Contents

[Tools every Web Developer should know about 4](#_Toc89037460)

[1. HTML 4](#_Toc89037461)

[1 head 4](#_Toc89037462)

[2 body basics 5](#_Toc89037463)

[3 table 5](#_Toc89037464)

[4 http input 1.0 6](#_Toc89037465)

[5 Input 2.0 7](#_Toc89037466)

[6 miscellaneous 8](#_Toc89037467)

[2CSS 9](#_Toc89037468)

[1 intro 9](#_Toc89037469)

[2 basics 10](#_Toc89037470)

[3 font and color 10](#_Toc89037471)

[4 Styling link and botton 11](#_Toc89037472)

[5 CSS Box Model 11](#_Toc89037473)

[6 Navigation Bar 12](#_Toc89037474)

[7 box alignment 13](#_Toc89037475)

[8 Flex Box 14](#_Toc89037476)

[9 Responsive design 15](#_Toc89037477)

[10 pseudo selectors 15](#_Toc89037478)

[11 static animation 16](#_Toc89037479)

[12 transition element 17](#_Toc89037480)

[13 CSS Grids 17](#_Toc89037481)

[14 miscellaneous 18](#_Toc89037482)

[3. JavaScript 19](#_Toc89037483)

[2 variable 20](#_Toc89037484)

[3 string 20](#_Toc89037485)

[4 fn 20](#_Toc89037486)

[5 user input 20](#_Toc89037487)

[6 loops 21](#_Toc89037488)

[7 navigating through html 21](#_Toc89037489)

[8 Events and listening events 21](#_Toc89037490)

[9 Set and Clear timeOut 22](#_Toc89037491)

[10 arrow fn 22](#_Toc89037492)

[11 maths fn 23](#_Toc89037493)

[12 JSON in Java Script 23](#_Toc89037494)

[4 Node.JS 24](#_Toc89037495)

[1 intro 24](#_Toc89037496)

[2 how to use module 24](#_Toc89037497)

[3 synchronous and asynchronous coding 25](#_Toc89037498)

[3 create server 25](#_Toc89037499)

[a Custom server 25](#_Toc89037500)

[4 create custom module 26](#_Toc89037501)

[how to use terminal 26](#_Toc89037502)

[5 install package in node.js 26](#_Toc89037503)

[5 other small backend application or api 27](#_Toc89037504)

[1 Postman 27](#_Toc89037505)

[2 Express 27](#_Toc89037506)

[3 pug 27](#_Toc89037507)

[6 Mongodb 29](#_Toc89037508)

[1 basic 29](#_Toc89037509)

[Administrative window powershell me waala part run it as admin 29](#_Toc89037510)

[2 inserting data 30](#_Toc89037511)

[3 searching /querying data 30](#_Toc89037512)

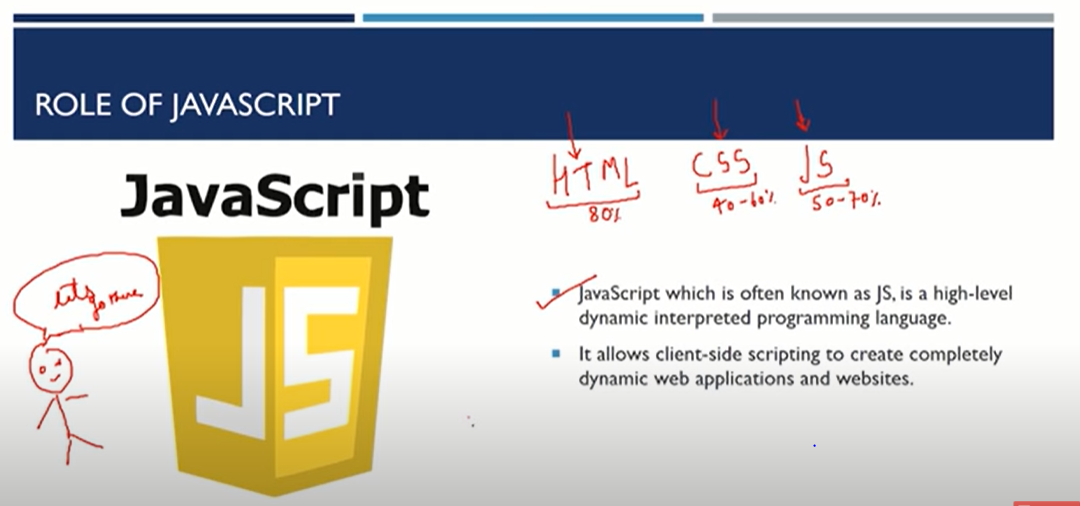
[4 deleting data 30](#_Toc89037513)

[5 updating data 30](#_Toc89037514)

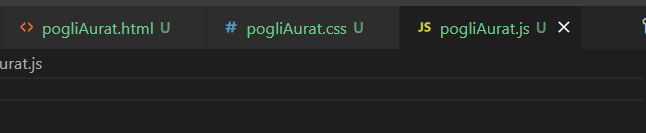
[7 Mongoose 31](#_Toc89037515)

[1 basics 31](#_Toc89037516)

[2 inside live website 31](#_Toc89037517)







## Tools every Web Developer should know about

* [UX wing](https://uxwing.com/) for Icons
* [PixaBay](https://pixabay.com/) for free Images
* [FreeHTML5](https://freehtml5.co/) for free templates
* [UnSplash](https://unsplash.com/) for Image API
* [Json Placeholder](https://jsonplaceholder.typicode.com/) for fake Online REST API
* More resources are available on this [GitHub Repository](https://github.com/bradtraversy/design-resources-for-developers)

# HTML

Vs code

//https://karngyan.com/

## 1 head

! html (Emmet Abbreviation)//syntax of website development //Emmet helps in writing fast coding in html, xml,xslt

<!DOCTYPE html> //Doctype is a html <tag jiske andar code kra jaata he>

<html lang="en"> //html opening tag tag sub tag lang(string represent language) En==english

// html > head > body \n //head me<meta> a set of data that describes and gives information about other data. <title>//title of search bar <external-link/sheets> \n //<!—for comments🡪

<control + </?> for comment n/ // body me actual content // <mata> use search engine optimization

<head>

    <meta charset="UTF-8"> // meta tag charset attribute

    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta name="description" content="naughty pogli aurat">

    <meta name="keywords" content="yaha pogli aurat dolti he">

    <meta name="robots" content="INDEX,FOLLOW"> //NOINDEX,NOFOLLOW

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

    <script src="pogliAurat.js"></script>

    <title>Pogli Aurat</title>

</head>

## 2 body basics

<body>

<h1>Pogli Aurat</h1> // h1>h2>h3>h4>h5>h6 heading \n //heading for heading only

<p>tum ek pogli aurat ho</p> //p only paragraph P\*4 for 4 paragraph

<p>Lorem40<p> //40 word continuous dummy text // command pallete >> toggle word warp (put them in line)

<strong/b> <strong/b> //for bold <em/i> <em/i> //for italic tag //not both can be change by use of css as per needs <br> //self-closing tag for line break

<hr> // self-closing tag for horizontal line //\n ctrl+enter for new line

<a href="https://www.facebook.com/">fb</a> //a fir bracket me //<a> is anchor

<a href="https://www.facebook.com/" target="\_blank">fb</a> // target= “blank” /n attribute to open link in new tab // for internal link put internal link in place of fb // attribute give quality to the tag

<a href="#contacts-header">Contacts</a> <h2 id="contacts-header">Contacts</h2>

<a href="#"> <a> // to create dead link, it can be used for other task eg for photo

//the above method was use to contact internal section within page using <a>

<img src="https://source.unsplash.com/random" width="100" height="150" alt="error"> //img is self-closing tag // img will display image in un src width for actual width height for actual height // \nelse display alt here (error) // if image is taken locally it should be present in local server ramdom display random image

<ul>tum <li>ek</li> <ul> //<ul> for bullet type based list <l1> be list //<ul> me heading aaegi <ol type="A"> <li>pogli</li> <ol> //<ol> me numbering aaegi except type se hum no type decide kr sakte he // list ke andar bhi list daal sakte he for loop ki tarah //

## 3 table

<table> <thead> <tr> <th>name</th>   <th>class</th> <th>roll no</th> </tr> </thead>

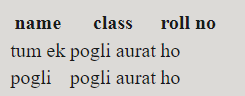
 <tbody>

  <tr> <td>tum ek</td> <td>pogli aurat</td> <td>ho</td>     </tr>

            <tr> <td>pogli </td> <td>pogli aurat</td> <td>ho</td> </tr>

</tbody>

</table>

//n  give this table as a result

</body>

</html>

## http input 1.0

<form action="pogliAurat.php"> //n type form ang give in put to either php or other backend

 <label for="name">Name</Label> //provide label to name ip ie if name is clicked it will select name id

<div> <input type="text" name="name" id="name"> </div> //text shows it will take text as input name is variable defined to it at backend id show label output //id use to identify this text

<div> Roll NO.: <input type="text" name="roll no" placeholder="roll no"> </div>

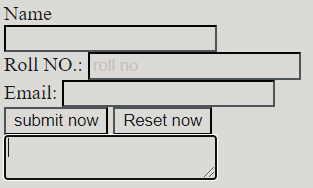
// placement holder will show light ip to guide user

<div> Email: <input type="email" name="email" required> </div> // I will take email address and is compulsory

<div> <input type="submit" value="submit now"> //it will submit data at backend

<input type="reset" value="Reset now"> </div> /it will reset all data at frontend

<div><textarea type="message" value="right anything"> </div> //to take message as input </form>

Output in http 

## 5 Input 2.0

<form action="https://www.freecatphotoapp.com/submit-cat-photo"> //form will submit data to the link url

<label for="indoor"><input id="indoor" type="radio" name="indoor-outdoor" value="indoor" checked> Indoor</label> //radio be option //checkeced means it will be default checked

                    <label for="outdoor"><input id="outdoor" type="radio" name="indoor-outdoor" value="outdoor"> Outdoor</label><br>

                    <label for="loving"><input id="loving" type="checkbox" name="personality" value="loving" checked> Loving</label>

                    <label for="lazy"><input id="lazy" type="checkbox" name="personality" value="lazy"> Lazy</label>

                    <label for="energetic"><input id="energetic" type="checkbox" name="personality" value="energetic"> Energetic</label>

                  </form>

Output in http 

## 6 miscellaneous

note: inline element which take only its space eg in above case w/o division or <span> in case of <p>

block which take whole block space eg in above case <div> and <p>

<div id= “mainBox” class=”redBg blackBorder”> </div> //mainBox beng id or identifier of this div class be summation of all property // note class will be defined in CSS appln

span.redBg // to put span class be redBg note dot will be used continuously to give data span#mainSpan //# to put span id be mainSpan

.blackBg //by default it will make div tag with class blackBg //\n hence Emmet keep div class as default

// java by default break spaces if one don’t want to break spaces use **&nbsp;**\*n for n spaces

// print <p> in <p> or paragraph use &lt;p&gt;

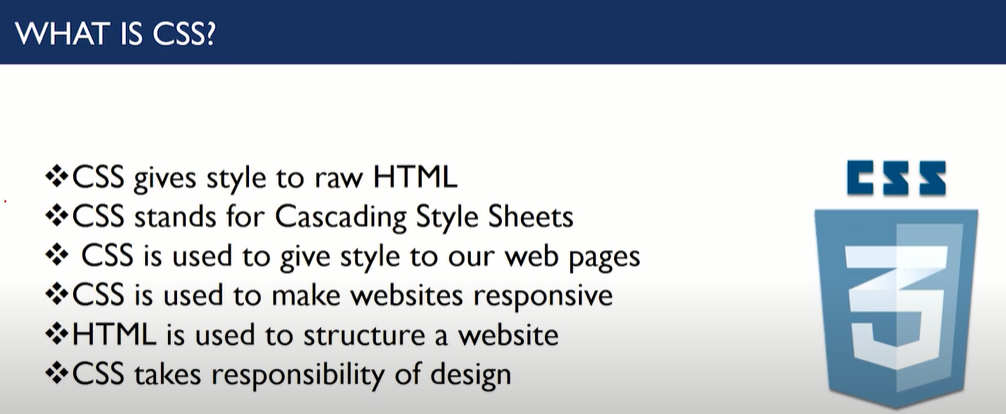
&pound; //to print pound symbol in <p> or<h(n)> &copy; //for copyright symbol

//HTML ENTITY REFERENCE ON GOOGLE //&#8203; for empty character

<details> <summary>check data<summary> tum ek pogli aurat ho

<details>

# 2CSS

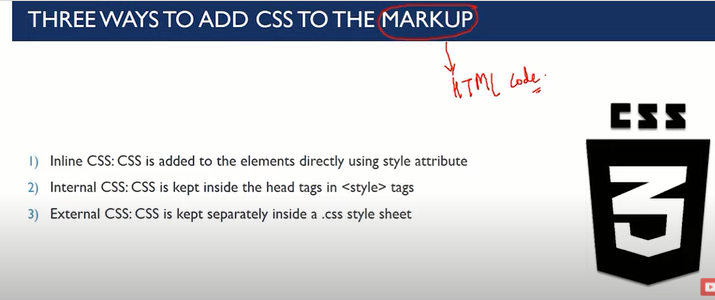


## 1 intro

p( color : blue;} //p = selector(where) color: = property(this) blue; = value(set value)

header, p.intro{ background-color : red; border-radius : 3px; } / header, p.intro being Group of selectors

, means dono jo bhi part ho //inside bracket{} declaration block

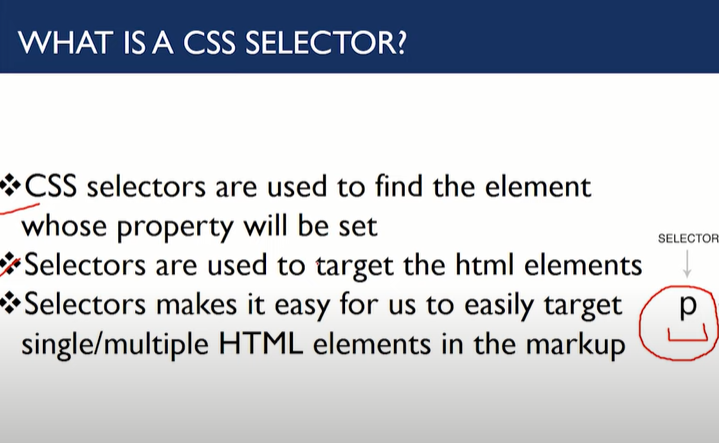


Inline matlab jo html me style de rhe the wo eg <p style= “\_”> internal matlab upper define krke or niche http daal ke //external matlab eg pogliAurat.css bana kr or include that css file to **Markup //** inline css has more priority than internal css

Internal css used <head> <style> (internal css) </style> </head>

For external css <link rel=”stylesheet” href=”pogliAurat.css”> /link:css fn se //in head only

Http me jo chij baadme likhi he uski priority jaada he ( baadme < - > priority ) else put (!important) fn with max priority

## 2 basics

CSS element Selector p { color : red;}

CSS id Selector eg(<p id= “red”> </p> #red{ color : red;} it will make p with id name red -> red

CSS class Selector .bgRed{ color : red;} //wherever class bgRed will be used it will make font colour red //multiple classes could be allotted to same element but only one unique id to one para

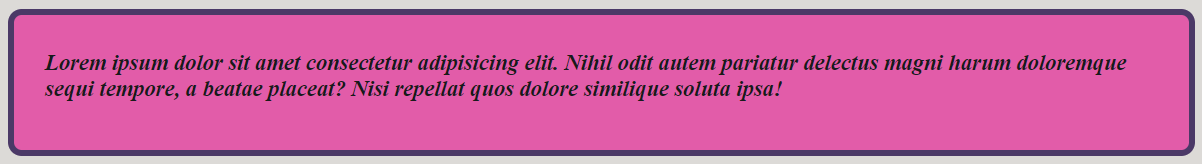
/\* comment\*/ for comment in CSS

border: 2px red; //will make border of 2 pixel red

// java css could be change by inspect element

## font and color

 .p1 { width: 900px; background-color: rgb(197, 157, 180); font-size: 18px; font-weight: bolder; font-style: oblique; border: 5px solid rgb(57, 5, 141); border-radius: 11px; padding : 10px 15px 20px 25px; margin: 10px 15px 20px 25px; box-sizing: border-box; }



float: left; width: 25%; clear: both; /to float box on left with width 25% and box will be cleared from both side text-align: right/ left/center/justify \*{ for all css headers}

## 

## Styling link and botton

<header> <nav class="navbar"> <ul>

<li><a href="https://www.facebook.com/" target="\_blank">fb</a></li>

<li><a href="https://www.youtube.com/" target="\_blank">youtube</a></li>

<li><a href="https://www.google.co.in/" target="\_blank">google</a></li>

<div class="search"> <input type="text" name="search" id="search" placeholder="Search this website"> </div> </ul> </nav> </header>

## CSS Box Model

a{me website} a{in general} a:hover{jb a ko serve kr rhe he} a:visited{agar a ko visit kr liya he} a:active{jb a active ho na ki visited ho}

font-family : ‘ Arial Narrow ’ , ‘ Arial Bold ’ , Times New Roman; //PRIORITY FROM LEFT TO RIGHT

<link href="https://fonts.googleapis.com/css2?family=Estonia&family=Noto+Serif+Devanagari&display=swap"

        rel="stylesheet"> // font family imported from google

    <style>

        p { /\*font-family: 'Estonia', cursive;\*/ font-family: 'Noto Serif Devanagari', serif; font-size : 28px; line-height : 1.25em; //em means relative to font size font-weight: bold; font-style: italic; } </style>

height : 20px; width  : 1000px; border : 4px , solid green; border-radius: 11px;

color : blue violet; color : rgb(15, 160, 45); color : #f5a709; // rgb in hexadecimal code

background-color : blue violet; height : 20px; width : 80px; border : 4px , solid green; border-radius: 11px; background-image: url(‘image link’); background -repeat : repeat-x / repeat / //or// no-repeat;

background position : center center // center top // center 300px//etc in 2d

## 6 Navigation Bar

<style> \* { box-sizing: border box; } //default boxing

.navbar { color: rgb(91, 28, 107); background-color: rgb(245, 210, 210); border-radius: 10px; }

// outside navigation bar .navbar ul { overflow: auto; } /\*to break list in line\*/ .navbar li { float: left; list-style: none; margin: 7px 3px; } /\*all three search bar on left and list-style will delete bulletins\*/

.navbar li a { padding: 1px 1px; text-decoration: none; } /\*t put external links .navbar li a:hover {color: rgba(119, 8, 86, 0.219); background-color: rgba(255, 198, 236, 0.384); } .search { float: right; color: rgba(255, 255, 255, 0.8); padding: 5px 5px; } .navbar input { border: 2px solid black; border-radius: 5px; padding: 3px; width: 150px; } /\*for searchbar\*/ </style>



## 7 box alignment

 img { width: 300px; margin: auto; display: block; } //margin auto will aligned image at center /\*-ve margin take whole padding space\*/

                    h3 { text-align: center; }//it will aligned text in h3 at center

                    .box { border: 4px solid indigo; margin: 5px; padding: 15px; width: 32%; box-sizing: border-box; display: inline-block;   color: rgb(5, 66, 47); } //inline- block will aligned all boxes at same line //border box will bordered text in box;

  box { **position: relative**; /\*relative to original posn of box and leave a gap at its normal posn\*/

**position: absolute;** /\*relative to position of its first parent or first containter; /\* by using posn on can move box left right top down \*/

**position: fixed;**  /\* than right posn \*/ /\*the position of box remains fixed with cursor at the point given\*/

**position: sticky;** /\*position of box stick to its original position\*/}box

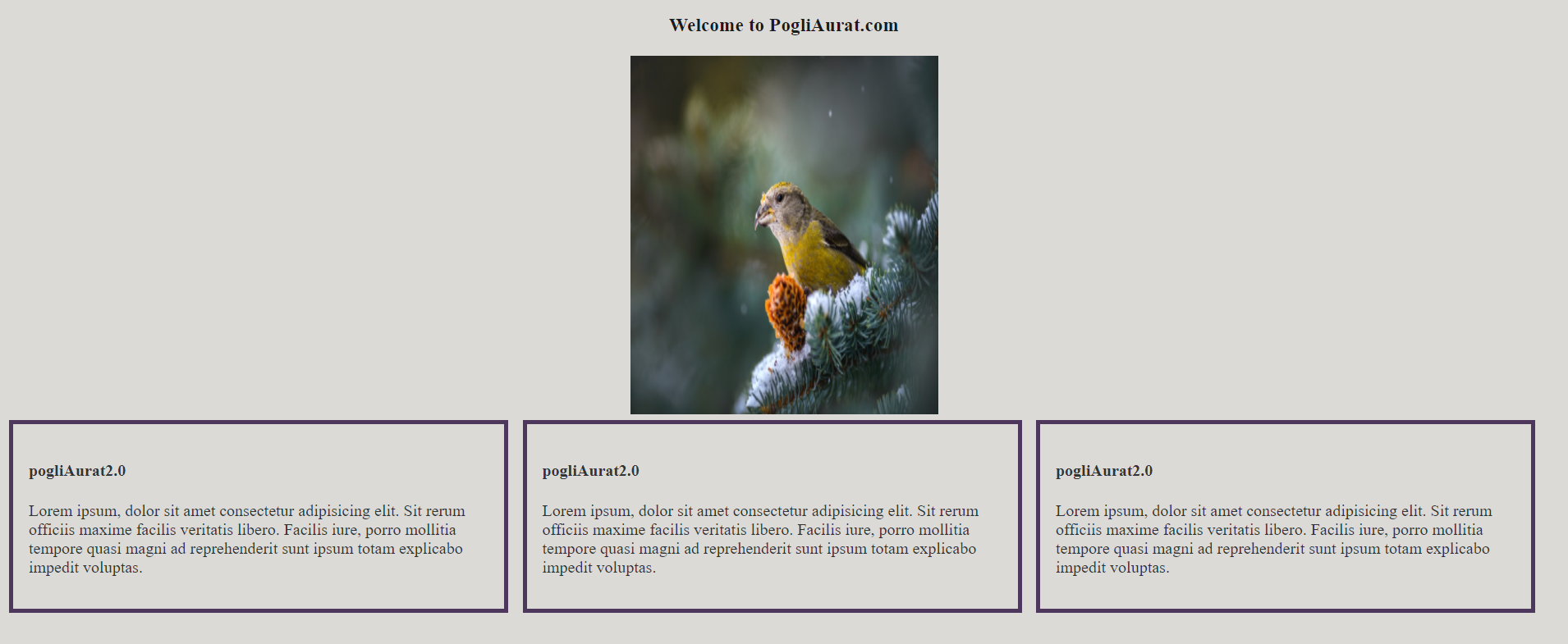
div.box#box$\*4 /\*$ will automatically no all box id as box1 box2… \*/

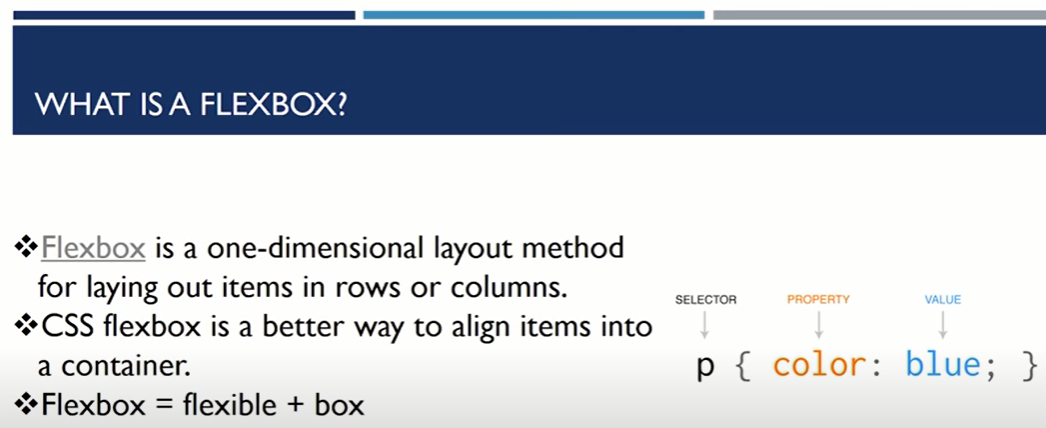
 background: url('img/pexels-anush-gorak-1431282.jpg'); height: 2000px; width: 1500px;

// above will make background image of 2000\*1500px cursor: pointer; will change cursor on button

visibility : hidden; // box gayab krke jagay reserve krne ke lie display : none; //box ki jagay or space ko hataane le lie //after displaying posn as relative absolute fixed sticky **z-index :** could be used to set priority of element;

for default set z-index : -1;





## 8 Flex Box

 display: flex; /\*to make box flexible to use website in tablet phone or pc effectively\*/ flex-direction: column-reverse; /\*to decide flex direction from starting to end flex-wrap: wrap; \*/to make sub-box stay inside main box\*/ flex-flow: wrap-reverse warp; /\*alignment of flex direction justify-content: center / space between / space around /\*to decide in between alignment of flex box align-items: center / flex-end /stretch /\*alignment of flex item wrt parent box

order:2; //to prioritize flex as per need, higher value means it will come afterwards flex-grow:3; to prioritize flex as per growing with increase stretch higher value means it will stretch faster //no -ve no. use flex-shrink:2; instead flex-basis : 320px //to limit box size in case of flex-growth or flex-shrink

/note size limit will be decided by above property\*/ flex: 2 2 340px; /to put flex-growth, flex-shrink, flex-basis together.\*/ align-self: flex-start or end / center; /\* where will box come with stretch will be removed\*/



## 9 Responsive design

Initialize font size: 10px; later in sub box it use eg 3em using such fn in sub container it will act as font size=3\*font size/3\*3\*font size … and so on; in all font size rem will multiply only once to take full screen width use height :100vh; /\*width be 100px and full height\*/ width:100vw; /\* height be 100px and full width\*/ /\*note to take full screen use both of above together\*/

@media (min-width: 300px) and (max-width: 500px) //tu use conditional css according to device screen

div > p{ prop} //kisi bhidiv ka first p me hi apply hoga div + p {prop} // kisi bhi div ke baad waale p pr hi apply hoga

## 10 pseudo selectors

/\*tu select input eg select imput text pseudo code is\*/ input[type = ‘text’ ] {prop}

given(<a href="https://www.facebook.com/" target="\_blank">fb</a>) a[target]{prop} /\*set property with target a\*/ /\*to be more specific\*/ a[ target= ‘\_blank’]{prop}

li:nth-child(3){prop} /\*change prop of 3 element in li only\*/ or li:nth-child(2n+0){prop} /\* change prop of above qn elements even for odd/even \*/

## 11 static animation

.box{ position: relative; /\*to max box dynamic\*/ animation-name: poppy1; /\*to initialized animation with name poppy1\*/ animation-duration: 2s; /\*time taken for each iteration\*/ animation-iteration-count: 3; /\*no of iteration in animation

@keyframes poppy1 { from{ width:200px; }/\*initial posn\*/ to {width:400px;} } /\*final posn\*/ /\*to max continuous animation make\*/ animation-iteration-count: infinite;

animation-fill-mode : alternate; /\*animation 1 time and stop in initial posn\*/

animation-fill-mode : forwards; /\*animation 1 time and stop in initial posn\*/

animation-timing-function : ease-in; /\*1ST SLOW THAN FAST REVERSE IN\*/ ease-out;

ease-in-out; /\*1st slow than fast and at last slow\*/ animation-delay : 3s; /\*to create initial delay before animation;\*/ animation-direction : forward/reverse; /\*dirn of animation in which it will take place\*/

@keyframes poppy1 { 0%{prop} 50% {prop} 75% {prop} 100%{prop} } /\*another way to make animation work\*/

animation : poppy2 5s ease-in 1s 12 backwards; /\*all prop set at once\*/ /\*{name durn timing-fn delay count fill-mode}\*/

header::before { background: url('https://source.unsplash.com/collection/190727/1600x900') no-repeat center center/cover; //create static background image content: ""; position: absolute; top: 0; left: 0; width: 100%; height: 100%; z-index:-1; opacity: 0.3; } /\*to set static background for changing display

.navigation {font-family: 'Bree Serif', serif; font-size: 15px; display: flex; }

section {height: 344px; font-family: 'Bree Serif', serif; margin: 3px 230px; display: flex; flex-direction: column; align-items: center; justify-content: center; } /\*to set flexible navigation-bar note by using flex item themselves aligned in order we want to either row or column

box-shadow: 2px 3px red;//2px shadow in x-axis 3 in y-axis //shadow could be -ve //if we add one more px in 3rd posn it will clear 3d blur effects add one more px it will give spreading effect like sun

/\*if we want shadows inside use\*/ box-shadow: inset 2px 3px red; use comma and write multiple color for multiple shadows same for text-shadows

Fn in css --box-color: blue; /\*above could be use as fn by using\*/ color(--box-color: blue;) /\*as fn syntax but only inside give class\*/

:root{ --box-color: blue;} /\*to use it globally in all css class or id\*/

## 12 transition element

#box{ transition-property : background color;} #box:hover{ background color red;} /\*it will change bkg color of box when hoveraround it\*/ transition-duraton :2s; transition-timing-function : ease-in; transition-delay : 4s; transition : background color 2s ease-in 4s; /\*above same as animation\*/

/\*In case of transition only property mention transit others will change immediately\*/

/\*To avoid above use\*/ transition : all 2s ease-in 4s;

Transform : rotate (45deg); /\*rotate box by 45 degrees\*/ /\*skew (40) will turn rectangle to parallelogram with angle 40 degree \*/ /\* scale (2) will make 2x size\*/ /\*translatex/y(100px) it will move in x or y dirn by 100px\*/ /\* translate(123px,100px) it will translate or move 123px in x dirn and 100px in y dirn\*/

## CSS Grids

display: grid; /\*to initialize 2d grid\*/ grid-template-columns: 300px 200px 100px; /\*set grid column 300px 200px 100px\*/ grid-template-columns: 300px 100px auto; /\*set 3rd column auto ie center of remaining space\*/ grid-template-columns: 1fr 4fr 1fr; /\*to divide whole grid space in 6 part 1/6 in1st one, 4/6 in mid one, 1/6 in last one\*/ grid-template-columns: repeat(3, auto); /\*divide grid in automatically 3 columns no matter how many elements it will have\*/ grid-gap:2rem; /\*set gap of 2x of margin and padding in grid\*/ /\*add\*/ grid-template-rows: 300px 100px auto; /\*to make grid 2d\*/

li.items{a}\*5 //create 5 list items with anchor a inside all list

html{ scroll-behavior: smooth; } //to make scrolling smooth

/\*run html website in incognito mode\*/ /\*first make destop css than make mobile css\*/ using fn <link rel="stylesheet" **media="screen and (max-width: 1150px)** " href="css/phone.css"> /\*set max width accordingly\*/

Devide css in part first eg 1./\* navigation \*/ 2./\* home \*/ 3./\* services \*/ 4./\* clients\*/ 5./\* contact \*/ 6./\* footer \*/ 7./\*utility section\*/ for project 2

/\*Observation: use inspect to check for phone 1st flex-direction:column; 2nd flex-wrap: wrap; 3rd adjust other things accordingly\*/ /\*give attention to id and class names and their fn\*/ a<href= “#id”> /\*could be used to call such ids\*/

Within http

Auto turn items into center

## 14 miscellaneous

[type='radio'] {prop} /\*for radio checkbox\*/

--ps: gray; background: var(--ps); background: var(--ps, black); /\*if ps had more than one colour\*/

text-transform: uppercase;

<audio id="meowClip" controls>

  <source src="https://s3.amazonaws.com/freecodecamp/screen-reader.mp3" type="audio/mpeg">

</audio> /\*same <a>

<figure> <img src> /\*fn\*/ <figcaption> Aaaa… </figcaption> </figure> /\*figcaption act as heading\*/

<form> <fieldset> <legend>heading</legend> /\*heading inside form\*/ inputs-lists </fieldse**t> </form>**

#navbar{ grid-area: navbar; } #section{ grid-area: section; } #aside{ grid-area: aside; } footer{ grid-area: footer; } /\*initialization of grid elements\*/

<div class= “container”> <div id= “navbar” class= “item”></div> <div id= “section” class= “item”></div> <div id= “section” class= “item”> </div>

</div> <footer class="item"> </footer> /\*INSIDE BODY\*/

.container{ display: grid; grid-gap: 1rem;

grid-template-areas: 'navbar navbar navbar navbar' 'section section section aside' 'footer footer footer footer '; } /\*navbar navbar act as row space\*/ /\* ‘ ’ to make columns for grid area\*/

grid-rows-gap:2rem;/\*to create gaps bw rows same for column\*/

/\*increase span for 1st box:\*/ .box: first-child{ grid-column-start: 1; grid-column-end: 3; grid-row-start: 1; grid-row-end: 3; } /\*it will create 2st box occupy 2 rows and 2 column\*/ /\* shortcut of above \*/ { grid-column: 1 / span 2; grid-row: 1 / span 2; }

grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); /\*it will create grid of min 1column and width be 300px but as size of screen increase it will increase column value in rows\*/ grid-template-columns: repeat(auto-fit, minmax(300px, 400px)); /\*repeat above except min column size is 300px and max size is 400px\*/

@media only screen and (max-width:300px) { body { background-color: red; }

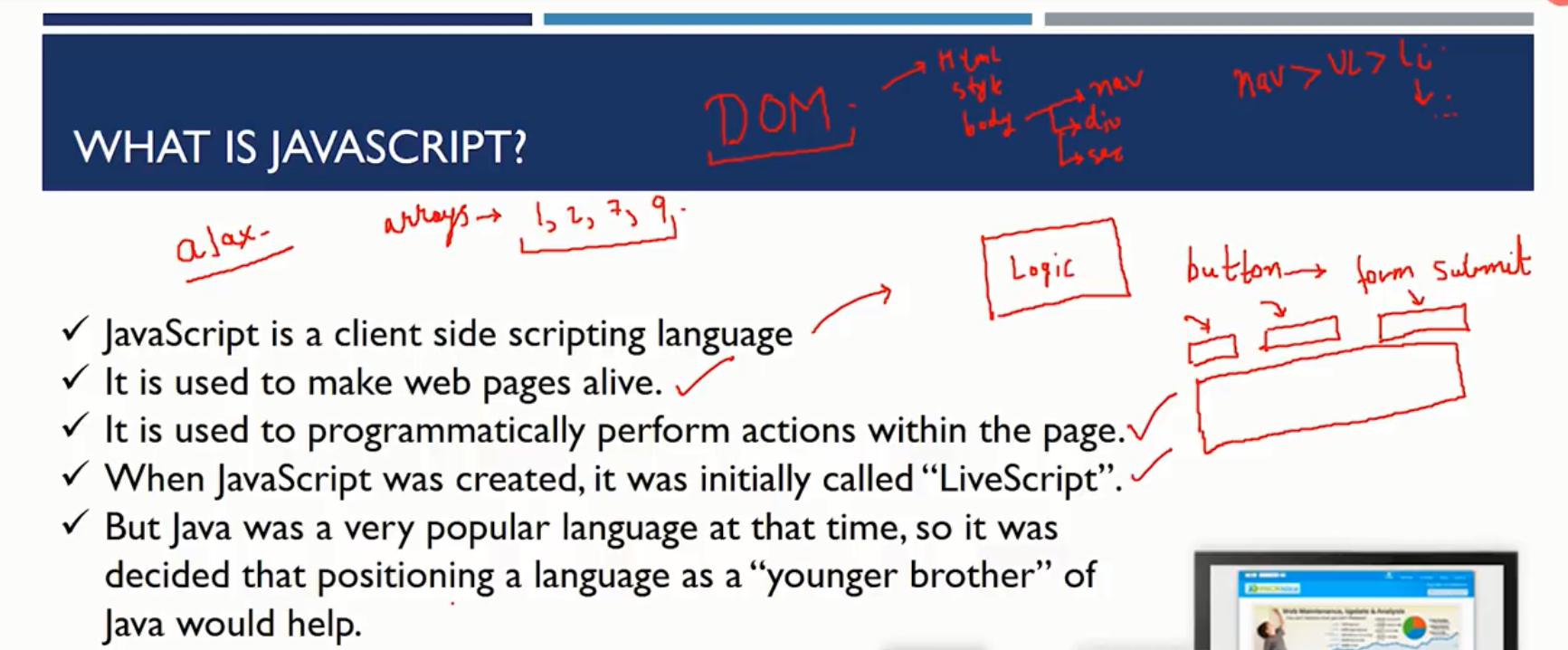
@media only screen and (min-width: 300px) and (max-width:500px) { body { background-color: blue;}

.container {grid-template-areas: 'navbar navbar navbar navbar' 'section section section section' 'aside aside aside aside' 'footer footer footer footer '; } aside{ display: none; } span{ display: block; `text-align: center; } }

@media (min-width: 500px) and (max-width:800px) { body { background-color: yellow; }

@media (min-width: 800px) { body { background-color: green; } }

# 3. JavaScript



What Can In-Browser JavaScript Do?

If JavaScripts are used in any websites, then it should not be given any low level CPU permissions like switching off the CPU etc. That is why JavaScript is made with extremely safe permissions that does not have any permission to access low level CPU usage.

JavaScript can add new HTML and change existing HTML from DOM.

It can even react to any events (actions).

It can also manage the AJAX requests (GET or POST request)

JavaScript can get and set cookies and use local storage.

What Can’t In-Browser JavaScript Do?

JavaScript cannot read or write to and from computer hard disk without user permissions.

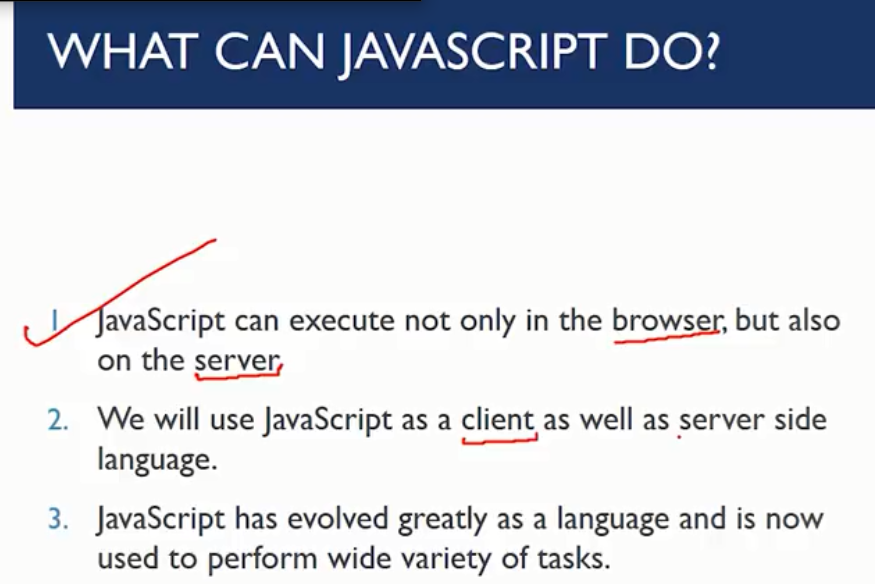
The browser does not allow the JavaScript of any website to collect the AJAX information of the other website because it generates the error of  same origin policy.

To summarize, JavaScript can only access the permitted resources but cannot access your documents on personal computers.

These strict policies are developed to make sure that your computer is safe.

**What Makes JavaScript a Unique language?**

* The most important thing that makes it a unique language is, it has a complete integration of HTML and CSS. They provide it with a lot of extra support.
* Also it provides the use of simple APIs (Application Programming Interface).
* It also supports the major modern browsers which are enabled by default. If you turn off the feature of JavaScript in the browser, you cannot access any website.



## 2 variable

console.log('hello world'); \* hello world\* var a=5; var b= “pogli”; console.log(a); \*5\* console.log(b); \*pogli\* /\* (\*\*) means power eg 2\*\*5 means 2^5 or 32 \*/

/\* for string both ‘ ’ and “ ” can be used if “ ” use inside only ‘ ’ can be used and vice versa\*/ /\*string were added like int in console.log();\*/

## 3 string

var c=` ${a} ${b} ho`; /\*in case of inside ` ` string bracket ${ } act as fill in the blank which can be stored by string or any other method \*/ var l=c.length;console.log(`${l}`); /\* l will store char length of String c and `${l}` method will give length stored in l

\n /\*new line\*/ \t /\*new tab\*/ /\*[\\n](file:///\\n) will print \n in console.log\*/ and so on.

 document.getElementById('content').innerHTML = '<h3>this is pogli aurat</h3>'; /\* document.getElementById will select id name content from innerHTML and give <h3>this is pogli aurat</h3> string as op\*/

var p= c.indexOf(‘is’) /\*it will give 1st occurrence of is in string\*/ c.lastIndexOf /\*it will give last occurrence of is in string\*/ c.slice(1,6); /\*it will print substring from char posn 1 to char

posn 5\*/ similarly c.substring /\*it work similar to slice but cant take -ve values\*/ c.substr(1,6); /\*it will start printing from char 1 to length of string as 6\*/

d=c.replace(‘pogli’, ‘pagli’); /\* in d string pogli will be replaced by pagli \*/ c.toUpperCase /\* make all elements capital\*/ /\*similar for toLowerCase\*/ c.concat(.) /\* will add . in string c\*/ c.trim /\*it will trim end and start space (but not middle spaces)\*/ c.charAt(5); /\*it will store 6th char of string\*/ \*/First use let instead of **var**\*/ eg **let** a= “pogli”; **cont** a= “pogli”; /\*however constant will remains constant and will not change;\*/ isme let a =[1, 2, ‘pogli’, null]; /\*array me kuch bhi daal sakte he\*/ a.sort() /\*to sort array\*/ a.length/\*for array length \*/

a.push( “aurat”); /\*new element will b push in array at last place\*/ /\*note let channel 2 should come like\*/ let “channel 2” let a= new array(23); /\* to create empty array of length 23\*/;

## 4 fn

<Script src= “pogli.js”> </script> /\*Now made pogli.js script\*/ /\*else code should be under <Script> </Script>\*/ /\*script exist under body\*/

function greet(name, greetText){ fn } /\*fn syntax with name, greetText as its variable\*/

/\* in fn we can give default value to fn variable which if user done give it will consider default value as input \*/ /\*for above fn as there is no return I cant be used in console.log\*/ /\*we need to give return (eg return a;) to give fn opat console.log\*/

## 5 user input

**alert**("pogli aurat so jao"); /\*to alert user\*/ /\*it is one way message\*/

let name = **prompt**("pogli aurat kya kr rhi he?", "Dol Rhi he"); console.log(name); /\*prompt will take input from user with qn "pogli aurat kya kr rhi he?" and default ans "Dol Rhi he" \*/ /\*such data could be store in var let or const and used further\*/

let deletePost = **confirm**("Do you really want to delete this post?"); /\* it will give Boolean answer as either true or false\*/ /\*to ensure conformity from user\*/

## 6 loops

/\*Loop for no same as java\*/ /\*for string or mix array\*/ for (let index = 0; index < a.length; index++) { console.log(a[index]); }

friends.forEach(function f(element){ console.log(element); });

for (element **of** friends){ console.log(element); }

let employee = { name: "Poppy", salary: 2, hobby: "pogli aurat" } /\* Use this loop to iterate over objects in JavaScript \*/

for(key **in** employee){ console.log(`The ${key} of employee is ${employee[key]}`); }

## 7 navigating through html

The HTML DOM (Document Object Model) is a standard **object** model and **programming interface** for HTML. It defines:

* The HTML elements as objects
* The properties of all HTML elements
* The methods to access all HTML elements
* The events for all HTML elements

let main = document.getElementById('main'); console.log(main); /\*it will give access to all elements of main id elements\*/ main.innerHTML= “<li>pogli</li> /\*it will change element inside main to pogli in list li\*/ /\*by using console\*/

let containers = document.getElementsByClassName('container'); console.log(containers); /\*above for class\*/

let sel = document.querySelector('container’); console.log("Selector returns ", sel); /\*If you want to find all HTML elements that match a specified CSS selector \*/

let sel = document.querySelectorAll('#nav');or(.container); console.log("Selector returns ", sel) /\*to get all selectors of #nav or .container\*/ /\*.for class # for id\*/

## 8 Events and listening events

Here is a list of some common HTML events-

* **Onchange-** An HTML element has been changed
* **Onclick-** The user clicks an HTML element
* **Onmouseover-**  The user moves the mouse over an HTML element
* **Onmouseout-**The user moves the mouse away from an HTML element
* **Onkeydown-**The user pushes a keyboard key
* **Onload-**The browser has finished loading the page

<button id="btn" onclick="toggleHide()">Show/Hide</button>

function toggleHide(){ let btn = document.getElementById('btn');

let para = document.getElementById('para');

if(para.style.display != 'none'){ para.style.display = 'none'; }

else{ para.style.display = 'block'; } }

/\* above fn to show or hide para when clicked\*/

Method 2

let para = document.getElementById('para');

para.addEventListener('mouseover', function run(){ alert('Mouse Inside') });

para.addEventListener('mouseout', function run(){ alert('Mouse now went outside') });

<!-- Browser events:

1 click contextmenu mouseover/mouseout mousedown/mouseup mousemove

2 submit focus DOMContentLoaded transitionend -->

/\*note all above code should be inside <script> (code here) </script>\*/

## 9 Set and Clear timeOut

There are times when we want a particular function should run after a specific interval of time**(setTime)** or it should run repeatedly after certain intervals**(setInterval).** In that case, we can use setTime and setInterval functions respectively./\*they are predefined fn in javascript\*/

The **setTimeout()**method calls a function or evaluates an expression after a specified number of milliseconds. This function is only executed once. Use the **clearTime()**method to prevent the function running.

<div class="container"> Time now is <span id="time"> </span> </div>

<script> function displayTime(){ time = new Date(); console.log(time); document.getElementById('time').innerHTML = time; } setInterval(displayTime,  1000); </script>

/\* time = new Date();\*/ /\*it is a function in js to fatch real-time in 5:30pm IST using API\*/

let newDate = new Date("2029-09-30"); console.log(newDate)

let newDate = new Date(3020, 4, 6, 9, 3, 2); console.log(newDate);

let yr = newDate.getFullYear(); console.log("The year is ", yr);

let dat = newDate.getDate(); console.log("The date is ", dat); /\*and so on.\*/

function greet(){ console.log("Hello Good Morning "); }

timeOut = setTimeout(greet, 5000); console.log(timeOut); clearTimeout(timeOut);

intervalId = setInterval(greet, 1000, "Harry"); console.log(setInterval); clearInterval(intervalId);

/\*note inside setTimeout or setInterval greet fn used as open fn with no brackets()\*/

## 10 arrow fn

function greet() { console.log('Good morning');} /\*such fn could be written as given below\*/

let greet = ()=> { console.log('Good morning'); }

let sum2 = (a, b) => a+b;

/\*difference bw arrow and normal fn arrow take outside parameter also as constant while normal fn don’t take outside fn as constant\*/

## 11 maths fn

let m = Math; console.log(m); /\*math.e give value of e math.PI give value of pi etc\*/ /\*Math is pre-defined fn\*/

/\*to generate random no\*/ math.random(); /\*b/w 0 and 1\*/

let a = 1; let b1 =100; let rbw100 = a + (b-a)\*Math.random();

## 12 JSON in Java Script

**What is JSON?**

* JSON stands for **JavaScript Object Notation**
* JSON is a lightweight data-interchange format
* JSON is language independent
* JSON is ***"self-describing"*** and easy to understand

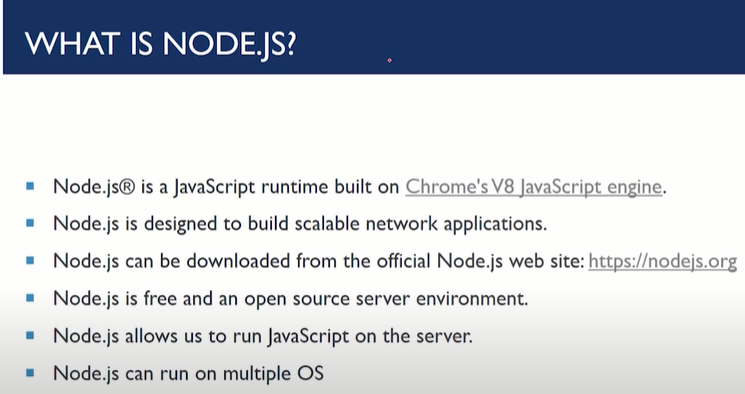
\*/In object was first converted to string before sending it into backend or any other programming language and may turn back into object again\*/

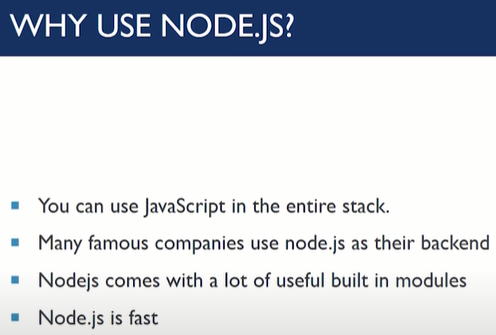
let jsonObj = { name: "poppy", role: "pati", gf: "pogli", full name: "pogli aurat" } /\* given object\*/

let myJsonStr = JSON.stringify(jsonObj); /\*first object was converted to string\*/

newJsonObj = JSON.parse(myJsonStr); /\*now converted string is turn back to object\*/

# 4 Node.JS





/\*To open node.js use **node** in new console\*/ /\*to exit node.js type **.exit**\*/

## 1 intro

const http = require('http'); const hostname = '127.0.0.1'; const port = 3000; const server = http.createServer((req, res) => { res.statusCode = 200; res.setHeader('Content-Type', **'text/plain'**); res.end(**'Hello World'**); }); server.listen(port, hostname, () => { console.log(`Server running at http://${hostname}:${port}/`); }); /\*note hello world with pogli aurat in **res.end** to change op with pogli aurat\*/ /\* change **‘text/plain’** to **‘text/html’** in **‘Content-Type’** and write code in **res.end(` code `);** to run original code in node.js**/\***

## 2 how to use module

const fs = require("fs"); let text = fs.readFileSync("po.txt", "utf-8"); console.log(“old file is ”text); text = text.replace("pogli", "pagli"); console.log("NEW file is"+text) fs.writeFileSync("po2.0.txt", text);

/\* fs being imported module\*/ /\* fs.readFileSync("po.txt", "utf-8") to fatch text from po.txt in nodejs\*/ /\*text.replace to replace old text with new one\*/

## 3 synchronous and asynchronous coding

The code which runs via line by line execution is known as **synchronous**or **blocking code.**It means the line of code written first, will be executed first. On the other hand, a block of code where line by line execution is not guaranteed is known as **asynchronous**or **non-blocking**code. These types of codes accept a call-back function.

/\*Ctrl +f =find syntax in node js documentation\*/ /\*use syntax in code\*/

eg for above code in asyn coding

const fs = require("fs"); fs.readFile("po.txt", "utf-8", (err, data)=>{ console.log(err, data); }); console.log("This is a message");

/\*asyn code req call back fn it will be used when we want to run prog w/o waiting for past command to execute which is issue of syn code where first will run and other code will execute\*/

/\*asyn code run almost parallel where part of code will start and other code will execute asynly hence it will make processing optimized\*/

## 3 create server

const http = require('http'); const fs = require('fs');

const fileContent = fs.readFile("po.html", (err, data)=>{ console.log(err, data); }); const server = http.createServer((req, res)=>{ res.writeHead(200, {‘Content-type’ : ‘text/html’}); res.end(fileContent) }); server.listen(80, ‘127.0.0.1’, () => { console.log(“Listening on port 80”); });

step1 /\* first http module was imported in http than const fs module was imported in const fs\*/

step 2/\*file Content in asyn mode based read data is stored in po.html file\*/

step3/\*than html server had been created in const server \*/

step4/\*last step server was colled by server.listen(80, ‘127.0.0.1’, () =>\*/ /\*fn in case of **80** will be used directly using webserver else no(eg 8000) mention had to be mentioned by **.8000** in generated link in server\*/

### a Custom server

const http = require('http'); const fs = require('fs');

const hostname = '127.0.0.1'; const port = 3000; const home = fs.readFileSync('index.html') const about = fs.readFileSync('./about.html') const services = fs.readFileSync('./services.html') const contact = fs.readFileSync('./contact.html')

const server = http.createServer((req, res)=>{ console.log(req.url); url = req.url; res.statusCode = 200; res.setHeader('Content-Type', 'text/html');

if(url == '/'){ res.end(home); }

else if(url == '/about'){ res.end(about); }

else if(url == '/services'){ res.end(services); }

else if(url == '/contact'){ res.end(contact); }

else{ res.statusCode = 404; res.end("<h1>404 not found</h1>"); } });

server.listen(port, hostname, () => { console.log(`Server running at http://${hostname}:${port}/`); });

## 4 create custom module

function average(arr){ let sum = 0; arr.forEach(element => { sum += element; }); return sum/arr.length; } /\*it is fn stored in module\*/

module.exports = { avg: average, name: "pogli", sirname: "aurat" } /\*create exports that will act as handshake api b/w above js and host js \*/

const mod = require("./mod"); console.log(mod.avg([3,4])); /\*mod will import custom module from above js\*/ /\*mod.avg will import object average whose fn was created\*/

STEP 1 /\*import http and files\*/

STEP 2 /\*create local host by hostname and port\*/

STEP 3 /\*request different pages created in front end eg here( **home, about, services, contact,** )

STEP 4 /\*create our custom server with arrow fn \*/

STEP 4a /\* create let url=req.url url be **STEP 4 fn** ip\*/

STEP 4b /\* run remaining code part\*/

STEP 4c /\* put condn to run requested pages in **STEP 3** fn\*/

STEP 5 /\* put created web server in console log to run by user by **server.listen** and **console.log**\*/

### how to use terminal

/\*cd folder name\*/ /\*tab to get complete name \*/ /\*cd .. /.. to revert back\*/ /\*node file.js\*/ /\*for file\*/

/\*cls for clear command\*/ /\*to exit program\*/ /\*ctrl+c\*/ in general /\*include ../js/index.js \*/ /\* ../ (to go back) js/ (to go inside js folder) \*/

## install package in node.js

STEP 1 /\*run node.js file in terminal\*/

STEP 2/\*type npm init in terminal\*/ /\* it will create pakage.JSON file for our current file\*/

Case1 /\*npm install slugify\*/ install slugify /\*shortcut **npm I \*/**

Case2 /\*npm uninstall slugify\*/ uninstall slugify

Case3 /\*npm install\*/ incase package got deleted due to some reason

Case4 /\* npm install [slugify@1.3.5\*/](mailto:slugify@1.3.5*/) install slugify version 1.3.5

Case5 /\* npm install nodemon --save-dev\*/ install nodemon only for save-dev

Case6 /\* npm install nodemon –-global\*/ install nodemon for all servers inside computer

"^4.16.1" /\*install this verision\*/

"~4.16.1" /\*install only this version\*/

">4.16.1" /\*install version above this\*/

"<4.16.1" /\*install version below this\*/

# 5 other small backend application or api

## 1 Postman

## 2 Express

/\*Create **Static** folder inside parent **Express** folder\*/ /\* put index .js inside Static folder\*/

app.use('/static', express.static('static')); /\*localhost/static/js\*/ /\*site will give code inside index.js\*/

const express = require("express"); /\*import express \*/ const app = express(); /\*create template to create custom app\*/ const port = 80; /\*initialize port 80 as reference port\*/

app.get("/", (req, res)=>{ res.status(200).send("This is homepage of my first express app with Poppy"); }); /\*to test app on get terminal of postman\*/ with status 200\*/

app.get("/about", (req, res)=>{ res.send("This is about page of my first express app with Poppy"); }); /\*to test app on guest terminal\*/

app.post("/about", (req, res)=>{ res.send("This is a post request about page of my first express app with Poppy"); });

/\*to test app on post terminal\*/

app.listen(port, ()=>{ console.log(`The application started successfully on port ${port}`); }); /\*to check for port execution last but essential step\*/

## 3 pug

html head title=title body h1=message /\*content inside demo.pug file which was used in above fn\*/

const path = require("path"); app.set('view engine', 'pug'); /\*create **view** folder inside **express** folder and generate **demo.pug** file\*/ /\*than write code which is below\*/

app.set('views', path.join(\_\_dirname, 'views')); /\*after generating folder\*/

app.get("/demo", (req, res)=>{ res.status(200).render('demo', { title: 'Pogli aurat', message: 'pogli aurat pug me itna dolo mat' }) }); /\*demo is pub-file need to be executed followed by arrow fn\*/ /\*title and message were mentioned in demo.pug file\*/

res.status(200).code /\*server is working fine if\*/ res.status(404).code /\*error 404 note code in general further code\*/

/\* two use pug file as html click pug icon below and type html inside it \*/

// ENDPOINTS

app.get('/', (req, res)=>{ const con = "pogli aurat mat dolo"; const params = {'title': 'Pogli Aurat kitna dologi', "content": con}; res.status(200).render('index.pug', params); })

/\*step1 create constants \*/ const express = require("express"); const path = require("path"); const fs = require("fs"); const app = express(); const port = 80;

/\* Step 2EXPRESS SPECIFIC STUFF\*/

app.use('/static', express.static('static')) /\*STEP 2a For serving static files\*/

app.use(express.urlencoded({extended:true}))

/\*Step 3 PUG SPECIFIC STUFF\*/

app.set('view engine', 'pug') /\*STEP 3a Set the template engine as pug\*/

app.set('views', path.join(\_\_dirname, 'views')) /\*STEP 3b Set the views directory\*/

/\*STEP4 ENDPOINTS\*/

app.get('/', (req, res)=>{ const con = "Pogli Aurat ka gym" const params = {'title': 'Pogli Aurat', "content": con} res.status(200).render('index.pug', params); }) /\*action on sending query to the user\*/

app.post('/', (req, res)=>{ name = req.body.name age = req.body.age gender = req.body.gender address = req.body.address more = req.body.more /\*take input from user\*/

let outputToWrite = `the name of the client is ${name}, ${age} years old, ${gender}, residing at ${address}. More about him/her: ${more}` /\*${fill in the blank fill by user input}

fs.writeFileSync('output.txt', outputToWrite) /\*to write file in text\*/

    const params = {'message': 'Your form has been submitted successfully'}     res.status(200).render('index.pug', params); /\*to send message your data had been submitted through object params\*/ })

/\*STEP 5 START THE SERVER\*/

app.listen(port, ()=>{ console.log(`The application started successfully on port ${port}`); });

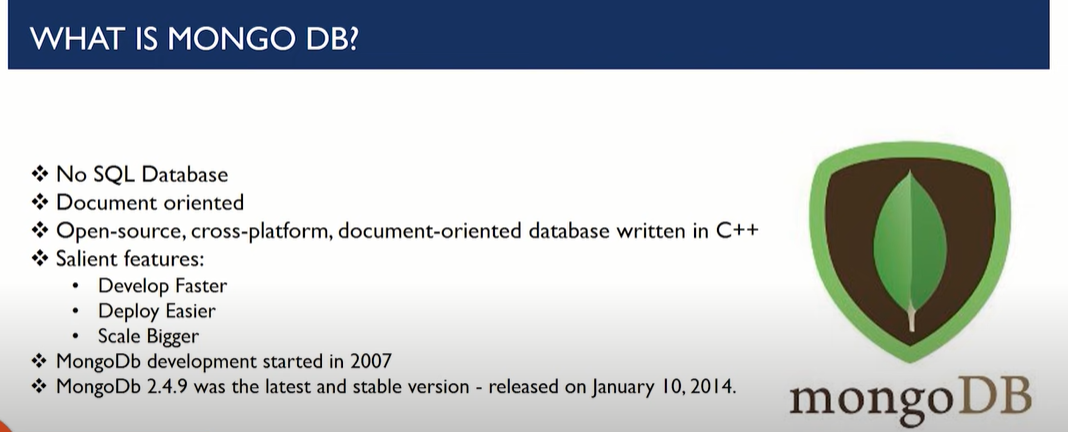
<h1> #{title} </h1> <p> | #{content} </p> /\* | is used in pug as new line #{} is used to define from backend code or endpoints \*/

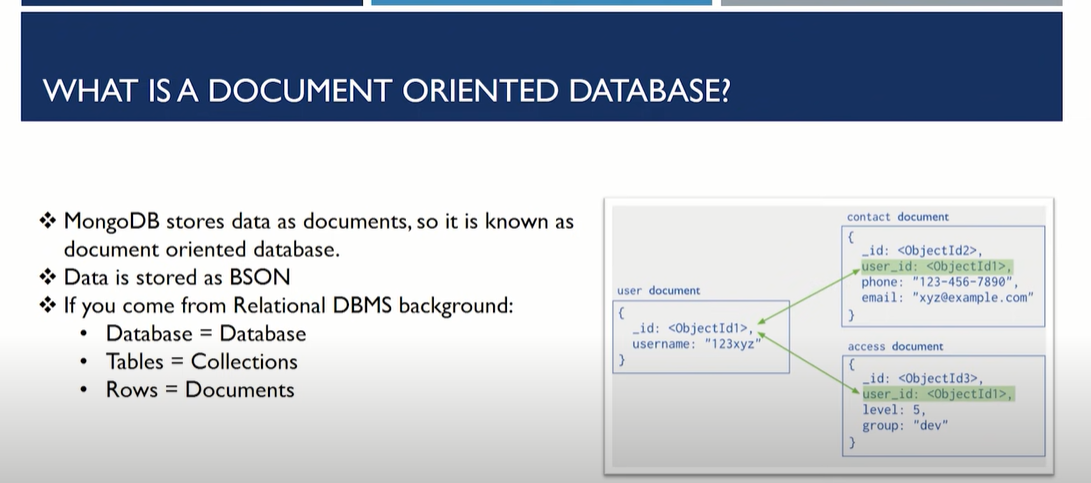
<style> include ../static/style.css </style>

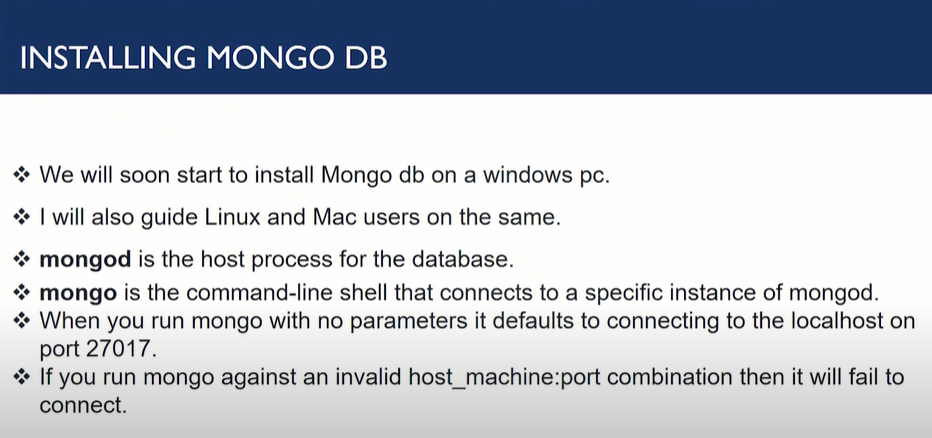
li #[a(href='#') About]

/\*pug to html converter online\*/

# 6 Mongodb







## 1 basic

### Administrative window powershell me waala part run it as admin

/\*to start mongodb using window powershell\*/

/\*Step 1 run\*/ mongod /\*run this command\*/ /\*run window powershell in another window\*/ /\*it is essential command to other file eg python, c++, js, mongo, etc in database\*/ /\*Step 2 run\*/ mongo /\*run this command\*/ /\*above command will be used to run command from remote database\*/ /\*Step 3 run\*/ db /\*run this command\*/ /\*op will show as test\*/ /\*Step 4 run\*/ use poppy /\*run this command\*/ /\*poppy database was created or switched\*/ /\*Step 5 run\*/ db.text /\*run this command\*/ /\*it will show poppy.text file\*/ /\*Step 6 run\*/ db.text.insert({}) /\*run this command\*/ /\*it will create original data with doc and poppy will be shown as database\*/

/\*To insert text in abovedatabase use command\*/ db.text.insert({“name”:”poppy”}) /\*it will write result in doc of poppy database\*/ /\*here text is collection in data base other collection could be items or of any name(x) \*/ /\*db.name(x).find(); /\*here anem(x) represent any name\*/

/\*other command of mongodb in window powershell\*/

## 2 inserting data

db.items.insertOne({name: "a", price: 2}) /\*code to insert one element\*/

db.items.insertMany([{name: "a", price: b}, {name: "a1", price: b1, }]) /\* code to insert one element\*/

db.items.find() /\*to find all elements\*/

/\*run\*/ show dbs /\*run this command\*/ /\*it will show all database\*/ /\*database atleast need one document to show in database collection\*/ /\* run\*/ show collections /\*run this command\*/ /\*it will show all collection of the selected database database\*/

## 3 searching /querying data

db.items.find({rating: 3.5}) /\*it will search all the object with rating: 3.5 \*/

db.items.find({rating: {$gt: 3.5}}) /\* $gt means greater than \*/ /\* $gte means greater than equa to\*/ /\*$lt means less than similarly $lte\*/

db.items.find({rating: {$gt: 3.5}, price:{$gt: 4000}}) /\* comma (,) act as and operator\*/

db.items.find({ $or[ { rating: {$gt: 3.5}, price:{$gt: 4000} } ]}) /\* the give fn is to make or iteration\*/

## 4 deleting data

db.items.deleteOne({price: 22000}) /\*Deleting only one item from the Mongo Database\*/

db.items.deleteMany({price: 129000}) /\*delete more than one item together in Mongo database \*/

/\*else same as searching (and, or combination)\*/

## 5 updating data

db.items.updateOne({name: "Moto 30s"}, {$set: {price: 2}}) /\*update only one item from the Mongo Database\*/ /\*with Moto 30s be searching such data\*/

db.items.updateMany({name: "Moto 30s"}, {$set: {price: 3, rating: 1}}) /\*delete more than one item together in Mongo database \*/ /\*else same as searching (and, or combination)\*/ /\*with Moto 30s be searching such data\*/

# 7 Mongoose

## 1 basics

const mongoose=require(“mongoose”); /\*step1 initialize mongoose\*/

main().catch(err => console.log(err)); async function main() { await mongoose.connect('mongodb://localhost:27017/pagliauratdance');} /\*STEP2 connect mongoose to local server with database pagliauratdance\*/

//CREATE SCHEMIA FOR MONGOOSE DATABASE

const contactSchema = new mongoose.Schema({ name: String, }); /\*doc format which will be stored in website by mongoose\*/

contactSchema.methods.speak = **function** **speak**() { **const** greeting = **this**.name ? "My name is " + **this**.name : "I don't have a name"; console.log(greeting); }; /\*

  const Contact = mongoose.model('Contact', contactSchema); /\*this code will be used to model doc in local server\*/

const Contact1 = new Contact ({ name: 'poppy' }); console.log(Contact1.name); /\* poppy will come in console.log which was object created in contact model of schemia \*/

await a.save(); /\*to save data in server with db Contact\*/ a.speak(); /\*it is save fn argument\*/

**const** kittens = **await** Kitten.find(name: “poppy”); console.log(kittens); /\*fn to find all poppy in pogliauratdance database\*/

**await** Contact.find({ name: /^Poppy/ }); /\* it will give filtered search\*/

## 2 inside live website

const mongoose=require(“mongoose”); /\*step1 initialize mongoose\*/

main().catch(err => console.log(err)); async function main() { await mongoose.connect('mongodb://localhost:27017/pagliauratdance');} /\*STEP2 connect mongoose to local server with database pagliauratdance\*/

//CREATE SCHEMIA FOR MONGOOSE DATABASE

const contactSchema = new mongoose.Schema({ name: String, phone: String, email: String, issues: String }); /\*doc format which will be stored in website by mongoose\*/

  const Contact = mongoose.model('Contact', contactSchema); /\*this code will be used to model doc in local server\*/

app.post('/contact', (req, res)=>{ var myData = new Contact(req.body);

myData.save().then(()=>{ res.send("This item has been saved to the database") }) /\*code to save data which was taken from server\*/

.catch(()=>{ res.status(400).send("item was not saved to the database") }); }) /\*in case of error such code run\*/