

```

<input type = "text" id = "txtPwd">
    onkeyup = "ValidatePassword()>
<span id = "msg"> </span>
<meter min = "1" max = "100" value = "1">
    id = "status">
</div>
<div>
    Ex:- Enter any char:
    <input type = "text" id = "txtChar">
        onkeypress = "GetCode(event)">>
    <span id = "result"> </span>
</div>
</form>
</body>
</html>

```

(Q) How to avoid / restrict to use keyword as general variable?
 ⇒ at top <Script> after this write
"use strict" ie

Ex:- Main Logic for checking CapsLock Status?

on keypress

```

<Script type = "text/javascript">
    use strict
    function CheckCapsLock(e)
    {
        var capsLockON;
        keyCode = e.keyCode ? e.keyCode : e.which;
        shiftKey = e.shiftKey ? e.shiftKey : ((keyCode == 16));
        ? true : false);
    }

```

```

if(CC keyCode >= 65 && keyCode <= 90) && !shiftKey)
  || ((keyCode >= 67 && keyCode <= 122) && shiftKey))
{
  capsLockON = true;
}
else
{
  capsLockON = false;
}

if (capsLockON)
{
  document.getElementById('divWarningCapslock').style.
    visibility = 'visible';
}
else
{
  document.getElementById('divWarningCapslock').style.
    visibility = 'hidden';
}

```

</Script>

JavaScript Miscellaneous Events.

| Event | Description |
|--------------------|--|
| ① onclick | → Actions to perform when element clicked. |
| ② onchange | → Actions to perform when selection changed. |
| ③ onselect | → Action on Selection. |
| ④ onselectionstart | → Actions when selection extended. |
| ⑤ oncut | → Actions when cut operation performed. |
| ⑥ oncopy | → Actions on copy. |
| ⑦ onpaste | → Actions on paste. |

- ⑥ onfocus → Actions when control gets focus.
- ⑦ onblur → Actions when control loses focus.
- ⑧ oncontextmenu → Actions when user right clicks on any element.
- ⑨ ondoubleclick → Actions on double click.
- ⑩ onsubmit → Actions when form submit clicked.
- ⑪ onreset → Actions when reset clicked.

Eg:- Disable selections, cut, copy, paste and right click in a web page.

```

<!DOCTYPE html>
<html>
<head>
<script type="text/javascript">
document.getElementByIdentId(
document.oncontextmenu = new Function("return false");
document.onselectstart = new Function("return false");
document.onselectstart = new Function("return false");
if(window.sidebar)
{
document.onmousedown = new Function("return false");
document.onclick = new Function("return true");
document.oncut = new Function("return false");
document.oncopy = new Function("return false");
document.onpaste = new Function("return false");
}
</script>

```

```
<head>
<body>
    how to Disable cut, copy, paste & right-click
    and context menu using Javascript
</body>
</html>
```

* OnSubmit and onReset.

- These are the events that indicates action to be performed when user clicks Submit or Reset buttons.
- However Submit and Reset button by default will use click event. Hence onsubmit and onreset are defined on form.

Ex :- <script>

(later)

```
function SubmitClick()
{
    alert ("Form Submitted");
}

function ResetClick()
{
    alert ("Form will reset");
}

</script>
<body>
<form onsubmit = "SubmitClick()"
      onreset = "ResetClick()">

    <div>
        name:
        <input type = "text" id = "txtName" />
        <br>
        <input type = "submit" value = "Submit" />
        <input type = "reset" value = "Reset" />
    </div>
</form>
</body>
</html>
```

Note → onsubmit & onreset events are very important for server side programmers to identify whether its get request or post request and if post request then capture the values from query string. (★ when we click the button "Submit" button then the value's are submitted through query string).

Animations →

- ① setTimeout()
- ② clearTimeout()

Ex:- f1()

{

 image.left = current position + 1px
 of image

3

 Start Stop.

↓

⇒ on start button click → milliseconds.

animate = setTimeout (f~~1~~ f1, 100);

⇒ on stop click →

clearTimeout (animate);

→ This Example is present in Sir's blog.

in Bootstrap →

www.getbootstrap.com

↳ JavaScript (menu)

↳ examples at right side.

↳ select any example & see it.

↳ for this to implement we need 2 files

① bootstrap.css

② bootstrap.js