web applicationBrowsing

- * Browser is a software for your computer
- * It is the core of the web pages
- SQL - Superior
- Open

Search

- Search Engine
- Browsers
- Based on

Server

Server is

Datab

- * It is a collection of data
- * Database
- the Data

Be

Business

log

web

web is an information space which contains web related resources, such as images, audio, video, text, documents and etc ---

web development

web development is the process of developing web applications, managing web application and hosting

A HTTP

* It is

the ser

C

* https

web application

Browser

- * Browser is a client side software which is installed in your computer and allows us to access webpages.
 - * It is the only software by using which we can access the web pages.
- Safari, google chrome, internet explore, mozilla fire fox, opera, NO browser

Search Engine

Search Engine is a software which is installed in browsers and allows us to search the information exist on web. ex:- google, yahoo etc

Server

Server is a computer where all the websites are hosted

Database server

- * it is used to store the data in server
- * Database server is media which is used to store the data

Business logic

Business logic server is a where we write the logic.

HTTP

- * HTTP stands for Hyper text transfer protocol
- * It is responsible to carry request from client to the server as well as response from server to client.
- * https is used to provide secure connection between

contains web
o, video,

ng web
hosting

HTTP from server to client

HTML

- * HTML stands for Hyper text Markup language
- * HTML is used to create a structure ~~or~~ layout out of the web page.
- * By using HTML we can create static pages.
- * (.htm) and (.html) are the only extensions which are supported by the browser.

CSS

- * CSS stands for Cascading style sheet
- * CSS is used to apply designs and formation on the web project (makeup)
- Javascript is client-side scripting language, which means the source code is processed by the client's web browser rather than on the web server
- * JavaScript is client side scripting language which is used to make web pages dynamic

Web page

single HTML document is known as Web page

Websites

collection of Web pages is known as websites

Bootstrap

Bootstrap is a framework of CSS which is used to develop responsive web pages ~~or~~ websites

MERN → MongoDB ExpressJS ReactJS NodeJS
NodeJS

MERN → MongoDB ExpressJS ReactJS NodeJS

Father of web tech → Tim Berners-Lee

Father of CSS → Håkon Wium

Father of JS → Brendan Eich

Bootstrap is invented by Twitter

27/6/19

Sub-IDE Semi IDE → Integrated Development Environment

- 1) Note Pad
- 2) Note Pad ++
- 3) Edit +
- 4) Eclipse
- 5) Sublime Text
- 6) Visual Studio

current JavaScript = ECMAScript 2018

current HTML ver = 5

current CSS ver = CSS3

Structure of HTML it means belongs to html file

<!DOCTYPE HTML> → ① Version Information
<HTML> → header file

<HEAD>
</HEAD>

} → ② Head section

<Body>
</Body>

} → ③ Body section

</HTML>

which is used to in Head tag you can use only 5 tags

- ① <meta/> → it is used to store webpage info which is not visible
- ② <link/> → it is used to insert external CSS
- ③ <title> </title> → it is used to search title
- ④ <styles> </styles> → CSS
- ⑤ <script> </script> → JS

Angular JS

React JS node JS

Different types of tag in HTML

- 1) paired Tags / Double Tags
- 2) unpaired Tags / single Tags

1) Paired Tags

<Tag name> content </Tag name>

Syntax:- ~~Diff~~

<p> Data </p>

<HTML> </HTML>

<body> </body>

<form> </form>

2) Unpaired Tags

<Tag name>

<1 Tag name>

<2 Tag names>

Syntax:- <link/>

<meta/>

<hr/>

Attributes

Attributes are the properties which provide extra information to the element or tag.

Effect to the element or tag

Syntax:- <body bg = "green"> → we can also pass with single code
</body> or without

paragraph tag and break tag

<p> </p>

Attributes Parameters

(1) Align left (default)

right,

center

justify

paragraph tag is used to insert or create paragraph on web page

$\langle b\sigma | \rangle$

Break tape is used to break the lines.

HTML and CSS are case sensitive

Project:

<!doctype html>

Lecture 13

L kneed

<1 head>

< body >

Welcome to web technology class
 welcome to
web Technology class

21 bodies

21 inton1>

option -L !doctype html

Lithos 1 >

`<head>`

<1 head>

2 body's

Welcome to web technology (class 2 hr/1) Welcome
to web technology (class 2 hr/1)

LPS welcome to web technology class LPS

< p align="right" > welcome to web technology

```
<P align="center"> Welcome to Web Technology  
rcaos </P>
```

Lp class="jacket">welcome to web technology

class welcome to web Technology class

\rightarrow \leftarrow \downarrow \uparrow $\Rightarrow \perp P$

• Libdys

2 (4th or 1)

Attributes of Body tag
the visible part of web page

<body>	</body>	css
1) Background	color name, color code	
2) Text	color name, color code	
3) Background	Image path, Image name	

proj:- <!doctype html>

<html>

<head>

</head>

<body> bgcolor="yellow" text="white" background
= "c:\udent\-----">

welcome web technology
 welcome webTechnology

</body>

</html>

proj!:- <!doctype html>

<html>

<head>

<title> JspDemos </titles>

<link href="1.jpg" rel="icon" />

</head>

came here previously

</body>

</html>

Image tag

Attribute

Parameter

(1) src

Image name

Image path

(2) Title

Any name → used to if we pass title

(3) Alt

Any name wherever we place over the center on image it shows

(4) Height

px, pt → percentage

(5) width

px → pixel

<!DOCTYPE html>

pt → point

<html>

% → percentage

<head>

<title> JSR100 </title>

<link href="1.jpg" rel="icon" />

</head>

href → the href attribute specifies the link's destination

<body>

<img src="1.jpg" height="200px" width="200px"

title="Fareen cricket player">

<img src="mala.jpg" height="200px" width="200px"

<img src="ai.jpg" height="200px" width="200px"

alt="Dhoni">

</body>

</html>

Marquee Tag

Marquee tag is used to apply movement to the html element.

<marquee> </marquee>

Attributes

Parameter

i) Behavior

scroll (Default)

slide

Alternate

2) Direction

left (Default), right, up, down

3) Height

Right, top, down, left, v.

4) width

px, pt, %

5) Bg color

color name, code

6) Loop

any number

7) scroll amount

-11-

8) scroll delay

time in millisecond

prty:- <!doctype html>

<html>

<head>

</head>

<body>

<marquee scroll amount = "50"> Dingi </marquee>

<marquee behavior = "slide" loop = "3">

bg color = "red" scroll delay = "1000"

direction = "right" > Dingi </marquee>

<marquee behavior = "alternate" scroll amount = "100"

bg color = "red" height = "200px" width = "100px"

direction = "up" > Dingi </marquee>

</body>

</html>

we can pass any no of tag

movement to the

up, down
left, right

```
prg1: <!DOCTYPE html>
<html>
<head>
</head>
<body>
```

→ we can also pass paragraph tag

```
<marquee><img src = "1.jpg"/></marquee>
<marquee><img src = "2.jpg"/></marquee>
```

we can pass paragraph

Order list

OL > → order list also

Attributed

list is used to arrange html elements in proper order

there are two types of list **or**

- order list
- unorder list

Attributed	parameter	OL > → order list	list item
1) TYPE	1 (default), A, a	 	
2) start	Any Number	401>	

unorder list → unorder list

Attributed	parameter
1) TYPE	DISC (default) - square circle

<!doctype html>

<html>

<head>

</head>

<body>

<ol type="1">

 Java

→ same

 JavaScript

 Python

 HTML

 CSS

<ol type="A">

- same here above

<ol type="a">

same here above

<ol type=":">

same here above

<ol type="I">

same here above

<ol type="A" start="8">

same as above

same

same

same

prg2:-

4

L

L

L

L

L

then it will start
"H" alphabet

curve ball above

<ul type="square">

some more above

<ul type="circle">

same as above

prob:-

<body>

 programming (conquer)

 Java

 python

 c

 scripting (complex)

<ul type="square">

 VB script

 JAVASCRIPT

1

1

1

1

<body>

<html>

"will start
"H" alphabet

<tr>

<td>

<td>

Font Tag

Attributes

- 1) color
- 2) face
- 3) size

Parameters

- color name, code
Font - family name
px, pt, %.

proj:- <!doctype html>

<html>

<head>

</head>

<body>

Dhruv
TFC Dhruv

</body>

</html>

Creating table using htmlStructure of Table

Table is a collection of cells

in html By Default Table has no borders

<table>

<tr> → Table Row

<th>

<th>

<th>

<th>

Table Header

</tr>

Attributes

1) Align

2) Valign

3) Bgcolor

<td>

Following

Attributes

- 1) border
- 2) Height
- 3) weight wi
- 4) Bgcolor
- 5) cell padding
- 6) bordercolor
- 7) cell spacing
- 8) rules
- 9) align

1) Trig

- 1) rowspan
- 2) colspan
- 3) Align
- 4) Valign

5) Bgcolor

<tr>

<td>

<td>

<td> ? Table Data

</tr>

<table>

Following are attributes of Table

→ inside Table tag

Attributes Parameter

- 1) border px, pt, %.
- 2) height — —
- 3) width width — —
- 4) bgcolor color name, code
- 5) cell padding px, pt, %.
- 6) bordercolor color name, code
- 7) cell spacing px, pt, %.
- 8) rules border, colr, all
- 9) align left, right, center

<th> / <td>

Attributes

parameters

- 1) rowspan Any number
- 2) colspan — —
- 3) Align left, right, center
- 4) Valign middle (Default)
- 5) Bgcolor color name, code

valign → Vertical Align

<tr> <td>

Attributes parameters

- 1) Align left, right, center
- 2) Valign middle (Default)
- 3) Bgcolor color name, code

f cells

no borders

} Table Header

prel:- < !doctype html>

<html>

<head>

</head>

<body>

<table border="1" height="200px" width="300px"
 border-color="yellow" cellpadding="8px" cellspacing="20px"
 align="center" rules="all" bordercolor="black" border="1"
 background="100px" >

<tr>

<th> USN </th>

<th> name </th>

<th> Branch </th>

<th> marks </th>

<td>

<td>

1234 </td>

<td> shilpa </td>

<td> ECE </td>

<td> 50% </td>

</tr>

<tr>

<td> 3456 </td>

<td> Meenatha </td>

<td> ME </td>

<td> 90% </td>

</tr>

</table>

:

prel:-

<!doctype r

<html>

<head>

</head>

<body>

<table>

<tr border="1">

<tr>

<td>

<td>

<td>

<td>

<td>

</td>

</table>

proj1:
 <html>
 </html>
 <head> </head>

<body>

<table>

<tr> borders

<tr>

<th> USA </th>
 <th> Name </th>
 <th> Marks </th>
 <th> Branch </th>

</tr>

<tr>

<td rowspan="2" valign="bottom">123</td>

</tr>

<tr> bg-color="yellow" width="200" height="400"

<td> ABCDE </td>

</tr>

</table>

<table border="1">

some help above *

<th colspan="2" rowspan="2" style="text-align: center;"></th>

</

Assignment

A	B		
D	E	C	
F			
G	H	I	J
K			

student	name	dept	marks	Y.
feleety	-	-	-	
feleety	-	-	-	
POP	name	dept	evening	

HR Tag

- 1) hr tag is used to create horizontal lines on the web page
- 2) it is unpaired tag

<hr/>

Attributes

Parameters

- 1) color color code, name
- 2) size width px, pt ; t. (100% default)
- 3) Align center (default) left, right
- 4) no border
- 5) size px, pt, t.

<!doctype

<html>

<head>

<body>

<spider>

<hr color

ISpider

<hr color

ISpider

<hr color

ISpider

<hr color

</body>

2nd

margin

width

height

t;

part: <!DOCTYPE html>

<html>

<head> </head>

<body>

<spider>

<hr color="red" size="50px" />

<spiders>

<hr color="green" size="50px" width="50%"/>

<spiders>

<hr color="purple" size="50px" width="80%" align="left" style="border: 1px solid black; margin-left: 10px; margin-right: 0; border-collapse: collapse; padding: 0; margin-top: 10px; margin-bottom: 10px; display: inline-block; vertical-align: middle; text-align: left; font-size: 1em; line-height: 1.2; font-weight: normal; font-style: normal; font-family: inherit; color: inherit; text-decoration: none; background-color: transparent; border: none; outline: none; position: relative; z-index: 1; transition: all 0.3s ease 0s; border-radius: 0; width: 100%; height: 100%;>

<spiders>

<hr size="50px" width="50%" align="left" style="border: 1px solid black; margin-left: 10px; margin-right: 0; border-collapse: collapse; padding: 0; margin-top: 10px; margin-bottom: 10px; display: inline-block; vertical-align: middle; text-align: left; font-size: 1em; line-height: 1.2; font-weight: normal; font-style: normal; font-family: inherit; color: inherit; text-decoration: none; background-color: transparent; border: none; outline: none; position: relative; z-index: 1; transition: all 0.3s ease 0s; border-radius: 0; width: 100%; height: 100%;>

</body>

assignment

2 ftds (fd)

Assignment

wowee	image	profession	About
default	'	'	'
phani	'	'	'
'	'	'	'

default)

left, right

Audio and Video Tag

<audio> / <video>

Attributes

1) controls

2) height

3) width

Parameter

\rightarrow mandatory

<source>

Attributes Parameter
1) src audio/video

pryl:- <1. document.html>

<html>

<head> <title>

<title> Audio and video </title>

<body> <link href="animating.css" rel="stylesheet">

<head> </head>

<body>

<audio controls>

<source src="1.mp3">

</audio>

<video controls>

<source src="1.mp4">

</video>

<body>

</body>

Iframe Tag

<iframe> iframes

Attributes parameter

1) src HTML filename

Any URL

iframe tag
(①) multiple

pryl:- <1. document.html>

<html>

<head>

<title>

<link href="animating.css" rel="stylesheet">

<body>

<script>

iframe tag is used to insert multiple web pages
 (i) multiple websites on a same web page

(source)

buttons Parameter,
audio/video

HTML:- <!DOCTYPE html>

<html>

<head>

<title> index</title>

<link href="univ.jpg" rel="icon">

</head>

<body>

<iframe src="Addto.html"></iframe>

<iframe src="Video.html"></iframe>

<iframe src="https://www.naukri.com" height="400px"
width="800px"></iframe>

<iframe src="https://www.unist.com"></iframe>

</body>

HTML:- <!DOCTYPE html>

<html>

<head>

<title> Audio </title>

<link href="aemr.a.jpg" rel="icon" />

</head>

<body>

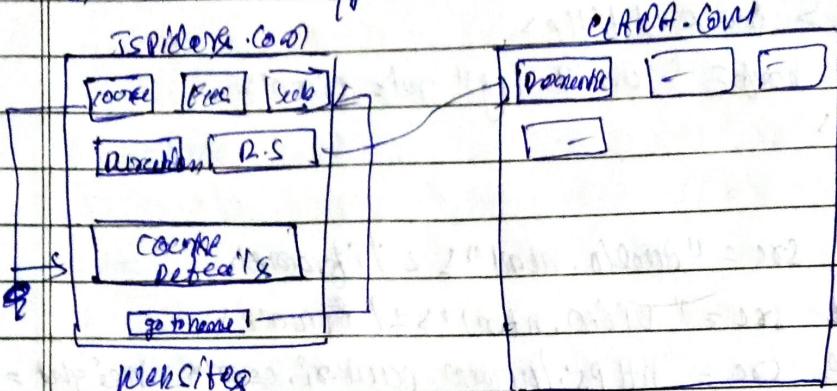
<Addto controls>

some need previous Addto & video tag

but do separate

Hyperlinks

- Hyper links can be created by using Anchor tag which is indicated by `<a>`
- There are two types of hyperlinks
 - Intrafiled hyperlink
 - External hyperlink



`<a>` > one directory

Attributes Parameter

1) href HTML File name

Any URL

2) Title Any name

3) Target - parent (Default)

- Blank

4) Download Any name

prgl:-

`<!DOCTYPE html>`

`<html>`

`<head>`

`<title> Index </title>`

`<link href = "cocaine.jpg" rel = "icon"`

`</head>`

`<body>`

` Cocaine `

prgl:-

choice

La

Lu

La

Lu

Lu

LP

Lia

LP

CC

L/H

prgl:-

CB

LC

LC

LC

LC

anchor tag

`
dis UAAMA Z (a)`

`<h2>clj</h2>`

`</html>`

`proj1:- <html><body>`

` Aonir kan Z (a)
`

` Salman kan Z (a) Z br />`

` DARSHAN Z (a) Z br />`

` Lubeep Z (a) Z br />`

` QCQn Z br />`

`<p id = "a"> SAGAN KAN Z br />`

``

`<p> ----- </p>`

`</`

`</`

`</body>`

`</html>`

`proj1:- <html><body>`

` salman2 (a)`

` Lubeep Z (a) Z br />`

` img src = "salman1.jpg" />`

`</body>`

`</html>`

- pre tag
- 1) <pre> </pre> → used to preserve the whitespace
 - 2) <mark> </mark>
 - 3) <progress> </progress>
 - 4)
 - 5)

deprecated tags

the tags which is removed from HTML file

is known as deprecated tags

proj :- <!doctype html>

<html>

<head> </head>

<body>

<pre>

welcome to ISFDSR

welcome to ISFDSR

welcome to ISFDSR

welcome to ISFDSR

<pre>

we are learning <mark> </mark>

 ISFDSR

ISFDSR lessons

<i> ISFDSR </i>

<progress value="50" max="100"> 50% </progress>

2

(for)

① <pre>

② <sel>

③ <opt>

④ <file>

⑤ <opt>

⑥ <leg>

⑦ <par>

⑧ <dec>

⑨ <te>

</pre>

Attrib

⑩ <typ

⑪ now

⑫ val

⑬ place

⑭ so

⑮ now

⑯ red

⑰ did

⑲ co

⑳ ab

㉑ ob

㉒ ab

㉓ ob

㉔ ab

㉕ ob

㉖ ab

㉗ ob

㉘ ab

㉙ ob

㉚ ab

form tag

<form>

- 1) <input>
- 2) <select> </select>
- 3) <optgroup> </optgroup>
- 4) <fieldset> </fieldset>
- 5) <option> </option>
- 6) <legend> </legend>
- 7) <button> </button>
- 8) <datalist> </datalist>
- 9) <textareas> </textareas>
</form>

<input>

Attribute Parameter

- 1) type Text, password, Email, Radio, Checkbox, URL, file, reset, submit, number, date, button, color
- 2) name Any Name
- 3) value ——
- 4) placeholder ——
- 5) size Px, Pt, %
- 6) max-length Any Number
- 7) required
- 8) disabled
- 9) checked

<textareas> </textareas>

Attribute parameter

- 1) rows Px, Pt, %
- 2) cols ——

<select> </select>

Attribute Parameter

- 1) multiple

< forms & form >

Attribute parameter

1) Action URL, HTML, file name

2) Name Any name

3) Target _parent (parent), _blank

4) Method get (default), post

proj 1:-

< html >

< head >

< body >

< form >

< input >

password

< project >

< select >

< option >

< optgroup >

< option >

< input >

proj 1:- < bodies >

< form >

username: < input type = "text" >

password: < input type = "password" >

Email: < input type = "email" >

URL: < input type = "URL" >

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

< input type = "reset" value = "clear" >

< /form >

< /body >

proj 1:- <!DOCTYPE html>

< html >

< head > </head >

< bodies > < form >

< input type = "radio" name = "r1" checked > option 1

Select your course

< input type = "checkbox" name = "c1" > science

< input type = "checkbox" name = "c2" > java script

< input type = "checkbox" name = "c3" > python

< input type = "checkbox" name = "c4" > angular

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

< input type = "reset" value = "clear" >

< /form >

< /body >

< /html >

prel 1 - <html>

<head> </head>

<body>

<form>

username : <input type = "text" name = "u" >

password : <input type = "password" name = "p" >

<input type = "submit" name = "s" >

<select>

<options select your state</options>

<option> karnataka </options>

<option> Andhra </option>

<option> up </options>

<options> Bihar </options>

<option> Tamil nadu </option>

<option> Maharashtra </option>

</selects

<select multiple>

<optgroup label = "non-veg">

<option> shawarma </option>

<options> BBQ chicken </options>

<options> kebab </options>

<option> chicken 65 </options>

</optgroup>

<optgroup label = "veg">

<option> lemon rice </option>

<option> paneer </options>

<option> veg biryani </options>

<option> mushroom </option>

<option> French fries </options>

</optgroup>

</selects

<fieldset>

<legend align="right"> Select your Gender </legend>

<input type="radio" value="male"> male

<input type="radio" value="female"> female

<input type="radio" value="others"> others

</fieldset>

<button type="button"></button>

<textarea rows="10" placeholder="write your comment" cols="30"></textarea>

</form>

</body>

</html>

Ques :-

<body>

<form>

username:<input type="text" name="un" placeholder="Enter valid username" minlength="6"/>

password:<input type="password" name="pw" minlength="8" value="abc123" />

email:<input type="email" name="e">

URL:<input type="url" name="u">

dob:<input type="date" name="d">

upload your documents <input type="file" name="f"/>

<input type="number"/>

<input type="button" value="click" />

Select your Gender:

<input type="radio" name="m" disabled> male

<input type="radio" name="f" checked> female

<input type="radio" name="o" disabled> others

Header & Legacies

NOTE:- whenever we are sending the data to the server by using method is equal to get (method="get") then the data will be send as a part of URL

Q1:- <!DOCTYPE html>

<html>

<head> </head>

<body>

<form action="table.html" name="f1" method="post">

username: <input type="text" name="un">

password: <input type="password" name="pw" >

Enter your Favourite colour: <input type="text"

list="color"

<select id="color" > it is used to select the
options and & options suggestion

<option> green </option>

<option> Blue </option>

<option> purple </option>

<option> pink </option>

<option> yellow </option>

<option> cream </option>

<option> reddish white </option>

</select>

Details

<summary> vibrant look list </summary>

<p> --- </p>

</details>

<input type="submit" >

</form>

</body>

1a) & 2b)

abed > male

abcd > female

abed > other > b)

CSS(1) Document level style sheet:

Syntax:- <head>

<style type="text/css">

Tag names, Tag names {css property name: value; ...}

<1 styles

<1 head>

html comments:-

<!-- content -->

eg:- <!doctype html>

<html>

<head>

<styles>

p, a { color: red; background-color: cyan; }

span { color: green; }

<1 style>

<1 head>

<body>

<p> Pingi </p>

<p style="color: red"> Pingi </p>

<p> Jspidra </p>

 aspidra

 Range

<body>

<html>

- * whenever we come using style tag. then it becomes document level style sheet. the properties which are passed using document level style sheet will be applied to the body of html document (based on tag names).

- * Inline style
- * External style
- * If the properties

External

<head>

<link href="

rel="

<head>

<body>

<1 body>

</body>

</html>

Tag

{ CSS }

CSS

}

p

}

st

- * Inline style sheet always overwrites document level or nested as external style sheet
- * Inline style sheet overwrites only when the properties are same
- * If the properties are different it inherits the properties from document / external style sheet

External style sheet

index.html	front.html
<pre> <head> <link href="style.css" rel="stylesheet"> </head> <body> <p> Paragraph 1 </p> <p> Paragraph 2 </p> </body></pre>	<pre> <head> <link href="style.css" rel="stylesheet"> </head> <body> <p> Paragraph 1 </p> <p> Paragraph 2 </p> </body></pre>

index.html front.html

<pre> Tag name 1, Tag name 2 {css property 1: value; css property 2: value; : } p {color: red}</pre>	<p>3 types</p>
--------------------------------------------------------------------------------------------------------	----------------

- 1) Inline
- 2) Internal / Document
- 3) External

style.css

e.g. then if becomes
properties which
style sheet need //
for document

CSS background Properties

- ① background-color : color name, code
- ② background-image : URL ("image name") image has to be in the same folder
- ③ background-repeat : no-repeat, repeat-x, repeat-y
- ④ background-attachment : scroll (defalut), fixed
- ⑤ background-position : left, right, center
- ⑥ background-size : width, height (px, pt, %), cover
to give both height & width value → 100% took space

eg:- <!doctype html>

<html>

<head>

<title> background </title>

<styles>

body { background-color :rgb(123,55,78); }

background-image : url (bg.jpg) ; background-repeat : no-repeat

background-attachment : fixed ; background-position : right;

background-size : cover }.

</styles>

</head>

<body>

<a>

100px

is index

</pre>

</body>

</html>

CSS

Attribute

① list-style

<ul style

L

L

L

Border

1) Border

2) Border

3) Border

→ i) ① Border

ii) Bo

iii)

iv)

•) image has to be in the same folder

•, repeat-y

Fixed

px, pt, %, cover

px or value → loop in footer space

and -repeat: no repeat.

!-position: right;

css property to insert image in a bullet

Attribute

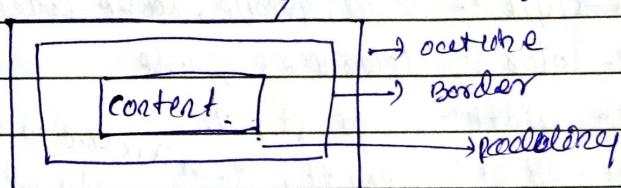
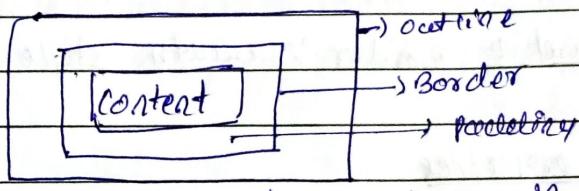
① `List-style-type: url ("image.png")`

`<ul style="list-style-type: url (o.png)">`

↳ is applied to li's

↳ li's

css box model



Border properties

- 1) `border-style` :- solid, groove, double, dotted, dashed, none
- 2) `border-color` :- color name, code
- 3) `border-width` :- px, pt, %

- | | |
|--------------------------------------|--------------------------------------|
| ↳ 1) <code>border-top-style</code> | 2) <code>border-top-color</code> |
| ii) <code>border-bottom-style</code> | ii) <code>border-bottom-color</code> |
| iii) <code>border-left-style</code> | iii) <code>border-left-color</code> |
| iv) <code>border-right-style</code> | iv) <code>border-right-color</code> |

- 3) for border width :-
- 1) border-top-width
- 2) border-bottom-width
- 3) border-left-width
- 4) border-right-width

Ex:- <body>

<p style="border-style: solid; border-color: blue;
border-width: 4px"> Isider </p>

</body>

Short hand :- border : width style color;

outline properties

- 1) outline-style :- solid, groove, double, dotted, dashed, none
- 2) outline-color :- colorname, code
- 3) outline-width :- px, pt, %.
- 4) outline-offset : px, pt, %. → it is used to give space between border and outline

The outer edge of an html element is known as outline

Shorthand :-

outline: width style color;

Ex:- outline: 2px solid green;

Ex :- <body>

<p style="border: green 2px solid solid"> border </p>

<p style="outline: red 2px solid solid"> outline 2px

<p style="border: green 2px solid solid; outline: red solid
3px"> both 2px

<p style="border: green solid 2px; outline: solid red
3px; outline-offset: 3px"> offset 2px

</body>

- 1) padding
- 2) padding
- 3) padding
- 4) padding
- 5) padding

the span padding

Ex:- <body>

<span

padding

- 1) margin
- 2) margin
- 3) margin
- 4) margin
- 5) margin

The

Ex:- <body>

<p

margin

<p style

margin

</body>

padding

- 1) padding :- px, pt, %.
- 2) padding-left :- px, pt, %.
- 3) padding-right :- px, pt, %.
- 4) padding-top :- -q-
- 5) padding-bottom :- -n-

the space between content and the border is known as padding

eg:- <body>

Margin

- 1) margin :- px, pt %.
- 2) margin-top :- -n-
- 3) margin-bottom :- -n-
- 4) margin-left :- -n-
- 5) margin-right :- -n-

The space between 2 brother is known as margin

eg:- <body>

<p style="border: 2px solid red; outline: 4px solid green; margin: 2px; display: inline-block; width: 100px; height: 50px; margin: auto;"></p>

<p style="border: 2px solid yellow; outline: 4px solid green; margin-top: 10px; margin-bottom: 10px; margin-left: 10px; margin-right: 10px; display: inline-block; width: 100px; height: 50px; margin: auto;"></p>

2<p>

: red solid

Solid green

Selectors

- 1) Element selector / simple selector
- 2) class selector
- 3) generic selector
- 4) Id selector
- 5) embeep selector

Element :- <head>

<style>

Syntax: Tag name { CSS property : value; ... }

e.g. :- p { color : red; ... }

<style>

</head>

<body>

<p> Dings </p>

<hr> Isorder </hr>

</body>

embeep :- <head>

<style>

Syntax: Tag name1, Tag name2, Tag name ... { CSS property : value; ... }

<style>

</head>

e.g. p, a, h1 { color : red; }

Class selector :- <head>

<style>

Syntax: Tag name . class name { CSS property : value; ... }

e.g. P. Dingce { color : red; ... }

P.2 { color : pink; ... }

<style>

</head>

<style>

< p class = "Dinaya" > aspiderx </p>

<p> Ispiderx </p>

 Pu spiderx X ^{"it will not affect to the anchor tag"}

it want to apply to cell itself ~~not~~ class because it's apply same tag.

generic selector

<head>

<style>

Syntax: <classname > {css property1: value1; ...; }

Ex: < Dinya { color: red; ...; }
P { color: pink; }

< style >

< head >

< body >

< p class = "Dinaya" > aspiderx </p>

<p> Ispiderx </p>

 Pu spiderx

< body >

one -> CSS property value: -> # Id selector

<head>

<style>

Syntax #<Name > {css property1: value1; ...; }

Ex: # Dinya { color: red; ...; }
P { color: pink; ...; }

< style >

Element & class selector

```

<link href="style.css" rel="stylesheet">
<link href="rel-19" rel="icon">

<style>
p {color: green}
p, a {color: red}
</style>
<head>
<body>
<p> Divided</p>
<p class="a"> Divided b </p>
<p> Spider </p>
<a href=""> Spider <a> b </a>
<span> Spider </span>
</body>

```

Attribute & id selector

```

<head>
<style>
.a {color: red}
#b {background-color: purple}
</style>
</head>
<body>

```

```
<p id="b"> Divided</p>
```

```
<p class="a"> Divided</p>
<p> Spider </p>
```

```
<a href="" class="a" id="b"> Spider <a> b </a>
```

```
<span class="a" id="b"> Spider </span>
```

```
<body>
```

Attribute selector

```
<head>
```

```
<style>
```

```
p, span, a {color: red}
```

```
</style>
```

```
<head>
```

Some we have

- 1) Font -
- 2) Font-style
- 3) Font-size
- 4) Font-weight
- 5) float -

TYPING
1) color
2) Text-decoration
3) Text-align
4) Text-indent
5) letter-spacing
6) line-height
7) word-spacing
8) text-align
9) color

- 1) body
- 2) p
- 3) p
- 4) p
- 5) p
- 6) p
- 7) p
- 8) p
- 9) p
- 10) p

CROSS

filesheet "S
"con"

Font - properties

- 1) font-style : italic, normal (Default)
- 2) font-weight : bold, lighter (Default)
- 3) font-family : Any font family name
- 4) font-size : px, pt
- 5) font-variant: small-caps, small-small

Text Properties

- 1) color : color code, name
- 2) text-align : left (Default), right, center, justify
- 3) text-transform : uppercase, lowercase, capitalize
- 4) text-decoration: overline, line-through, underline, none
- 5) letter-spacing : px, pt, %
- 6) line-height : - (px) px are mandatory
- 7) word-spacing : - (px) px
- 8) text-shadow : horizontal value vertical value ^{values} blur radius
(px, pt, -.) (px, pt, -.)
color

Text Font - propertiesHTML

- <p style="font-style: italic;"> my first heading </p>
- <p style="font-weight: bold;"> Diya </p>
- <p style="font-family: 'Myanmar'"> Diya </p>
- <h1 style="font-family: 'Brush Script MT'"> Diya </h1>
- <p style="font-size: 5px;"> Diya </p>
- <p style="font-variant: small-caps;"> Diya </p>
- <p> Diya </p>

2. ~~style~~ class

use
X21Icons

Text Properties

```

<p style="color: pink"> ISpider </p>
<p style="text-align: right"> ISpider </p>
<p style="text-transform: uppercase"> ISpider </p>
<p style="text-transform: capitalize"> ISpider </p>
    ↓ URL memory negotiate </p>
<p style="text-decoration: underline"> ISpider </p>
    ↓ p
        ↓ href=" " style="text-decoration: none">
            ISpider </p>
        ↓ p style="letter-spacing: 3px"> ISpider </p>
        ↓ p style="word-spacing: 10px"> ISpider get id
            ISpider </p>
        ↓ p style="line-height: 30px"> ISpider </p>
        ↓ p style="text-indent: 4px 3px 1px 0px">
            ISpider </p>
    ↓ body

```

10/7/19position

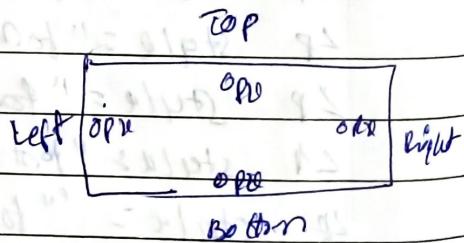
i) position: Relative, sticky, static (Default)

ii) Top: px, ft, %.

iii) Bottom: -(-)

iv) Left: -(-)

v) Right: -(-)



proj:- <head>

<title> Background Concepts </title>

<link href="style.css" rel="stylesheet">

<link href="site.1q" rel="icon">

<style>

p { position: relative; left: 20px; top: 50px; }

<style>

<body>

<p> ISpider </p>

proj:-

prg1:-

```

<html>
  <head>
    <link href="style.css" rel="icon">
  </head>
  <body>
    position: sticky; bottom: 0px;
  </body>
</html>

```

JSP code

```

<input type="text" placeholder="Search" />

```

</body>

Border Radius and Box shadow

1) Border-radius: / top-left corner top-right corner

① bottom-right corner bottom-left corner
②

a) Box-shadow: Hr value Vr value blur ratio spread ratio
color

↳ if you give one value it applicable for all

if you pass ~~float~~ value it applicable 2 parameter at a time

prg1:-

 & background-color: orange;

position: relative;

top: 100px;

border: 2px solid red;

border-radius: 50px;

padding: 10px;

box-shadow: 3px 2px 2px purple;

<style>

<body>

 Inside

<body>

Q1: <style>

```

p { background-color: orange; position: relative;
left: 200px; top: 100px; border: 4px solid red;
border-radius: 120px 0px;
padding: 0px; box-shadow: 0px 6px 3px purple;
height: 200px; width: 150px; }

```

Combinators

Combinators are the special type of selector which explains the relationship between other selectors.

- 1) Descendant selector (>spare) → div > p (only div contains p)
- 2) Child selector (>) → only direct child (E)
- 3) Adjacent sibling selector (+) → it selects the direct div + sibling
- 4) General sibling selector (~)

<html>

it selects only div's direct child

<head>

<style>

div > p { color: red; }

<style>

<head>

<body>

<div>

<p> Diagonal </p>

<p> Diagonal </p>

 <p> Leveled </p>

<div>

<div> our

<p> Is older

<p> experience

<p> respect

<div>

Head

Style

<div>

color

<head>

<div>

style

div

<style>

</head>

<body>

<div>

</div>

<div>

</div>

<div>

</div>

</div>

<div> Banega </div>

<p> Isoldore </p>

<p> orson </p>

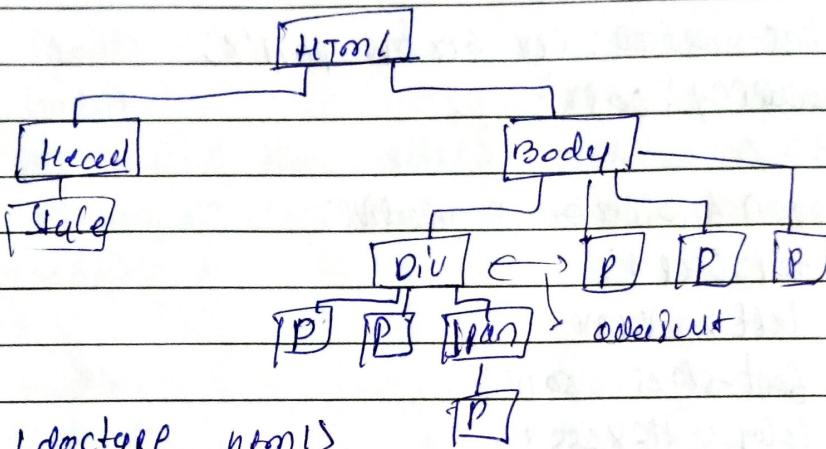
<p> suspiria </p>

</body>

; relative;

: #18 solid red;

px 3px purple

type of selector
between other(only div child +
enclosed in)+ select the direct
div &
only one
selected

<!doctype html>

<html>

<head>

<title> Booksgrouse Tongue </title>

<style>

div & p { color: red; }

</style>

</head>

<body>

</div>

<p> Banega </p>

<p> orson </p>

<div> & <div> </div> </div>

<p> suspiria </p>

</div>

<p> Isoldore </p>

<p> orson </p>

<p> suspiria </p>

proj:-

```

p { background-color: orange;
  width: 400px;
  height: 200px;
  position: relative;
  top: 10px;
  left: 10px;
  border-radius: 6px 6px 6px 6px / 6px 6px 6px 6px;
  border: 2px solid red;
  box-shadow: 6px 6px 6px 6px purple;
  padding: 0px;
}

```

```

position: relative;
top: 60px;
left: 180px;
font-size: 80px;
color: #E8C5F7;
< 9 styles >
< 1 header >
< body >
< p > < span > I love you! < span > < /p >

```

proj:-

< head >

< styles >

a: link

a: hover

a: visited

p: first

< /style >

< /head >

< body >

< a >

< a >

< /body >

< /a >

< /a >

< /body >

< /head >

< /style >

< /head >

< /body >

< /body >

< /head >

< /style >

< /head >

< /body >

< /body >

< /head >

< /style >

< /head >

< /body >

< /body >

< /head >

< /style >

< /head >

< /body >

< /body >

< /head >

< /style >

< /head >

< /body >

< /body >

syntax: Tagname : Pseudo class name

in anchor tag

we can use

1 : link

2 : Hover → we can use in all tag

3 : Active

4 : Visited

5 : Focus → Focus → Input tag

Pseudo Classes

pseudo classes are used to change the current state of HTML elements.

1. Zing

button

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

-

proj 1:-

<head>

<style>

a: link {text-decoration: none;}

a: hover {text-decoration: underline;}

a: visited {color: green;}

p: font-size: 16px; background-color: yellow; color: pink;
outline-color: red;

</style>

<head>

<body>

 Spiderman

 Liver
 Spiderman

</body>

proj 1:-

<head>

<style>

button {background-color: red; border: 3px solid black; border-radius: 12px; box-shadow: 4px 4px 4px orange; outline: none;}

11/3/19

button: hover {background-color: red; border: 3px solid yellow; border-radius: 12px; box-shadow: 4px 4px 4px blue; outline: none;}

button: active {background-color: green; border: 3px solid green; border-radius: 12px; box-shadow: 4px 4px 4px pink; outline: none;}

<style> </head>

<body>

<button> Click </button>

</body>

Opacity

It is used to increase or decrease the brightness of HTML elements and it takes the value from 0.1 - 0.9.
 , } highest opacity
 lowest }

proj:-

<head>

<style>

img { opacity: 0.4; }

img : hover { opacity: 0.9; }

</style>

<head>

<body>

</body>

Visibility

syntax: visibility: visible (default), hidden

Visibility property is used to hide the content

proj:-

<body>

<p style = "visibility: hidden;"> Ispider </p>

<p> Spider </p>

</body>

proj:-

<head>

<style>

div

border

</style>

<body>

<div>

</div>

</body>

</html>

check -> div

Decrease the brightness
and the value

Transform Properties

Span tag is a dummy tag which has no effect

prop:- 2 bodies

```
<body style="color: red"> SS & span style="color: green"> P
<span> id=xx </span>
</body>
```

Transform - properties.

- 1) Transformation: Rotate (value in deg) | Translate (x-axis value)
scale (width height) ↓
↓ use command (5, 3) (12px, 4px)

prop:- 1 header

1 style

```
div { background-color: red; height: 100px; width: 100px;
transform: rotate (45deg); }
```

1 styles

1 bodies

```
<div> is red </div>
```

</body>

hidden

the content

Transition

- 1) Transition : CSS property Time in sec
- 2) Transition-Delay : Time in sec
- 3) Transition-Duration : Time in sec

```
check - div { hover { background-color: purple; transform: rotate
(45deg) } }
```

pral:- 2 heads

<style>

```
div { background-color: green; height: 100px; width: 100px;
      transition: background-color 2s, height 2s, transform 2s;
      transition-duration: 3s; @transition-delay: 0s;
```

}

```
div:hover { height: 200px; background-color: purple;
            transform: rotate(180deg); }
```

<style>

Animation - properties

- 1) Animation-name: Any name → mandatory
- 2) Animation-Duration: Time In sec → mandatory
- 3) Animation-Delay:
- 4) Animation-Direction: Forward(Default), reverse
- 5) Animation-Iteration-count: Any number, Infinite

2 heads < styles

pral:- div { background-color: red; height: 100px; width: 100px;
 animation-name: dingi; animation-duration: 2s }

}

@keyframes dingi {

0% { background-color: orange; }

10% { - - - : red(251, 188, 12); }

20% { - - - : red(66, 133, 247); }

30% { - - - : red(234, 67, 13); }

40% { - - - : red(108, 56, 99); }

50% { - - - : green; }

60% { - - - : cyan; }

70% { - - - : maroon; }

80% { - - - : black; }

90% { - - - : purple; }

100% { - - - : yellow; }

}

pral:- 2 heads

<style>

```
div { b
```

height:

width:

animation:

Animation CSS

Assessment →

Animate.CSS

px; width: 100%; prg: - cheed;

<check>

<style>

div { background-color: red;

height: 100px;

width: 100px;

animation-name: dinya;

animation-duration: 20s

animation-direction: reverse

animation-delay: 5s

animation-iteration-count: 2

{}

@keyframes dinya

0% { background-color: orange; position: relative; left: 50px;

transform: rotate(0deg); }

10% { left: 100px; transform: rotate(5deg); }

20% { left: 150px; transform: rotate(10deg); }

30% { left: 200px; transform: rotate(15deg); }

40% { left: 250px; transform: rotate(20deg); }

50% { left: 300px; transform: rotate(25deg); }

60% { left: 350px; transform: rotate(30deg); }

70% { left: 400px; transform: rotate(35deg); }

80% { left: 450px; transform: rotate(40deg); }

90% { left: 500px; transform: rotate(45deg); }

100% { left: 550px; transform: rotate(50deg); }

{}

Father

&nbsp;+&nbsp;

class = "

Email :

place hol

&nbsp;

place hol

password

value

&nbsp;

place hol

DOB: +

&nbsp;-

value

&nbsp;

+&nbsp;

<&nbsp>

+&nbsp;

<&nbsp>

<&nbsp>

&nbsp;

<&nbsp>

<&nbsp>

<&nbsp>

<&nbsp>

&nbsp;

&nbsp;

&nbsp;

&nbsp;

&nbsp;

<&nbsp>

<&nbsp>

<&nbsp>

<&nbsp>

<&nbsp>

<&nbsp>

<&nbsp>

<&nbsp>

<&nbsp>

Special characters →

proj:- <!doctype html>

<bodies

&sp; +&nbsp; +&nbsp; +&nbsp; +&nbsp; +&nbsp; +&nbsp; +&nbsp;

deos

</bodes>

Headless in HTML

proj1:- <bodes>

<n1> ISpiders </n1>

<n2> ISpiders </n2>

<n3> ISpiders </n3>

<n4> ISpiders </n4>

<n5> ISpiders </n5>

<n6> ISpiders </n6>

</bodes>

~~pixabey.com~~

pixabay.com

Syncop form

<!doctype html>

<htan>

<head>

<ttitle> Index </ttitle>

<link href="conisr.jpg" rel="icon">

<link href="style.css" rel="stylesheet">

<style> </style>

<head>

<bodes>

<n1> Registration Form </n1> → pass login formame to
me direct log in page

<form action="" name="f1" method="post" target=_blank>

First Name:&nbsp;&nbsp;&nbsp;<input type="text"

class="inp1">

&nbsp;--> <input type="text" class="inp2">

&nbsp;

Father name: <input name="name" type="text" class="Pnp">
 &nbsp&nbsp mother name: --- <input type="text"
 class="inp">

 Email: --- <input type="email" class="inp"
 placeholder="Enter valid Email" style="border: 1px solid black; width: 200px; height: 30px; border-radius: 10px; padding: 5px; margin-bottom: 10px;"/>
 --- confirm Email: <input type="email" style="border: 1px solid black; width: 200px; height: 30px; border-radius: 10px; padding: 5px; margin-bottom: 10px;"/>
 password: --- <input type="password" placeholder="Enter
 valid password" class="inp">

 --- password: --- <input type="password"
 placeholder="Enter valid password" class="inp">

 DOB: --- <input type="date" class="inp">

 --- Address: <textarea placeholder="Enter
 valid Address" class="inp" style="width: 300px; height: 100px; border: 1px solid black; border-radius: 5px; padding: 5px; margin-bottom: 10px;"/>
 --- select your gender:
 <input type="radio" name="g" value="male" style="margin-right: 10px;"/>
 Male
 <input type="radio" name="g" value="female" style="margin-right: 10px;"/>
 Female
 <input type="radio" name="g" value="others" style="margin-right: 10px;"/>
 Others
 --- Select your course:
 <input type="checkbox" name="c" value="java" style="margin-right: 10px;"/>
 Java
 <input type="checkbox" name="c" value="python" style="margin-right: 10px;"/>
 Python
 <input type="checkbox" name="c" value="C" style="margin-right: 10px;"/>
 C
 <input type="checkbox" name="c" value="SQL" style="margin-right: 10px;"/>
 SQL
 --- Select your course:
 --- upload your document:
 --- <select style="width: 200px; height: 30px; border: 1px solid black; border-radius: 10px; padding: 5px; margin-bottom: 10px;"/>
 <option>A</option>
 <option>B</option>
 <option>C</option>
 <option>D</option>
 <option>E</option>

please
not login page

target=_blank
text

>"Pnp">


```

<input type="file"> <br><br/>
+<br> --- select your state;
+<br> --- <select>
    <option> A </option>
    <option> B </option>
    <option> C </option>
    <option> D </option>
    <option> E </option>
</select> <br><br/>

```

+
 --- <input type="checkbox" checked="">
 Accept All terms and conditions

+
 --- <input type="submit" value="Sign Up" class="btn-1">

CSS property

body {
 background - image: url(desert.jpg);
 background - repeat: no-repeat;
 background - size: cover;
 }

h1 {
 text-align: center;
 color: black;
 }

form {
 background: rgba(67, 230, 118, 0.4);
 width: 553px;
 border: 5px solid red;
 padding: 30px;
 border-radius: 30px;
 position: relative;
 top: 20px;
 }

left: 220px
 box-shadow:
 }
 .input {
 background-color: #f0f0f0;
 border: 1px solid #ccc;
 padding: 5px;
 margin-bottom: 10px;
 }
 .input: focus {
 border: 1px solid #007bff;
 outline: none;
 }
 .button {
 background-color: #007bff;
 border: none;
 color: white;
 padding: 10px 20px;
 font-size: 16px;
 cursor: pointer;
 }

left: 220px;

box-shadow: 1px 4px 3px rgba(67, 230, 118);
}

.pop {

background: rgba(67, 230, 118, 0.1);

border: 1px solid white;

}

.inf:focus {

outline: 2px solid purple;

}

.bfch {

background: rgba(67, 230, 118);

padding: 5px;

border: 2px solid white;

border-radius: 20px;

& outline: none;

.a { background-image: url(desert.jpg);

background-repeat: no-repeat;

background-size: cover;

}

bx
gallery

login form

<!DOCTYPE html>

gallary

```
<!DOCTYPE html>
<html>
<head>
<title> gallary </title>
<link href = "gallary.ipynb" rel = "icon">
<style>
div img { height: 400px;
            width: 340px;
            border: 8px solid orange;
            outline: 1px solid lime;
            margin-top: 2px;
            margin-left: 10px;
            margin-right: 10px;
            margin-bottom: 10px;
            opacity: 0.4;
        }
    
```

<!DOCTYPE html>

`div { border: 8px solid purple; }`

`div img: hover { opacity: 0.9; }`

</style>

</head>

<body>

`<h1 align="center"> Gallery </h1>`

</div>

` ![---](1.jpg)`

` ![---](2.jpg)`

` ![---](3.jpg)`

` ![---](4.jpg)`

``

↓
horizontal

</div>

``

``

pass target attribute to open in different
tag

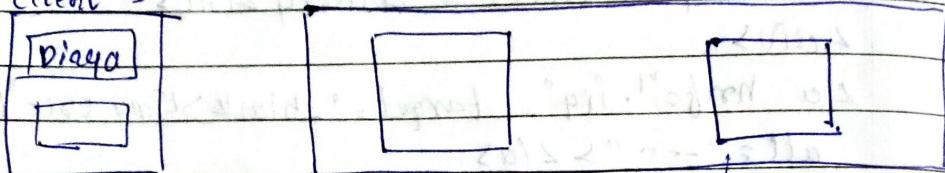
</body>

</html>

Java Script

- * Java script is a client side scripting language, which is used to make web pages dynamic.
- * Java script is a loosely typed scripting language.

client s:



Display

- 1) document
- 2) document
- 3) console
- 4) innerHTML

Type of
internal
<Head>

Following are the frameworks / frameworks of JS

1. Node JS → it's used to write business logic
2. Express JS → it's used to connection purpose
3. Angular JS (Angular) → is used to create single page application
4. React JS
5. Vue JS

Data types in Java script

1) primitive Data type

1) Number ex: var a = 20; var a = 55.88;

2) String ex: var a = "Dinaya"; var a = 'Dinaya';

3) Boolean ex: var a = true; var a = false

4) NULL ex:- var a = null;

5) Undefined ex:- var a;

2) Extension

2) Non primitive Data type

1) object()

2) Array()

3) Date()

Display statement or Print statement in Java script

string (e.g.,

dynamic
c type

- 1) document.write () → not mandatory
- 2) document.writeln ()
- 3) console.log ()
- 4) innerHTML

Type of JavaScript

i) Internal JavaScript

<Head>

<script>

// JS code

</script>

</head>

<body>

<script>

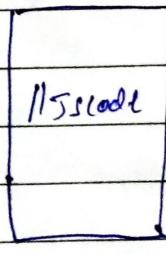
// JS code

</script>

</body>

(ctrl+shift+i)

ii) External JavaScript



<script src = "a1.js">

</script>

a1.js

prg1:- <head>

<titles> JavaScript </titles>

<script>

document.write("Spiderz" + "4brs");

document.write("aspiders")

</scripts>

</head>

a) let

syntax :-

Ex :-

b) const

syntax :-

Const :-

prg1:- <head>

<script>

var a = 20;

0x

document.write(a + "2brs");

20

var b = a + c;

20.45

document.write(b + "2brs");

Diaq1

var c = "Diaq1";

Diaq1

document.write(c + "2brs");

true

var d = 'Diaq1';

null

document.write(d + "2brs");

undefined

var e = true;

var a = 20;

document.write(e + "2brs");

js.htm: 21

var f = null;

if (true)

document.write(f + "2brs");

{

var g;

var a =

document.write(g + "2brs");

do

document.write(h + "2brs");

while

</script>

FOUND ! - JS

JS ! - JS

keyword to declare variable in java script

i) var

assessment :-

System :- var variableName = value;

prg1:- <

Ex:- var a = 20

let

if (

){

a) let

syntax :- let variable-name = value

ex :- let a = "Dinga"

b) const

syntax :- const variable-name = value;

ex :- const a = 'Dinga';

ECMAS → European Computer Manufacturers Association

JavaScript = q

Ei → ECMAScript

o/p

20

20.45

1) var

var u = 20; (declaration)

u = 40; (initialization)

var u = "Dinga"; (declaration)

undefined

2) let

let a = 20; ✓

u = 40; ✓

let a = Dinga; X

js.html: 21

3) const

const a = 20; ✓

u = 40; X

const a = "Dinga"; X

var u = 20;

if (true)

{

var a = 'Spiderman';

d.w(a);

d.w(a)

TQWU :- Spiderman, 20

JS :- Spiderman, Spiderman

let u = 20;

if (true);

{

let a = "Dinga";

d.w(a);

d.w(a)

TQWU :- Dinga, 20

JS :- Dinga, 20

const a = 20;

if (true);

{

const a = "Dinga";

d.w(a);

>

const a = "Dinga";

d.w(a);

d.w(a);

Java script

Assignment :-

<Horde>

print :- <script>

let a = 20;

if (false)

{

var a = "Dinga";

}

document.write(`at\${nr}`);

2
document.write(`at\${nr}`);

4 script

<head>

proj:- <head> <script>

let a;

a = "Dynam";

d.w(`at\${nr}`);

var b;

b = "Dynam";

d.w(`at\${nr}`);

const c;

c = "JavaScript";

d.w(`c\${c}`);

</script>

</head>

<head> <script>

proj:- var a=20;

d.w(`at\${nr}`);

var b=a; d.w(`b\${b}`);

a = "Dynamic";

d.w(`at\${nr}`);

document.write(`at\${nr}`);

2/20/2023 Head

var :-

* var

variable

* whenever

word +

* in case

on com

* whenever

return

function

let :-

* let is

variable

* whenever

let is

the be

* in co

and

re-de

*

const

* const

variable

* whenever

using

to

* in

use

* re-

var:-

- * Var is a keyword which is used to declare variables in Java Script
- * whenever we are declaring the by using var keyword then the scope is not limited to the block.
- * in case var declaration, re-declaration & initialization are possible
- * whenever we are declaring by using var keyword within function then the scope is limited to the function that means var is a function scope

let:-

- * let is a keyword which is used to declare the variables in Java Script.
- * whenever we are declaring variables by using let keyword then the scope is not limited to the block.
- * in case let keyword we can declare the variables and we can reinitialize the variables but re-declaration is not possible
- *

const:-

- * const is a keyword which is used to declare the variables in Java Script.
- * whenever we are declaring the variables by using const keyword then the scope is limited to the block.
- * in case of const only declaration is possible and both re-declaration, ~~initialization~~ not possible re-declaration, initialization is not possible.

prg1:-

```
<!DOCTYPE html>
<html>
<head>
<title>JAVASCRIPT</title>
</head>
<body>
<script src="app.js"></script>
</body>
```

prg2:-

```
document.write("Hellooo");
```

Functions

- * Function is a ~~block~~ block of code which is used to perform a particular task.
- * by using functions we can achieve code re-useability and code modularity
- * ~~functions~~ are the different type of function

1) Function with Argument & return Type

a) function f1() { } No return

3) Function with No Argument & with Return Type

4) function f1() { } No return type

* Function is a keyword which is used to declare the function in Java Script

Syntax:- function function-name (Arguments) → optional

{

↳ code to be executed ;

}

ex 1) function
{

return

;

FWith

3) function

{

return

;

No Argu

func

syntax:- fo

d

prg1:-

func

3

4

add

ex 1) function m1 (a,b)
 {
 }

return ab

}

F with Argument with return type

2) function m2 (a,b)
 {
 }

}

with Argu no return

3) function m3 ()
 {
 }

return "Dinakar";

}

no Argu with return

4) function m4 ()
 {
 }

}

using true naming

in id

de re-useable

Anonymous function

function with no name

function

syntax:- function ()
 {
 }

{ code to be executed }

}

declaration function

Type

in Type

return type

Var a = $\begin{cases} \text{function ()} \\ \text{---} \\ \text{---} \end{cases}$ } initialization

to declare

a () \rightarrow calling function

mai:- function odd()
 {

d.w (20+20+"20+20")

}

odd():

Function m1()

{

Var a=20;

return a;

{

Var b=m1();

d.w("b+" + b);

prg:- same as html prg

prg:- Function m1()

{

Var i=50;

for(Var i=0; i<=10; i++)

{

d.w(i+" ");

}

d.w(" "+ " "); → 11 0 r

{

m1();

d.w(" "+ " "); → error → i is not defined

Q2:- same has html prg

prg:- d.w(" "+ " "); → undefined

Var i=20; Var a;

Hoistin

* in creation

function

in creation

initialization

prg:- function

{

Var

d.w

func

{

v

m2

d.w

{

m1

prg2:-

same

prg1:- function

{

d.cel

d.u

{

conic

var

{

bu

Hoisting

- * In creation stage all the variables and function declarations are set to the memory
- * In creation stage all the variables will be initialized with undefined

prob:- function m1()

{

01 20 40 20

var a=20;

d.ow (at " ");

function m2()

{

function & block are

var a=40;

different

d.ow (at " ");

{

b) var is allotted to function & not block

m2();

d.ow (at " ");

{

m1();

prob2:- come here before prob1

OP

name is: 3+

age is: Disney
Hello world.

prob1:- function m1(name, age)

{

d.ow ("name is: " + name + " & " + age);

d.ow ("age is: " + age + " & " + name);

{

m1(3, "Disney")

var func = function ()

{

d.ow ("Hello World" + " " + age);

{

func(); } d.ow (func() + " "); -> it will give undefined

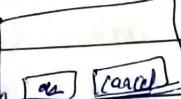
pop-up boxes

1) Alert ()

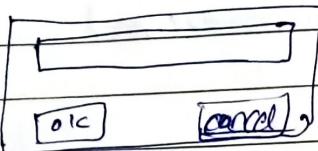


→ it's accept any type
prints any type

2) Confirm ()



3) Prompt ()



→ only data type is
considered as string
only

different

= → c

prbl: <body>

prbl: function m1()

{

 alert ("Hello");

 var a = confirm ("press ok or cancel");

 d.w(a);

 var a = prompt ("enter Any value: ")

 d.w(a);

}

m1();

< /scr

4 body

prbl: same has html

prbl: function m1()

{

 var a = prompt ("enter value of A")

 var b = prompt ("enter value of B")

 alert (a+b);

}

m1();

= op

= =

====

Type of

of the after

Difference between =, == and ===

= → assign

here it's
as string
ity

```
<html><body><script>  
var a = 20  
var b = "20"  
console.log(a);  
console.log(b);  
if (a == b)  
{  
    console.log("True")  
}
```

018 → 20
20
false

else

{

console.log("False")

}

else

need to be object

undefined is not
string is a object

</script>

</body>

= operator is used to assign the value

== operator is used to check the condition
only based on values

=== operator is used to check the condition
based on values as well as data types

Type of operator

Type of operator is used to check the data type
of the other variable.

```

prog1:- <body>
        <script>
var a = 20;
var b = "20";
var c = true;
var d;
var e = null;
console.log(typeof a);
console.log(typeof b);
console.log(typeof c);
console.log(typeof d);
console.log(typeof e);
</script>
</body>
    
```

or number
string
boolean
undefined
object

```

prog1:- <body>
        <script>
var a = 20;
var b = "20";
var c = true;
var d;
var e = null;
console.log(typeof a);
console.log(typeof b);
console.log(typeof c);
console.log(typeof d);
console.log(typeof e);
</script>
</body>
    
```

NaN → not a number

```

prog1:- <body>
        <script>
var a = Number(prompt("enter the value of A"));
var b = Number(prompt("enter the value of B"));
var c = alert(a+b)
</script>
</body>
    
```

isNaN function - checks whether the argument is not a number or not.

prog1:- <body>	isNaN(3);	false
<script>	console.log(isNaN("20"));	true
	console.log(isNaN("durga"));	false
	console.log(isNaN(true));	isNaN(c)

```

prog1:- <body>
        <script>
var a = 20;
var b = "20";
var c = true;
var d = NaN;
    
```

passing attributes inside javascript code

number
true
false
defined
object

pre1 :- <body> (current)

<script> → error

document.write("<p style='color:red'>Hello</p>")

</script>

</body>

pre2 :- <body>

<script>

document.write("Hello" + "" + "World")

document.write ("World")

</script>

</body>

a number

Event Handlers

Input Event

1) onsubmit()

Button Event

1) onclick()

2) ondblclick()

dbl -> double click

NOTE:- onsubmit() event should be inserted only
once if not
possible form trap bcz we are sending whole
form data to the server

false

true

false

pre3 :- <head>

<script>

function m1()

{

function m2() alert ("Hello");

}

</script>

</head>

How to

<body>

<button onclick="m1()"> click </button>

<button ondblclick="m1()"> click </button>

<form onsubmit="m1()">

<input type="scroll" />

</form>

</body>

onclick & ondblclick

can work in form tag

also

res = docu

prbl:-

<head>

<script>

function val1()

{

var res = confirm("press ok or cancel");

if (res == true)

{

return true;

}

BOM → browser object model

else

DOM → Document object model

{

return false;

}

</script>

</head>

</body>

<form action="pl.html" onsubmit="return val1();">

<input type="submit" />

</form>

</body>

BOM → browser object model

DOM → document object model

prbl:- <body>

<p> welcome </p>

</body>

How to call functions in console window

`res = document.querySelector('p')` → If it's not present return null
`<p style="background-color: red; color: green; font-family: arial, sans-serif; font-size: 50px;">background-color: red</p>`

`res.style.color = "black"; res.style.backgroundColor = "red"; res.style.fontSize = "50px"; res.style.fontFamily = "algerian"; res.style.fontStyle = "italic";`

`"background-color: red"`

`-----`

`-----`

`res = document.querySelector('p')`

ondblclick
in form tag

object model
object model

return value ~~of~~ ^{of}

base
node1

feature of node1

Applying CSS properties through JS

Inside console + copy the code into .js file

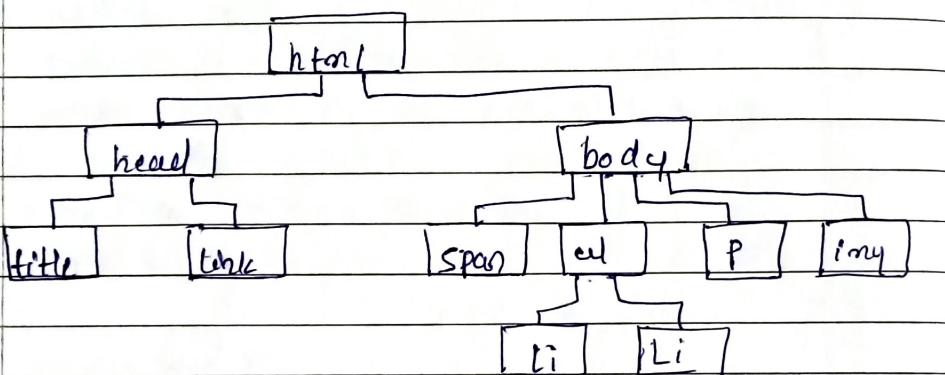
```
res = document.querySelector('p')
```

```
res.style.cssText = "background-color: red;  
color: pink; font-family: cursive; font-size: 50px";
```

different

- 1) document.querySelector
- 2) document.getElementById
- 3) document.getElementsByClassName
- 4) document.getElementsByName
- 5) document.querySelectorAll

DOM → Document object model



<HTML>

<head>

<titles & titles>

<links>

<nth child selector> -

<body>

res = document.querySelector('ul')

li : nth-child(3)

↳ selects 3rd child from ul

List

 Javascript

res.style.color = "red"

<p> <p>

<body>

<html>

res = document.

HTML collecti

res[2].style

using for

res = document

HTML collecti

for (var i = 0;

i < res.length;

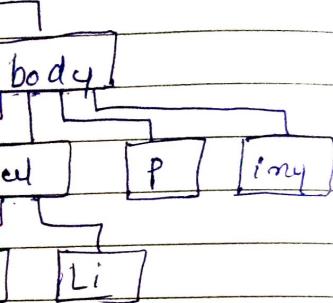
i++)

res[i].style

3

graph TD
ode into .js file
('P')
color:red;
margin; font-size: 50px;

object model



different methods of DOM

- 1) document.querySelector ('selector name')
- 2) document.getElementById ('ID name')
- 3) document.getElementsByClassName ('classname')
- 4) document.getElementsByTagName ('tag name')
- 5) document.querySelectorAll ('selector name')

1) sq:- res = document.querySelector ('#2')
variable classname/tag
<div id="2"> Spider </div>
res.style.borderColor = 4px solid yellow

querySelector returns the first selector or first tag which matches the name. If the element/selector is not present then it returns "null"

2) by tagname

res = document.getElementsByTagName ('span')

HTML collection (+) [span, span, span, span]

res[2].style.color = "yellow" → applies only to the 3rd index
yellow

using for loop

res = document.getElementsByTagName ('span')

HTML collection (q)

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

i

res[i].style.color = "red"

3

`document.getElementById()` returns all the elements as HTML collections. If the element is not present then it returns empty HTML collection.

3) using className

```
res = document.getElementsByClassName('b')
```

HTML []

`length: 0`

```
res = document.getElementsByClassName('a')
```

HTML collection [3] [p.a, span.a, div.a]

`res[1].style.color = "purple"`

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

{

`res[i].style.border = "5px solid red"`

}

`document.getElementsByClassName()` returns all the classes as HTML collection. If the class is not present then it returns empty HTML collection.

4) document.getElementById()

```
res = document.getElementById('z')
```

~~for multiple
parent
for many~~

`<div id="z"></div>` It returns 1 div

`res.style.cssText = "background-color: yellow;
border-radius: 30px; font-family: arial"`

`document.getElementById()` returns first `id` element from the HTML document. If the `id` is not present then it returns null.

to collect the
content in
the collection

`res = document.getElementById('b')`
null.

(5)

`document.querySelectorAll()`

`res = document.querySelectorAll('p')`
NodeList(5) [p, p.a, p, p, p]
`length = 5`
`res[2].style.cssText = "border: 3px solid red;
border-radius: 20px; box-shadow: 2px 2px 2px purple;"`

Array in JavaScript

19/7/19

In JavaScript array is a collection of heterogeneous elements

- * There is no fixed size in array
- * In JavaScript anything enclosed in between square brackets is known as array

Syntax:- `var a = [value1, value2, ...];`

`var a = ['Dinga', 22, 43.98, true, null];`

to find the length of array \Rightarrow `a.length`

`d.w(a[1])` \rightarrow 22

to reinitialize \rightarrow `a[i] = value;`

`a[0] = g` (creates only works only for string values)

`var a = ["Dinga", "Dingi", 756, 56.98, true, null, undefined].`

`d.w(a);`

`d.w(a.length);`

`d.w(a[0])`: \rightarrow Dinga, its index position [1] and character is position [2]

`d.w(a[1][2])`: n

`d.w(a[2][2])`: undefined it's works only string

Different types of methods in Array

- 1) push (arry: ---)
- 2) pop ()
- 3) unshift (arry: ---)
- 4) shift ()
- 5) reverse ()
- 6) join ()
- 7) slice (arry1, arry2)
- 8) indexOf (arry1)
- 9) lastIndexOf (arry1)
- 10) split (arry1, arry2, (arry3, ---))
- 11) concat (arry1, arry2, ---);

a [Dinga | Dingi | nicks | mone] ssp | acc]

var b = a. push ("nbsp");
dow (b);

1) push() method always inserts the elements at the end

* It always refers the updated array length

2) unshift()

* unshift() method always inserts the element at the start

* It always refers the updated array length
a.unshift ("utay");

* Index also get shifted accordingly

Viray	Dinga	Dingi	nicks
0	1	2	3

3) pop():-

a.

* pop() removes the end

* It returns

4) shift()

a. sh

* shift() removes the start

* It removes

5) reverse()

a.

* never

never

* It never

6) join()

join

* joins

7) slice()

slic

arry1

va

array

3) pop () :-

a. pop () ;

* pop () method always removes the array element at the end.

* It returns the removed elements

4) shift () ;

a. shift () ;

* shift () method always removes array element at the starting position

* It returns the removed element and indexes get shifted accordingly

5) reverse () :-

a. reverse () ;

reverse () method is used to reverse the arrays

6) join ()

a. join ('*')

join () method is used to join the array element using any characters

7) slice (arr1, arr2)

it also works in arry

slice (arr1 , arr2)

starting index

last index value

value

a. slice (1,3);

a[Dinga, Dingi', Nick, moco, isP]

slice () method is used to cut the array where arr1 is → starting index value. and arr2 → last index value.

8) concat ()

b	JS P	or S P
---	------	--------

var c = a.concat (b, ["Dinga", "Dingi"]);

[a[], b[], Dinga, Dingi]

9) index of ()

var b = a. indexof ("Dinga");
→ appearance

It always returns the first occurrences of search element.

a. indexof ("Dinga", 1) → starting index

for the index of search element from the starting index position.

- * If the element is present in an array then it returns the respective index value.
- * It always returns first occurrences index value
- * if the element is not present in an array, then '-1'.

(10) last index of ()

It returns the last occurrences index value
If the element is not present it returns '-1'

sq:- var
d.c
d.c
var
d

① que

2) que

3) get

4) get

5) get

sq:-
 var a = ["Dinga", "Diagi", "nick", "Diaga", "manu"];
 d.w(a + "Lbr")
 d.w(a.length)
 var b = a.push("Kanega");
 d.w(b + "Lbr")

"Diagi"];

q1]

① query selector() → we can select tag, Id, class anything we can select → first occurrence

- 2) query selectorAll() → same as queryselector
- 3) getElementById() → select only Id's
- 4) getElementByClassName → select only class name
- 5) get Element by Tagname() → select only tagname

in an array then
value.

PS index value
in an array,

index value
it returns -1

11) Splice

splice (arr1, arr2, (crys, ...)) arrn → 2 arguments

index	number of elements to be added	new elements
value	elements to be added	removed

Ex:- [Danya | Dinyi | Sheela | Leela | JSP]
 1 2 3 4

- * splice method is used to add or remove the elements from the array based on index value
- * it already's returns updated removed elements from the array

Ex:- splice(2, 0, 'Danya')

[Danya | Danya | Dinyi | Sheela | Leela | JSP]
 1 2 3 4 5

Ex:- splice(2, 2, 'Danya');

[Danya | Dinyi | Danya | JSP]
 0 1 2 3

Prbl:- 1. bodies

2. script

Var a = ["Danya", "Dnyi", "Sheela", "Leela", "JSP"];
 declaration, write (at "`
`");

Var b = a.splice(2, 0, "Danya", "Dnyi")
 d.w (a + "`
`");
 d.w (b + "`
`");

Var c = a.splice(3, 2, 'Leela');
 d.w (a + "`
`");
 d.w (c + "`
`");

0 → Danya, Dnyi

Dnyi, Dnyi

Dnyi, Dnyi,

Dnyi, Sheela

Danya, Dnyi

String class

methods

1) toUpper case()

2) toLowerCase()

3) repeat()

4) concat (arr1, -)

5) substr / arr1

6) slice (arr1, -)

7) clearArr (arr1)

8) clear / arr1

9) substrArr (arr1)

10) concatArr (arr1, -)

Prbl:- 1. bodies

2. script

Var a = "

d.w (a + "

a.concat(

d.w (a + "

01 -> Danya, Deneji, sheela, leela, ISP

Danya, Deneji, honya, Reenyi, sheela, Leela, ISP

Danya, Deneji, honya, Leela, Leela, ISP
Deneji, sheela

Danya, Deneji, Reenyi, ISP.

we do

s value

return

String class

methods

- 1) toUpperCase()
- 2) toLowerCase()
- 3) repeat()
- 4) concat(`any1, -- /any2`)
- 5) substring(`any1, any2`)
- 6) slice(`any1, any2`)
- 7) charAt(`any1`)
- 8) charCodeAt(`any1`)
- 9) substr(`any1, any2`) no of chars
- 10) string.fromCharCode(`any1`)
- 11) indexOf(`any1, any2`)
- 12) lastIndexOf(`any1`)
- 13) startWith(`any1`)
- 14) endWith(`any1`)
- 15) include(`any1`)
- 16) split(`any1`)

parallel bodies

< script >

var a = "JSplider Aspideros";

d.w(at "`
/S`") uppercase

d.w(a.toUpperCase() + "`
/`)

d.w(a.toLowerCase() + "`
/`)

d.w(a.repeat(3) + "`
/`)

d.w(a.concat("Danya") + "`
/`)

d.w(a.substring(2, 6) + "`
/`)

d.w(a.substring(3, 7) + "`
/`)

d.w(a.slice(-13, 7) + "`
/`)

d.w(a.charAt(3) + "`
/`)

d.w(a.charCodeAt(0) + "`
/`)

d.w(string.fromCharCode(65) + "`
/`)

ISP?

d-cw (a. indexOf ("r") + "Lbs")
 d-cw (a. indexOf ("s") + "Lbs")
 d-cw (a. indexOf ("r", 8) + "Lbs")
 d-cw (a. lastIndexOf ("D") + "Lbs")
 d-cw (a. lastIndexOf ("i") + "Lbs")
 d-cw (a. startswith ("r") + "Lbs")
 d-cw (a. ends with ("D") + "Lbs")
 d-cw (a. startwith ("J") + "Lbs")
 d-cw (a. ends with ("S") + "Lbs")
 d-cw (a. includes ("r") + "Lbs")
 d-cw (a. includes ("pi") + "Lbs")
 d-cw (a. includes ("pi") + "Lbs")
 d-cw (a. split ("s") + "Lbs") J, Rider, -a, rider
 d-cw (a. split ("i") + "Lbs")
 L script
 L Lbs

Date classmethods

- 1) get Full Year () var Date = new Date ()
- 2) getMonth ()
- 3) get Day ()
- 4) get Hours ()
- 5) getMinutes ()
- 6) getSeconds ()
- 7) get millisecond ()
- 8) getDate ()

prot: Lbs

L script

```

var dt = new Date ();
d-cw (dt + "Lbs")
d-cw (dt. getFullYear () + "Lbs")
  
```

```

    d.o() (dt. getMonth() + "L b2s")
    d.W() (dt. getDay() + "L b2s")
    d.o() (dt. getDate() + "L b2s")
    d.m() (dt. getHours() + "L b2s")
    d.w() (dt. getMinutes() + "L b2s")
    d.w() (dt. getSeconds() + "L b2s")
    </script>
    <body>

```

- 1) write a java script program ~~any~~ to display lowercase & uppercase alphabets along with their ASCII value.
- 2) write a java script prg to check whether the element display present in an array or not
if true it return true if not present add to the array

28/7/18

JS objects

Implicit

```

var a=20; number
var a='Disha'; string
var a=true; Boolean
var a=null; null
var a; undefined
var a=[ 'Disha', 25];

```

Explicit

```

var a=new Number(20);
var a=new String('Disha');
var a=new Boolean(true);
var a=new Array('Disha',25);

```

- * objects are collection of related variables and functions which are represented represented as key ~~and~~ value pairs.
- * in Java script we can create object in 3 ways
 - 1) using new operator
 - 2) using literals

3) using constructor functions

- 1) for in
- if for off
- 3) for each

1) using new operator

```
Syntax: var student = new object();
student.name = "Dinaya"
student.usn = 256;
student.gender = "male";
d.co (student.name);
```

JS on -> Java script

object rotation

2) objects using literal

Syntax: var variable_name = {key1: value1, key2: value2, ...}

Ex: var student = { name: 'Dinaya', Age: 25 }

d.co (student)

d.co (student.name);

student.gender = "male"

delete student.age

3) using

syntax

1

var

name

parrt 1: Java script using new operator

Var student = new object();

console.log(student);

d.co (student + "Lbgs");

student.name = "Dinaya";

student.age = 25;

student.gender = "male";

d.co (student.name + "Lbgs");

d.co (student.age + "Lbgs");

delete student.age

d.co (student.age + "Lbgs");

parrt 2: for

{

q

mfa

"re

1) for in

if for off

3) for screen

using constructor

```

var student = { name: "Dinga", age: 25, gender: "male",
    arr: [2, 3] }

d.w (student + " is br>");

d.w (student.name + " is br>");

d.w (student.gender + " is br>");

delete student.gender;

d.w (student.gender + " is br>");

d.w (student.arr + " is br>");
```

ES6 → Java Script

my2:

HTML

<Head>

<script src="d.js">

</script>

</Head>

: value1, key2: value2, ...

, Age: 25 }

3) using Constructor Function

```

Ex:- function student (name, age, gender)
{
    this.age = age; this.name = name; this.gender = gender;
}

var s1 = new student ("Dinga", 25, 'male');
var s2 = new student ("Dipali", 22, 'Female');
```

my1:- function student (name, age, gender)

```

{
    this.age = age;
    this.gender = gender;
    this.name = name;
    this.describe = function ()
    {
```

```

        return "Name is :" + this.name + " age is :" + this.age +
            "gender is :" + this.gender;
    }
}
```

2

3

```
var s1 = new student ("Dinu", 26, "male");
```

```
d.w ( s1 + "  
");
```

```
d.w ( s1.name + "  
");
```

```
var s2 = new student ("Dinu", 23, "Female");
```

```
d.w ( s2 + "  
");
```

```
d.w ( s2.name + "  
");
```

```
d.w ( s2.details() + "  
");
```

```
d.w ( s2.details() + "  
");
```

proj1:- HTML same as previous

For In loop

Syntax: for (variable_name in objectname)

{

// Is code

}

Ex: for (x in student)

{

==

}

proj1:- function student (name, age, gender)

{

stu.age = age;

stu.s.gender = gender;

stu.name = name;

}

var s1 = new student ("Dinu", 26, "male");

var a;

var details = "";

for (a in s1)

{

e'li

$$\text{details} = \text{details} + s[\alpha] + " "$$

$$/\! d\cdot w (\text{details} + " \angle b\alpha s")$$

$$d\cdot w (\alpha + " \angle b\alpha s")$$

?

$$d\cdot w (\text{details} + " \angle b\alpha s")$$

prg1: var student = { name: "Dinya", age: 26, gender: male
marks: 50 }

var a

var details = " ";

$$\text{for } (\alpha \text{ in student})$$

?

$$\text{details} = \text{details} + \text{student}[\alpha] + " "$$

$$d\cdot w (\text{details} + " \angle b\alpha s")$$

?

$$d\cdot w (\text{details} + " \angle b\alpha s")$$

e'li;

Regular expression

Regular expression is an object which is used to match the patterns.

- 1) var a = new RegExp ("pattern", modifiers)
 - 2) var a = /pattern/ modifiers
 - i → case insensitive
 - g → case sensitive
 - g → global
- patterns
- 1) [A-Z]
 - 2) [a-zA-Z]
 - 3) [0-9]
 - 4) (A-Z a-z)
 - 5) [A-Z] [0-9]
 - 6) [a-zA-Z 0-9]
 - 7) [a-zA-Z A-Z 0-9]
- + → 1 or more occurrence
 * → 0 or more occurrence
 # → Starts with uppercase A-Z
 ^ → start of the regulation
 and

^ negation

- 1) [^A-Z]
 - 2) [^a-zA-Z]
 - 3) [^0-9]
 - 4) [^A-Z a-z]
 - 5) [^A-Z 0-9]
 - 6) [^a-zA-Z 0-9]
 - 7) [^a-zA-Z A-Z 0-9]
- ^ start
- 1) ^[A-Z]
 - 2) ^[a-zA-Z]
 - 3) ^[0-9]
 - 4) ^[A-Z a-z]
 - 5) ^[A-Z 0-9]
 - 6) ^[a-zA-Z 0-9]
 - 7) ^[a-zA-Z A-Z 0-9]

proj: chord's
 <script>
 var a =
 var req
 var n
 d.(o) (n
 <script>
 LI hood
 <body>

code

obj:-<body>
 <form>
 user area
 Lib

* 0 or more

- 1) *[A-Z]
 - 2) *[a-zA-Z]
 - 3) *[0-9]
 - 4) *[A-Z a-z]
 - 5) *[A-Z 0-9]
 - 6) *[a-zA-Z 0-9]
 - 7) *[a-zA-Z A-Z 0-9]
- +
- + 1 or more
- 1) +[A-Z]
 - 2) +[a-zA-Z]
 - 3) +[0-9]
 - 4) +[A-Z a-z]
 - 5) +[A-Z 0-9]
 - 6) +[a-zA-Z 0-9]
 - 7) +[a-zA-Z A-Z 0-9]

<script>
 function
 i
 var i
 if (i

- 1) [A-Z] \$
- 2) [a-z] \$
- 3) [0-9] \$
- 4) [A-Z A-Z] \$
- 5) [A-Z 0-9] \$
- 6) [a-z 0-9] \$
- 7) [a-z A-Z 0-9] \$

which is
easier
→ case insensitive
→ case sensitive
global

source
occurrence

uppercase A-Z

option

prel: check

<script>

var a = " Is spider site & it is spiders";

var reg = /E^a-2 A-2 0-9]/gi;

var res = a.match(reg);

cl.co (res)

</script>

<head>

<body> </body>

Validations

code for required Attribute (to avoid empty spaces)

37

-47 Only:- <body>

-47 <form action=" wel.html " >

-2 0-9] username: <input type="text" id="un" >

<input type="submit" >

c

<script>

function valid ()

{

var user = document.getElementById('un');

if (user.value.trim() == "")

{

alert ("user should not be empty");
return false;

{

else
{

 alert (" user accepted");
 return true;

}

prnt:-

<body>
<form action = "wel.htm1" onsubmit = "return valid();"

if you didn't write return in a function
calling it will shows require box and also
will goto the welcome box

username: <input type = "text" id = "un"> class

password: <input type = "password" id = "pw"> class

checkbox type = "checkbox" >

</form>

<script>

function valid()

var user = document.getElementById ('un');

var pass = document.getElementById ('pw');

if user.value.trim () == "" || pass.value.trim () == "")

{

alert (" user should not be empty");

return false; // it is compulsory bcz

{

it won't work above return valid();

 it will allow to last page if you
 not declare any bcz return
 falseelse
{

alert (" user Accepted");

return true;

}

code to vu

prnt:-

<body>

<form action =

username : <input type =

Input type

2 forms

<script>

function

{

var user = doc

var my = 1

var res = my

if (res)

{

alert (" user

return

{}

return

{}

alert (" user

return

{}

alert (" user

return

{}

script "L"

Code to validate only Alphabets

pr1: <body>

<form action="wel.htm" onsubmit="return valid()>
 username: <input type="text" id="un">

 <input type="submit">

</form>

<script>

function valid()

{

var user = document.getElementById('un').value;

var reg = /\^([A-Z][a-z]+)\\$/g;

var rec = reg.test(user);

if (rec)

{

alert("username accepted");

return true;

}

else

{

alert("username should contain only alphabets");

return false;

}

?

I wrote return in a function
 It shows page goes back also
 It goto the welcome box

onsubmit="return valid();"

>

id="un">

id="rcv">

</form>

</body>

</html>

! ("un") ;

! ("rcv") ;

value.trim () == "")

return false;

script bcz

return valid()

to last page if you

any thing return

code to validate mobile number;

proj:- same as previous

```
var userv = document.getElementById('mob').value;
var reg = /\^ [6-9] [0-9] {9} \$/g;
if (reg) {
    var res = userv.test();
    if (res) {
        alert("Valid number");
        return true;
    }
}
```

else

```
alert("Enter valid number");
return false;
```

proj:- ~~same~~ code for confirming password match

function Valid() { Same as previous }

}

```
var pass = document.getElementById('pw').value;
var cpass = document.getElementById('cpw').value;
if (pass == cpass)
```

}

```
alert("password matching");
return true;
```

}

else

h

```
alert("password not matching");
return false.
```

}

code

proj:-

<body>

<form>

password

 (st)

length

<input

<form>

<script>

function

}

var pu

it /pass

if

document

= "

docum

n

m

7

else

{

al

7

else

{

al

7

code to check length of password.

prel: 2 border

```
<form encType="multipart/form-data" action="wel.htm">
  <input type="text" id="pw" style="border: 2px solid red; width: 150px; height: 30px;"/>
  <span style="color: red; font-size: small;">Length must be greater than 5
</form>
```

```
<input type="submit"/>
```

</form>

<script>

```
function validate()
```

{

```
  var pass = document.getElementById('pw').value;
```

```
  if (pass.length < 6)
```

{

```
    document.getElementById('msg').style.visibility =
      "visible";
```

```
    document.getElementById('pw').style.borderColor =
      "solid red";
```

```
    return false;
```

}

else

{

```
  alert("password accepted");
```

```
  return true;
```

}

}