

JAWA SCRIPT INTRODUCTION

: - Jawa script se ham apne page ko interactive banate hai. Or esase ham back end bhi kar sakte hai Jiske liy hame NoteJS download karna hota hai. JS file ko ham body ke just pahale lagate hai script tag ka use karke src me JS case sensitive hai.

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

Alert("hello world"); Es properte se alert aata hai page par.

```
alert("hello world");
```

console.log("hai"); Es properte se console par kuchh bhi print kar sakte hai.

```
console.log("hai i am amar")
```

Prompt("hai"); Es properte se ham input le sakte hai alert box me.

```
prompt("Enter your number : ")
```

Yha par ham input to le lenge lekin print karane ke liy hame ese console par print karna hogा. Jiske liy hame variable bana padega . Variable ak contener hota hai jisme ham values ko align karte hai.

```
var a = prompt("Enter your name : ")
```

```
console.log("Your Name is : " +a)
```

variable ko assign karne ke liy (+) ka use karte hai.

VARIABLES DATA TYPE AND OBJECT

: - Variables ji ak contener hota hai jisme ham values ko assign karte hai. variable banana ke liy ham (var),(let),(const) keyword ka use karte hai var dublicat ho sakti hai lekn const dublikat nhi ho sakti hai.Or **ham hamesa (let) ka use karenge.Kuki let block scope pe kam karta hai.**

: Variables ke sath ham sabhi mathameticle opration kar sakte hai.

```
console.log("hai kaise ho jaldi se 2+2 karo or 3*4 karo")
let a = 2;
let b = 2;
let c = 3;
let d = 4;
```

```
let Sum = a+b;
let Mul = c*d;
console.log("Sum value " +Sum);
console.log("Multiply value " +Mul);
```

Variable 2 prakar ke hote hai globle or block globle .Jo upar variable bane hai vo sab globle hai.

Block variable : - {

```
let a = 33;
console.log(a);
}
```

PRIMITIVE DATA TYPE

```
let e = "Amar bhai"; //string type;
let f = 34; //number type;
let g = 3.0; //number type;
const h = true; //boolean type;
let i = undefined; //undefined type;
let j = null; //object type eska type null hota hai lekin js object dikhata
              //hai.;

console.log(e,f,g,h,i,j); //multiple variable print karne ke liy sab ak line me
                           //print hote hai;
console.log(typeof e,typeof f,typeof g,typeof h,typeof i,typeof j); //variable
                      //type pata karne ke liy;
```

OBJECT

:- object key values ke pyar me hoti hai key ko ham ("") ke andar likhte hai. Lekin agar key me space nhi hai to ("") lagana jaruri nhi hai. Multiple key value jodne ke liy (,) se seprate karte hai. Object ham tab banate hai jab hame dher sari detail ak sath add karni hoti hai.

Jaise :-

```
let Obj = {
  name : "amar",
  age : 20,
  "job role": 90000

}
console.log(Obj);
```

:- Ab agar bad me esme kuchh add karna ho to

```
Obj. Salary = "7000corors";
console.log(Obj);
```

Ye value Obj ke last me jud jaygi.

Conditionals : if,else

:- if Esase ham condition lagate hai else agar if condition sahi nhi hoti hai to else part run hota hai.

```
let age = 34;

if(age > 18){
  console.log("You Can drive");
}
else{
  console.log("You can not drive");
}
```

Code matlab :- Agar variable age me value 18 se jada hai to print kar do you can drive nhi to else part ko print kar do you can not drive.

Ham grace bhi de sakte hai jaise school me bachche 1 ya 2 number se fail hote hai to unhe grace dekar pas kar diya jata hai.

```
let age = 16;
let grace = 2;
if((age+grace) >= 18){
    console.log("You Can drive");
}
else{
    console.log("You can not drive");
}
```

Ye 16 sal ka hai lekin ese grace mili hai to ye 18 ka ho gya hai.

Ham Calulatin me sare Arthemetcle, Logicle, Alignment, True, False etc... calculation kar sakte hai.

Else if – Es statement se ham multiple condition lagate hai.

```
let age = 0;
let grace = 2;
if(age == 18){
    console.log("You Can drive");
}
else if(age == 0 ){
    console.log("are you kidding");
}
else{
    console.log("You can not drive");
}
```

Ye if se puchhe ga ki kya age 0 ke barabar hai if false dega fir ye else if ke pas aayga yaha ese true milega to ye print karega are you kidding.

Turnery Operator (?) – ham jo if else statement se kae line ka code likhte hai use ham es oprator se ak line me hi likh sakte hai.

```
// ternatry statement.
a = 5;
b = 9;
c = a > b ?(b-a):(b+a);
console.log(c); //14 Ans

//if else statement
if(a>b){
    let c = b - a;
}
else{
    let c = b + a;
}
console.log(c)//14 Ans
```

JS LOOPS

: - loops for loop , while loop, do while loop, etc.. loops ki madat se ham values ko repeat karate hai
alag - alag loop alag - alag tarike se kam karte hai lekin sabka kam sem hota hai.

```
let a = 1;
for(let i = 0; i < 100; i++){
    console.log(a + i);
}
```

Code Matlab – insillise i me 0 hai , condition i jabtak chhoti hai 100 se , increment dicriment tabtak i me 1 + karte roho. Fir a me i ko + kro or console.log par print karo.

Kabhi kabhi condition ki vajah se infinit loop ban jata hai esliy hame hamesa dhyan dena chahiya or code ko 2 bar check karna chahiya.

Object ki values ko bhi ham for loop se bari - bari print kar sakte hai.

```
let obj = {
    name: "amar rajbhar",
    age: 18,
    job: "web devlopor",
    salary: 500000
}
for(const key in obj){
    const element = obj[key];
    console.log(element);
}
```

Es code se sirf value print hogi or agar key bhi print karni hai to console.log me (key,element) dono value deni hogi.

```
console.log(key,element);
```

For of loop – Es loop me jo bhi value dete hai vo ak - ak karke print hoti hai jada tar eska use ham arry par karte hai. Jaise arry ka ak – ak element aata hai values hi .

```
for (const c of "amar"){
    console.log(c);
}
```

While loop – Es loop me ham pahale condition dete hai fir print karte hai or bad me increment dicriment karte hai.

```
let a = 5;
while(a<=10){
    console.log(a);
    a++;
}
```

Agar a++ nhi karenge to value nhi badegi or ye **infinit** loop ban jayga .

Do while loop – Ye kam se kam 1 bar jarur run hota hai. uske bad condition chek hogi or code aage jayuga.

```
i = 20;  
do{  
    console.log(i);  
    i++;  
}while(i<=6);
```

Eisme condition hai ki kya 1 chhoti hai 6 se to ye False hai lekin ye ak bar chalega or 20 ko print karega uske bad 20 ko 21 karega or fir condition check hogi ki kya 21 chhoti hai 6 se to ye False hogta to code aage nhi jayga.

FUNCTION

: - JS me function se ham apna ak block function banate hai apne condition ke hisab se or us function ko ham jitna chahe utni bar use kar sakte hai function call karke.

```
function nice(name){  
    console.log("Hay" ,name, "is good boy");  
    console.log("Hay" ,name, "is hansom boy");  
    console.log("Hay" ,name, "is mentally strong boy");  
    console.log("Hay" ,name, "is shree ram's friend");  
}  
nice("amar");
```

: - Function ka name (nice) hai or function me pairamiter pass kya gya hame (name) name ka uske bad jo bhi karna hai vo ham function me karte hai fir function ko call karte hai tab sari statements print hoti hai. (Jitni bar function ko call karenge utni bar ye statement print hogi).

Paramiter :- Parimitter ham kitna bhi de sakte hai (,) ka use karke pairamiter me value function ko call karte samky dete hai.

Function me ham vo sab kar sakte hai jo bina function ke bhi kar sakte hai.

Sum function –

```
function sum(a,b,c){  
    console.log(a+b+c);  
}  
sum(3,4,5);
```

: - ham sari calculation kar sakte hai.

Return – Esme ham jo bhi value denge vo ye return karega.

```
function sum(a,b,c){  
    // console.log(a+b+c);  
    return a+b+c;  
}  
result = sum(3,4,5);  
console.log("The sum value is next variable align : ",result);
```

Agar ham return ka use nhi karenge or sum ki value ko direct result me align karenge to nhi hogi kyuki js ko nhi pata hai ki result me kya align karna hai kyuki use kuchh return me mila hi nhi hai. or return ka use karne par function ko ak value wapas milti hai to function ab result me us value ko align dega jo wapas mili hai. Ham ese multiple bar estemal kar sakte hai.

```
function sum(a,b,c){  
    // console.log(a+b+c);  
    return a+b+c;  
}  
result1 = sum(3,4,5);  
result2 = sum(6,45,8);  
result3 = sum(12,23,45);  
console.log("The sum value is next variable align : ",result1);  
console.log("The sum value is next variable align : ",result2);  
console.log("The sum value is next variable align : ",result3);
```

Optional pairamiter :- Eska matlab esme value do ya na do aapki marji **Lekin** ham esme ak **DEFOUTL** value set kar sakte hai taki pairamiter me value na pass kiya jay to default value print ho jay Or agar ham c pairamiter me koe value dete hai to Yahi value pirnt hogi default value overwrite ho jaygi.

```
function sum(a,b,c=5){  
    // console.log(a+b+c);  
    return a+b+c;  
}  
result1 = sum(3,4); // 7Ans but defoult val (+) 12 Ans.  
console.log("The sum value is next variable align : ",result1);
```

Arrow Function – Ye ak type ka variable hota hai lekin ye function hota hai es function ko ham intern pass kar sakte hai ak dusre function me .

```
const func1 = (x) => {  
    console.log("i am a arrow functin ", x);  
}  
func1(2);  
func1("AMAR");  
func1(0.001);
```

STRINGS

:- Duble quvat ke andar ham jo bhi value rakhte hai vo string hoti hai ("string") ye **immutable hoti hai**;

Index – Index hamesha 0 se suru hota hai or eski madat se ham apni string ka ya kisi bhi data ka ko e ak element ya element ki shankhya ke hisab se 3 se 9 tak ya 1 se 10 tak ya ulta bhi value ko jaise chahe vaise ham slice kar sakte hai or print kar sakte hai.

```
let Name = "amar";
console.log(Name[3]);
```

Es code me var Name ke 3 index par jo value vo print ho rhi hai 0 par hai (a) or 3 par hai (r) to r print hoga. Indexing me ham space ko bhi count karte hai. Or range se jada index pass karne par Undefine dekhayaga.

Name.length – Ye property string me kitne caractor hai (string ki kitni length) hai vo nikalkar dega.

Template litrales Sintext – Kabhi kabhi hame text or var value bar bar print karni hoti hai jisme hame kabhi kabhi problem bhi hoti ha or esi problem ko solve karne ke liy ham **Template litrales Sintext** ka use karte hai. Esme ham (`) es symbol ka use karte hai or var value likhne ke liy (\$) ka use karte hai. eske andar ham ("""') ka bhi use kar sakte hai.

Normal:-

```
console.log("My name " ,Name, "And Friend name " , friend , " and sister name
is ", sister);
```

Template Litrales Sintext:-

```
console.log(`My name ${Name} and my friend name is ${friend} and my sister
name is ${sister}`);
```

toUpperCase () – Ye function string ko uppercase kar deti hai.

```
let b = "amar";
console.log(b.toUpperCase());
```

toLowerCase() – Ye function string ko lowercase kar deti hai.

```
let b = "AMAR";
console.log(b.toLowerCase());
```

slice(value) – Slice function se ham index ki madat se string ko slice karte hai jaise - mera name (amarrajbhar) hai to mujhe sirf (raj) chahiyo to mai index se ese slice kar sakta hu lekin ham jitni value denge **YE USASE AK KAM TAK CHALEGA**.

```
let Name = "amarrajbhar";
console.log(Name.slice(4, 7));
```

Agar ham sirf(starting) value denge jaise (1) to ye **1 se sari value ko print karega**.

Replace(starting , ending) – Ye function string ko replace karta hai starting me ham jo replace karna hai vo denge or ending me ham jisase replace karna hai vo dete hai.

```
let Name = "amarrajbhar";
console.log(Name.replace("rajbhar" , "007"));
```

Agar (rajbhar) kahi or bhi likha hai to jo sabse pahale hoga vahi replace hoga.

Concat(value) – Ye function 2 string ya variable value ko jodta hai ya add karta hai. ham multiple values ko add kar sakte hai (,) se seprat karke.

```
console.log(Name.concat("007")); //amarrajbhar007
console.log(Name.concat("007","Jai shree ram","ram","ram"));
```

ARRAYS

: - Array ke andar ham alag – alag data type store kar sakte hai , variables store kar sakte hai. Array ko ham [] ke andar banate hai . ham ak name par ke values ko add kar sakte hai Ye mutable hoti hai esme bhi ham index se elements ko acses kar sakte hai or slicing bhi kar sakte hai.

Jaise :-

```
let froots = ["mango","banana","papaya","apple","orange",]
console.log(froots);
```

Ham esme numbers bhi dal sakte hai.

```
let froots2 = ["mango",5,"banana",6,"papaya",1,"apple",2,"orange",3,]
console.log(froots2);
```

```
let froots = ["mango","banana","papaya","apple","orange",]
froots[3] = 34;
console.log(froots);
```

Aise ham array ke kisi bhi index ki value ko change kar sakte hai.

.length – Es property se element ki length pata kar sakte hai.

```
console.log(froots.length); //ye ak property hai function nhi hai.
```

Typeof arr – JS me arrays object type hote hai.

```
console.log(typeof froots); // object
```

.toString() - Es function se ham element ko string me convert kar sakte hai .

```
console.log(froots.toString());
```

.join(" hello ") = Es function se array join hoti hai jaise (,) ki jagah par hame kuchh or chahiyo to ham eska use karte hai.

```
let num = [3,4,5,5,6]
console.log(num.join(" and ")); // 3 and 4 and 5 and 5 and 6
.pop() – Ye method array ke last element ko return (nikalta hai) karta hai.
```

```
console.log(num.pop()) // 6
.push(100,"amar",0.2 etcVal...) – Ham esme kuchh bhi push kar sakte hai or ye value array ke last me add ho jaygi or ye push updated length ko return karata hai.
```

```
num.push(100);
console.log(num); // [3,4,5,5,6,100]
.shift() – Ye method array ke first element ko return karega . ya (nikalega).
```

.unshift(" kya hal hai") – ye method array ke starting me elements ko jodta hai.

```
num.unshift("kya hal hai");
console.log(num); // [ 'kya hal hai', 3, 4, 5, 5, 6 ]
.delete num[index] – Es method se value delete kar sakte hai lekin vo value memory me space legi or uski jagah par <empty item> likhega.
```

```
delete num[2];
console.log(num); // [ 3, 4, <1 empty item>, 5, 6 ]
```

Concatenat experiment.

Ye operation kene ke bad bhi a1,a2,a3 sab vaise hi rahega main array me ye kuchh change nhi karega esliy hame ese console.log ke andar hi likhna padta hai.

```
let a1 = [1,2,3]
let a2 = [4,5,6]
let a3 = [7,8,9]
console.log(a1.concat(a2,a3));
Ans [
  1, 2, 3, 4, 5,
  6, 7, 8, 9
]
```

.sort() – es method se elemets sort hote hai.

```
let a1 = [9,5,3]
console.log(a1.sort()); // [ 3, 5, 9 ]
.splice(start,end) – ye method slicing karta hai . jaise 1 se 4 likhne par ye 1 index se 6 index tak ki value ko cut kar dega or ab agar (,) dekar ham or value denge to ye un values ko vaha jod dega.
```

```
let a1 = [9,5,3,7,5,3,5,6,34,56,66,34,56,5];
console.log(a1.splice(1,6)); // [ 5, 3, 7, 5, 3, 5 ]
console.log(a1.splice(1,6,23,23,34,123));
jitni value cut hue hai unke jagaha par en values ko jod dega ye.sirf 1 ya 2 value bhi de sakte hai.
```

.slice(start,end) – Es function se ham array me se value ko slice kar lete hai or ye 1 kam tak chalta hai.

```
let a1 = [9,5,3,7,5,3,5,6,34,56,66,34,56,5];
console.log(a1.slice(1,4)); // [ 5, 3, 7 ]
```

Jaise esme 1 se 3 tak ka hi element print huua hai.

.reverse() – Ye function element ko reverse kar deta hai.

```
let a1 = [1,2,3,4];
console.log(a1.reverse()); // [ 4, 3, 2, 1 ]
```

loops with array

forEach loop – Esme 3 pairamiter pass kar sakte hai pahale array ki pahali value aaygi fir uska index aayga fir puri array print ho jaygi. Aise hi jitni bar arry me jitna element hogta bari - bari print hogta ham sirf ak pairameter bhi de sakte hai ak pairamiter denge to hame braket lagane ki jarurat nhi hai.

```
let a = [23,12,345,23];
a.forEach((value,index,arr) =>{//pahale value me 1st value aaygi fir uska
index aayga fir puri arr print hogi.
  console.log(value,index,arr);
});
```

For in loop – Eska use ham obj me karte hai uski key and values ko print karne ke liy or ham sirf key ya sirf value ko print kar sakte hai..

```
let obj = {
  a: 1,
  b: 2,
  c: 3
}
for (const key in obj) {
  if (Object.prototype.hasOwnProperty.call(obj, key)) {
    const element = obj[key];
    console.log(key,element);
  }
}
a 1
b 2
c 3
```

for of loop – Es loop ka use ham element ko bari – bari print karne ke liy karte hai ye bilkul python jaise kam karta hai.

```
let a = [23,12,345,23];
for(const value of a){
  console.log(value);
}// 23 12 345 23
```

Q- Original array ka hame square calculate karna hai or use dusre array me align karna hai original array normal hona chahiye.

```
let arr = [4,5,11,13];
let newArr = [];
for(let index = 0; index < arr.length; index++){
    const element = arr[index]; // arr ki value index ki madhyam se element me aaygi.
    newArr.push(element ** 2);
}
console.log(newArr);
```

Hamne etna sara code likhkar ye kam pura kiya hai lekin ham (forEach loop or map) ka use karke aasani se ye kam kar sakte hai.

```
let arr = [4,5,11,13];
let newArr = arr.map(e=>{ // ak pairamiter hai to hame bracket lagane ki jarurat nhi hai.
    return e**2;
});
console.log(newArr);
```

Code matlab – arr hamara same hai uske bad forEach loop ko hame pahale kisi variable me store karna hota hai fir loop ko start karte hai – to loop start karne ke bad arr ko map karke (e) pairamiter me store kar lenge bari – bari se fir (e^{**2}) karke return karke newArr ke pas bhej denge. Or fir print kar lenge newArr ko.

Filter – Q – agar hame arr me se vahi value print karni ho jo 7 se badi hai to ham filter ka bhi use kar sakte hai.

```
let arr = [4,5,11,13];
const greaterThanSeven = (e)=>{
    if(e>7){
        return true;
    }
    return false;
}
console.log(arr.filter(greaterThanSeven)); // [ 11, 13 ]
```

Jo GreaterThanSeven hai vo ak function hai.

Reduce – Q – agar hame factorial calculate karna pade to ham reduce ka use karte hai jaise hamare pass 1,2,3,4,5 hai to -1 ka * hoga 2 se fir 2 ka hoga 3 se fir 6 ka hoga 4 se 24 ka hoga 5 se 120 answer hoga aise hi ham +,-,/ etc sab kar sakte hai (**jo reduce function hai yhi ak – ak element par jayga.**)

```
let a = [1,2,3,4,5]
const red = (a,b)=> {
    return a*b
}
console.log(a.reduce(red));
```

Array.from ("amar") – Es function se ham kisi bhi obj ko array bana sakte hai.

```
let arr = Array.from("amar");
console.log(arr); // [ 'a', 'm', 'a', 'r' ]
```

DOM (DOCUMENT OBJECT MODULE)

:- Dom se ham html ke elements ko target karte hai or usme changes bhi kar sakte hai ya bol sakte hai ki dom se ham html elements ko map karte hai (Hame html ke kisi element ko JS me target karna hai to hame DOM se hokar gujarna padega) .

```
document.title = "jai shree ram";
```

Jaise yaha par ham document. Se title ko target karke title ko change kar rhe hai. title me jo hamne title diya hai pahle vahi rhega or or jab js load hogi tab title change ho jayga.

Ham document. Se kisi bhi element ko target kar sakte hai or print bhi kar sakte hai.

```
console.log(document.title);
console.log(document.body);
```

Es code se title console me print ho jayga or body bhi print ho jaygi or body ke andar jo element hoga vo bhi access kar sakte hai.Or eska use hota hai HTML element ko js se change karna. Ya element ki style bhi change kar sakte hai.

```
document.body.style.backgroundColor = "green";
```

Es code se body ka color green ho jayga or ye code body tag me inline css lagata hai.

BOM(BROWSER OBJECT MODULE)

: - Browser hame jo facility provide karta hai use hi ham browser object module kahate hai jaise Browser me hi ham element ka color change karte hai uspar style lagate hai jo bhi karte hai use hi ham BOM kahate hai.

Ye DOM or BOM or JavaScript core Window ke object hai Window ak global object hai esko ham nhi likhenge to bhi koe bat nhi hai Jaise Ham likh rhe the (document.body) lekin ese hame likhna chahiye

(Window.document.body) aise hame likhna chahiye lekin ham window nhi likhenge to bhi koe bat nhi hai.

document.body.childNodes : - Ye code browser par console me likha gya hai - Es code se body ke sare children print ho jayge or next line ka jo space hai use text kahate hai. Or agar hame kisi ak children ko target karna ho to index jaise ham kar sakte hai [0,1,2,3 etc...];

document.body.childNodes[1].childNodes :- Esse body ke child ke children ko target karte hai .

```
let cont = document.body.childNodes[1]
```

cont.firstChild :- Es code se ham body ke children ke firstChild ko target kar rhe hai.yha par ham index ka bhi use kar sakte hai.

cont.firstChild – Es code se ham body ke firstchile ko target kar rhe hai jo Element hai.or ab ham espar style bhi laga sakte hai .

cont.firstChild.style.backgroundColor = "red" – Es code se first chile element ka bg color red ho jyaga ham last chile element ko bhi target kar sakte hai.

document.body.firstChild.children – Es code se body ke first element ka sara element print ho jayga. Pahle code se sift 1 elemetn hi print ho raha tha kyuki ham ChildNodes ka use kar rhe the lekin ab children ka use kar rhe hai to sara element print ho rha hai.

```
let cont1 = document.body.firstChild;
let cont2 = document.body.firstChild.childNodes;
let cont3 = document.body.firstChild.children;
console.log(cont1);
console.log(cont2);
console.log(cont3);
```

Cont1 code se body ka firstElement or sath – sath child bhi print honge.

Cont2 code se body ke first element ke childNodes ke sath – sath text bhi print hogा (text ka matlab jo ham enter dete lagate hai or usase jo space aati hai use text bolte hai);

Cont3 code se body ke firstElement ke sare children yanhi sirf sare elements hi print honge.

Esme ham [0,1,2,3etc] index dekar kisi bhi element ko print kar sakte hai.

```
let cont3 = document.body.firstChild.children[3];
console.log(cont3);
```

Es code se box 4 print hogा kyuki index 3 diya gya hai.

NextelementSibling – Ease jo bhi elment abhi print hone vala hai just uske bad ka elemnt print hogा.

```
let cont4 = document.body.firstChild.children[3].nextElementSibling;
console.log(cont4);
```

PreviousElementSibling – Esase job hi element abhi print hone vala hai just uske pahale vala elemtn print hogा.

```
let cont5 =
document.body.firstChild.children[3].previousElementSibling;
console.log(cont5);
```

ParantElement - Esase children ke parant elemtn ko pata karte hai.

```
let cont6 = document.body.firstChild.children[3].parentElement;
console.log(cont6);
```

SELECTING BY IDS

:- Ham js je html element ko target karte hai or ab ham css selectors ki madat se html element ko js me select karenge – document.getElementById("id"); ya ClassName , ya ham query selectors ka use bhi karte hai.

Document.getElementsByClassName("name"); - Es selector se ham css class vale html elements ko target karte hai .

```
let myboxs = document.getElementsByClassName("box");
console.log(myboxs);
```

code – Jitne bhi element ka class name box hai sabko myboxs me align kar do or myboxs ko print karo.

```
myboxs[2].style.backgroundColor = "red";
```

code – myboxs ke index 2 par jo bhi element hai uska bg color red kar do.

: - Agar hame kisi particular element par style apply hai to ham id selector ka use karte hai .

```
let red = document.getElementById("red");
red.style.backgroundColor = "red";
```

code – Jiski id red hai uska bg color red karo.

QUERYSELECTION

:- Ye bhi id or class selectors ke jaise hi kam karta hai lekin ye thoda short tarika hai or esme ham class ke liy(.) lagaynge or id ke liy (#) ka use karenge lekin agar hamne box name ki class 4 element ko di hai to ye sirf first element par hi style apply karega .

```
document.querySelector(".box").style.backgroundColor = "green";
```

code – Jis element ki class .box hai uska bg color green karo lekin first vale ka sabka nhi.

: - Agar hame querySelector se sare element par style lagana hai to ham (querySelectorAll) ka use karenge lekin ye node list (html collection) return karta hai lekin ham style sirf elements par hi laga sakte hai esliy hame for loops ki madat se pahale sare elements ko iterable (iterate) karna padega fir bari - bari sab pe style lagani padegi.

```
document.querySelectorAll(".box").forEach(e =>{
  e.style.backgroundColor = "yellow";
});
```

Code - .box class vale elements bari – bari (e) pairamiter me jaynge or fir bari – bari unpar style apply hogi or sare element yellow ho jaynge.

TagName – Agar ham tag name se target karte hai to parent tag ke sath uske sare elements bhi print hote hai .

```
let tag = document.getElementsByTagName("div");
console.log(tag);
```

INSERTING AND REMOVING ELEMENTS

.innerHTML – Ham jis bhi element ko select karke uspar innerHTML lagaynge to uske andar jo bhi elements hain vo sab print ho Jayga.

```
let inner = document.querySelector(".container").innerHTML
console.log(inner);
```

.outerHTML – Esase container ke sath pura element print hoga.

```
let inner = document.querySelector(".container").outerHTML
console.log(inner);
```

.tagName – Esase tag ka name print hoga. Esame ham nodeName kabhi use kar sakte hai.

.textContent – Esase sirf element ka text print hoga element select hoga.

Agar hame innerHTML se text ko change karna hai to ham = ke bad jo bhi likhenge "" ke andar to vo change ho Jayga jo bhi element select hoga uska.

```
let inner = document.querySelector(".box").innerHTML = "hay i am amar"
console.log(inner);
```

hasAttribute – Esase ham pata kar sakte hai ki ak particular element me ak particular attribute hai ya nahi.

```
let inner = document.querySelector(".container").hasAttribute("style");
console.log(inner);
```

code – Esme puchh rhe hai ki kya style attribute .container me hai ya nahi hogा to True dega nahi to false dega.

getAttribute – Esase ham attribute ko get karte hai or us attribute me kya properties lagi hai vo dekhte hai. Matlab ki style attribute laga hai to getAttribute se style ke andar kya kya property lagi hai vo sab print ho Jayga.

```
let inner = document.querySelector(".container").getAttribute("style");
console.log(inner);
```

setAttribute – Esase ham attribute ko cheng kar sakte hai.

```
let inner = document.querySelector(".container").setAttribute("style" , "color = red");
console.log(inner);
```

code – pahale color blue tha lekin maine color red kar diay setAtt se.

attributes – Esase element ke andar jitne attribute honge vo sab print ho jaynge.

removeAttribute - Esase element par se ham attribute remove kar sakte hai.

```
let inner = document.querySelector(".container").removeAttribute("style");
console.log(inner);
```

```
document.designMode = "on"
```

Ese ham browser ke console par run karte hai or ese run karne ke bad duniya ki kisi bhi wesite ko ham apne hisab se change kar sakte hai bas console me jakar likhna hai or fir page par jakar uska text ham apne hisab se kuchh bhi likh sakte hai jaise unke wesite ka name hai Netflix to ham uska name amar likh denge akdam jaise ham typing karte hai vaise ham website me changes kar sakte hai .

Ham js se bhi html page me element add kar sakte hai.

```
let div = document.createElement("div");
div.innerHTML = "hai mai abhi abhi creat huua hu <b> by amar </b>";
div.setAttribute("class", "inser");
document.querySelector(".container").append(div);
```

code – sabse pahale ak element creat karo div name ka uske bad innerHTML se usme content dalo fir usme ak class attribute bhi set karo or uska name inser rakho uske bad selector ki madat se jaha par marji vaha par es element ko append kar do. Append method last me value ko add karta hai.

Ham kahi bhi append kar sakte hai body me bhi kar sakte hai jaise .

```
document.body.append(div);
```

.classList – Esase element ki sari calss aaygi.

.className – Esase element ki sari class ka name print ho jayga.

.classList.add("amar") – Esase selected element me amar name ki class add ho jaygi.

.classList.remove("amar") – Esase selected element me jo amar class hai vo remove ho jaygi ya koe or bhi class remove kar sakte hai.

.classList.toggle("red") – toggle ye bolta hai ki red class hai to hata do or agar red class nhi hai to laga do. Matlab ki element me red class hogi to hat jaygi or nhi hogi to lag jaygi.

EVENTS

:- Events me ham mouse ko set karte hai ki click karne par kya ho or hover karne par kya ho or double click karne par or klick karke rakhne par etc se sab me ham set kar sakte hai ki kya hona chahiya. Or kon sa event call hona chahiya etc..

addEventListener("click") – ye ak function hai jispar hamne click set kiya hai ham chahe to espar kuchh bhi set kar sakte hai .

```
let btn = document.getElementById("btn");
btn.addEventListener("click", ()=> {
    alert("kahe click kiye ho.");
});
```

Code – jis element ki id btn hai usko get karke btn me assign karlo or fir btn par even lagao ki jab btn par click kare to ak alert aay.

Content change even –

```
let btn = document.getElementById("btn");
btn.addEventListener("click", ()=> {
    document.querySelector(".box").innerHTML = "Ka ho button dekhte click kar dehalo.Tohara chakkar me ham chenj ho gaeni."
});
```

dblclick – Esase double click karenge to function kam karega.

```
let btn = document.getElementById("btn");

btn.addEventListener("dblclick", ()=> {
    document.querySelector(".box").innerHTML = "jai shree ram";
});
```

Mouse Events bahut hote hai ham chrome se MDN par ham dekh lenge. Or aise hi Keyboard ke bhi events hote hai.

Bubbling – Jab ham button par click event lagate hai or sath hi container par bhi lagayenge or uske parent par bhi lagayenge to jab ham sirf button par click karenge to by default li eske parent or fir uske parent fir body fir html par bhi click ho jaya ese hi bubbling karhati hai .

```
document.querySelector(".child").addEventListener("click", () => {
    alert("child was clicked.");
})
document.querySelector(".box").addEventListener("click", () => {
    alert("box was clicked.");
})
document.querySelector(".container").addEventListener("click", () => {
    alert("container was clicked."})}
```

code – Es code me hamne .box or .container par hi alert lagaya hai esliy en dono ka hi alert aayga lekin jab ham child par click karenge to .box or .container fir body fir html par bhi click hoga body html par hamne alert set nhi kiya to pata nhi chalega.

.stopPropagation() = Esase ham bubling ko rokte hai ham event me ak pairamiter pass karte hai or use .stopPropagation se stop dete hai lekin jispar ham ese lagaynge ye usi ki bubling ko rokega agar child par layenge to child ka bubling rok dega lekin uske parent ka nhi rokega parent par click karenge to fir vahi prosses hogta esliy hame sabpar ye lagana padega.

```
document.querySelector(".child").addEventListener("click", (e) => {
    e.stopPropagation()
    alert("child was clicked.");
})
document.querySelector(".box").addEventListener("click", (e) => {
    e.stopPropagation()
    alert("box was clicked.");
})
document.querySelector(".container").addEventListener("click", (e) => {
    e.stopPropagation()
    alert("container was clicked.");
})
```

setInterval(() =>{val}, miliSecond Val); - Ye function ak second function leta hai jisme ham mili second value dete hai or ye function us mili second me hi apna operation perform karta rhatा hai bar bar.

```
function getRandomColor() {
    let val1 = Math.ceil(0 + Math.random() * 255);
    let val2 = Math.ceil(0 + Math.random() * 255);
    let val3 = Math.ceil(0 + Math.random() * 255);
    return `rgb(${val1}, ${val2}, ${val3})`
}
setInterval(() => {
    document.querySelector(".box").style.background = getRandomColor()
}, 1000);
```

Code – Esase ham apne .box ka color har 1 second me change kar rahe hai random colors ganrat karke. Upar random color ganrat karne ke function hai or jo delay de rha hai vo setInterval function hai or esme hamne 1000 mili second yani 1 second diya hai taki ye har 1 second me color ko change karta rhe. Agar setInterval function ko ham var ke assign karke use print karenge to hame 1 milega khuki ye 1 return karta hai or fir ham **clearInterval(1);** karenge to setInterval ruk jayga.

setTimeout – ye akdam setInterval ke jaise hi hai sem to sem bas interval ki jagah par Timeout laga do ye sirf ak bar perform karta hai agar ham ese value de dete hai ki 5 se cond bad color change karo to ye 5 second bad color change karta bas ak bar hi.

Ese bhi ham clear kar sakte hai **ClearTimeout(1)** se eske color change karne se pahale agar ham clear kar dete hai to color change nhi hogta.

CALLBACK & PROMISIS

:- Collback ke andar ham function ko pass karte hai jisko callback function apne hisab se call karta hai matlab ham ak fun banate hai use ham callback ko de dete hai jo khud desite karega ki pahale mai ye sari calculation karunga fir tum call hoge ok .

To callback fucnotn aise kam karta hai ham callback function me kae callback fun pass kar sakte hai.

```
const callback = (arg) => {
    console.log(arg);
};

const loadScript = (src , callback) => {
    let sc = document.createElement("script");
    sc.src = src;
    sc.onload = callback("Amar");
    document.head.append(sc);
};

loadScript("function is call",callback);
```

code – Uapar hamne ak function banaya hai fir usko hamne callback function me pass kar diya hai or usko apne hisab se call kiya hai.Aise hi ham callback ke andar ham kae funcotn call dalte hai or call karte hai.

```
const Name = (b) => {
    console.log(b);
};

const fn = (a) => {
    console.log(a);
};

const callback = (arg,fn,Name) => {
    console.log(arg);
    fn("Nothing");
    Name("amar");
};

const loadScript = (src , callback) => {
    let sc = document.createElement("script");
    sc.src = src;
    sc.onload = callback("Amar",fn,Name);
    document.head.append(sc);
};

loadScript("function is call",callback);
```

Aise ham functins ko callback ke andar pass karke call karte hai ham kitna bhi fun pass kar sakte hai.

Aose hi ham jab bahut jada fun pass karege to code bahut jada complicated ho jayga jisko ham callback hell / ya pyramid of doom kahate hai. Or hame agar en sabko manage karna hai to ham **promises** ka use karenge.

Primitives – Esase ham js me cursor ko set karte hai. Ham chahe to cursor ko pahle dusri line me set kar sakte hai or bad me pahli line me set kar sakte hai isase pahle ham dusri line me likhenge or bhad me pahli line me likhenge. Cursor set karne ke liy ham \$1 likhte hai yani first line \$2 yani second line.

Primitives ko banana ke liy hame vs code ki setting me jana hia or Snipped par click karna hai or vaha par apni langvage ka name likhna hai (JavaScript) or enter karna bas. (video no 75 hai time line 17minutes par.)

Promise – Ese ham khud bhi bana sakte hai or esme ham condition bhi laga sakte hai Or eske andar ham **resolve,reject** pairamiter ka use karte hai resolve yani sab ok hai chalo reject yani error.

```
let prom1 = new Promise((resolve,reject) => {
    let a = Math.random();
    if(a < 0.5){
        reject("sorry number is not supported");
    }
    else {
        setTimeout(() => {
            console.log("Yes i am done");
            resolve("amar");
        }, 3000);
    }
});

prom1.then((a) => {
    console.log(a);
}).catch((err) => {
    console.log(err);
})
```

Hame jada tar promise bane huy milenge hame bas then or catch karke unko handle karna hogा jaise es condition me agar number 0.5 se chhota hai to reject kar do or jeject me jo stetment likha hai use error ke tour par de do or number sahi nikalkar aata hai to print else part ko run karo to ham prom1 fun me then ka use karke a ko print kar rhe hai or agar a false hota hai to error aaygi or error na aay eske liy hamne use catch kar liya or use console.log me print kar diya taki hamare console me error na aay.

Kul milkar ham then,catch kause error ko handle karne ke liy karte hai.

Promise ka matlab hi yahi hota hai ki ham error ko handle karke user ko sahi experience de sake error hone ke kae karad ho sakte hai jaise agar code load nhi huya ya server shi nhi chala ya user ka nettwork slow ho sakta hai ya kuchh bhi ho sakta hai to jini possible condition ho sakti ha ham unko then,catch ki sahayta se handle karte hai or error ki jagaha ham kuchh aasan likhkar user ko batate

hai jaise user ka net slow hai or error aane vali hai to ham use handle karke uski jagah par user ko dikhaynge ki (aapka net slow hai dobara try kiji). Aise hi errors handle hoti hai.

Async/Await & Fetch API

: - async ak function hai jo background me chalta rhata hai. Or agar ham chahate hai ki async function kisi perticular data par jakar vaha par entazar kare us data ka agar vo data bhi aaya nahi hai to Ab ham await function ka use karte hai jisko ham async function ke andar hi lagate hai.

```
async function getData() {
    return new Promise((resolve, reject) => {
        setTimeout(() => {
            resolve(455);
        }, 3000);
    })
}

async function main() {
    console.log("load module");

    console.log("load data");

    let data = await getData();
    console.log(data);

    console.log("data");

    console.log("Processes data");

    console.log("task 2");
}
main()
```

code – Ham jaise hi main ko call karenge vaise hi first console print hoga fir dusra hoga uske bad await ki sahayta se getData fun ko 3sec ke liy roka jayga fir uska bad baki ke console print ho jaynge. Aise hi ham bahut sare function creat kar sakte hai.

Fetch API – ye 2 promise return karta hai 1 data ko lata hai 2sara data ko pars karta hai.

TryCatch

:- Enka use ham errors ko handle karne ke liy karte hai .

Agar hame sum karna hai to ham code likhenge Ye.

```
let a = prompt("Enter first number : ");
let b = prompt("Enter second number : ");

let sum = a + b

console.log("The sum value is : ", sum);
```

Lekin ye code galat hai ye a or b ko sum nhi karega balki concat karega kyuki a or b str hai int nhi hai to hame a or b ko parseInt() fun ki sahayta se int me convert karna padega.

```
let sum = parseInt(a) + parseInt(b)
```

ab ye code sahi hai.

Ab agar apne user se number manga hai to jaruri nhi ha ki user number hi click karega vo kuchh bhi kar sakta hai vo apki wesite ke sath khel sakta hai kuchh bhi karke to agar vo aisa karega to error aayga or wesite me bugs aaynge to esi ko ab ham handle karenge.Or if ki sahayta se check karege or apne hisab se errors ko throw kar sakte hai throw yani ki bhejna.

```
let a = prompt("Enter first number : ");
let b = prompt("Enter second number : ");

if(isNaN(a) || isNaN(b)){
    throw SyntaxError("Sorry is not aloud.")
}
let sum = parseInt(a) + parseInt(b)

console.log("The sum value is : ", sum);
```

code – Esme check kiya ja rha hai ki agar a ya b jo value chahiya uski jagah par kuchh or pass kiya ja rha hai to syntaxError dijiy or print kijiya ki sorry is not aloud.

Aise hi bahut sare errors hoti hai js me jinko ham throw kar sakte hai.

JS me 2 prakar ki errors hoti hai ak jo ham dete hai yahi throw karte hai or ak jo JS khud deti ha jisko ham handle karte hai.

```
console.log("The sum value is : ", sum*x);
```

code – Yaha par ham x do define kiye bina hi use multiply kar rhe hai to JS hame error degi lekin ham Ese tryCatch ki madat se handle kar sakte hai.Kyuki Jab error aati hai to code usase aage nhi jata hai vahi par ruk jata hai esliy ham error ko handle karte hai taki code aagy ja sake.

```
try {
    console.log("The sum value is : ", sum*x);
} catch (error) {
    console.log("Ka ho x bhaiya to haeye nhi hai to guna kese karat bada. La
tab error");
}
```

Code – Pahale ham try me try kar rhe hai ki ye sahi hai ya nhi agar sahi hogा to result print ho jayga or nhi huā to catch use catch karke apne console ko print karega.

CLASS & OBJECTS

: - Eski madat se ham class banate hai or uske andar obj banate hai jisko ham apne hisab se set akrte hai matlab ki console.log function hai to usme ham jo bhi lihte hai vo console par print hota hai to ham chahe to apna khud ka kuchh bana sakte hai class obj se .

Sintaxt –

```
class Animal {
    constructor(name){
        this.name = name
        console.log("object is created....");
    }
    eats(){
        console.log("kha rha hu.");
    }
    jump(){
        console.log("kud rha hu.");
    }
}

let a = new Animal("kutta");
console.log(a);
```

ADVANCE JAVASCRIPT

: - JavaScript ke kam ke concepts = Dstructuring, Spredoperator, IIFE, Hosting.

```
// Dstructure method :- jaise a me 1 hai or b me 2 uske bad ka sara content  
rest me hai agar ... nhi Lagaynge to rest ak variable hoga or usme 3 hoga or  
uske aage ki value kahi asign nhi hogi. ...rest ye hamara sprade operator hai  
ye values ko khol deta hai.  
function main () {  
    let [a,b,...rest] = [1,2,3,4,5,6,7,8,9,10]  
    console.log(a,b,rest)  
}  
main()
```

Spred Operator – Ye values ko khold deta hai jaise upar bataya hai detail me samajhte hai.

```
let arr = [1,4,6]  
// console.log(arr[0] + arr[1] + arr[2]); // indexsing method se sum kar rhe  
hai.  
console.log(sum(arr[0], arr[1], arr[2])); // sum functoin se sum kar rhe hai.  
console.log(sum(...arr)); // Spred Operator se sum kar rhe hai.
```

AB DHER SARI PROBLEMS KO SHOLV KARNA HAI OR OR DHER SARA PROJECT BANANA HIA JABTAK
AAPKA HAAT NA BAITH JAY OR MAJA NA AANE LAGE.

PROJECTS BANANE HAI JAISE (SPOTYFI CLONE), (NETFLIX CLONE), (OYO CLONE) ETC....

JS KHATAM HO CHUKI HAI

JAI SHREE RAM.

Amar JS notes