Javascript ☺

Date __/__/____

Saathi

Hoisting : Variables & functions are hoisted which means their declaration is moved on the top of the code.

→ Sidha baat : Variable banne se pehle usey use kar sakte hain.

Primitives vs Reference :

Reference wale hote hain jinke pass Bracket hota hai. [], (), {}.

→ Aisi koi bhi value - jisko copy karne par usko reference pass hota hai naaki real value, that is known as reference while the object which copies its real value is known as Primitives.

Conditionals : (if, else, else-if)

```
if (condition) {
    // then this
}
else {
    //this
}
```

Loops :

```
for (var i=0; i<12; i++) {
    yeh wala console.log( );
}
```

Functions : Einhe hum code se pehle declare karte hain to use inside the code.

Date __/__/____

Saathi

```
funtion helobolo(){
        console.log("hello");
}
helobolo()
```

Output :- hello.

Arrays ⇒ when we have to store multiple number in same variable, then array. [].

```
Var a = [1,2,3,4,5];
console.log(a);
```

output :- 1,2,3,4,5. a[0] = 1, a[2] = 3

Push pop shift, unshift ⇒

Push ⇒ To add a member in array.

```
arr.push(8);
```

out :- 1,2,3,4,5,8

Pop ⇒ To remove a member in array.

```
arr.pop();
```

out :- 1,2,3,4

unshift ⇒ To add a member at starting.

```
arr.unshift(0)
```

out :- 0,1,2,3,4,5

Shift ⇒ To remove a member at starting.

```
arr.shift(0)
```

out :- 2,3,4,5.

Page No.

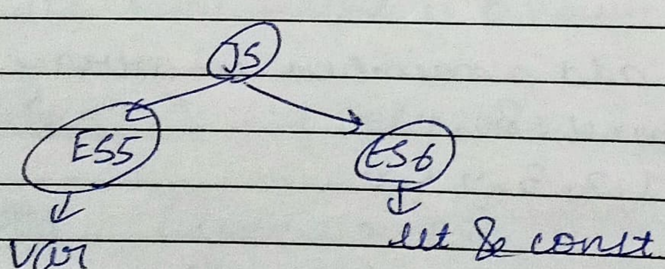
Date ___/___/______

Object:- Ek ke baare me hi saare baate ki jou hua Object.

//1) Blank obj
```
var a = {}
```

//2) Filled obj
```
var a = {
    age=24;
    name: "Note"
    email: "email@gmail.com".
}
```

Difference between var, let & const.



① Var old JS me hai & let-const new me hai.

② Var function scoped hota hai. Yaani, apne parent fun me kahi bhi use ho sakta hai.
```
function abcd(){
    for(var i=0; i<12; i++){
        console.log(i);
    }
    console.log(i); *
}
abcd();
```
out:- 1,2- - - -, 11, (12)

Date __/__/____

Saathi

Aur isi se dekhne ke liye let ⇒

```
funcName(){
    for(let i=0; i<12; i++){
        console.log(i);
    }
    console.log(i);
}
```

out ⇒ 1, 2, - - - - -, 10, 11 and error.

→ Var func. scoped hai baki let, const braces scoped hai.

③ Var adds itself to window object but let const doesn't.

Stack ⇒ works in FIFO principle.
→ Memory where the work is done.

Heap ⇒ → Temporary space where the data is kept for its further operation.

Execution context ⇒ It refers to an imaginary container to run a function where three things are included:-

① Variables.
② functions inside that parent func.
③ Lexical environment ⇒ Yeh ek chart hota hai jisme likha hota hai ki particular function kin cheeje ko access kar sakta hai aur kinko nahi.

Date ___/___/_____

Saathi

How to copy reference values? using spread operator.

```
var a = [1,2,3,4,5];
var b = [...a];
```

Truthy vs falsy?

Falsy? 0, false, undefined, null, NaN, document.all.
Baaki sab truthy hai.

forEach loop? Yeh sirf array pe work karta hai.

```
var a = [12,14,15,16,20]
a.forEach(function(val){
    console.log(val+2);
})
```

Date __/__/____

Saathi

Higher order functions :-

Higher order funtions are the funtions which accepts a funtion in parameter or return a funtion or both.

```
funtion abcd() {

}
abcd(funtion() {

})
```

→ Now the funtion abcd is higher order funtion.
→ Else hum closures banate waqt use karte hain.

Constructor funtions :-

```
funtion saanchaBiscuit() {
    this.width = 12;
    this.height = 20;
    this.color = "brown";
}
```

~~new saanchaBiscuit~~ var bis1 = new saanchaBiscuit()

→ Jab aapke paas aisa koi naksha ho ke aapko ek jaisi prop wali elements banane ho tab we use cons.

Saathi

Date __/__/____

First class functions :

A language is said to have a first class functions when the functions in that language are treated as normal values, you can save them, pass them as arguments to another functions.

New keyword : New keyword always creates a blank object for the constructor element/function.

iife : It is Immediately invoked function expression.

→ Jab hum C) either under function me koi variable banate hain to woh variable private ban jaati hai which can be accessed through "getter"/"name-getter".

```
var ans = (function () {
    var privateval = 10;

    return {
        getter : function () {
            console.log(privateval);
        }
    }
})()
```

Prototype : Prototype is an inbuilt properties provided by browser not like windows but when we create an object and type its name then . (dot) then we see all the prototypes.

Page No.

Global scope ⇒ Jab bhi koi bhi {} brackets ke andar nhi hoti
to hum use global scope bolte hain.

Saathi

Date ___/___/_____

Prototypal inheritance ⇒ Extra property ejisme to an object from prototype is known as prototypal inheritance.

This call apply bind ⇒

This ⇒ It is a special keyword whose values varies place to place.

in global scope. console.log(this) gives window.

in function scope,

```
function abcd(){
    console.log(this);   gives window.
}
```

⇒ NOTE ⇒ Ek function jo object ke andar ho use method bolte hain.

In method scope,

```
var obj = {
    baatkaro: function(){
        console.log(this);
    }
}
obj.baatkaro();
```

gives Object obj.

Date ___/___/___

→ call apply bind ka motive ek hi hai i.e. ⇒ agar tumhare pass koi function hai & koi object hai & tumhe function chalana hai us object pr. aur by-default jo this ki value window hai use window naa rakhkar point karwana hai kisi obj ki taraf.

calls

```
function abcd() {
        console.log(this);
    }
~~abcd.call~~ ~~(obj)~~ ~~all~~
var obj = {age: 24}

abcd.call(obj)
```

— Now this will be obj.

apply:

apply kehta hai ki aap sirf do values hi pass karwa sakte hain aur, >2 ke liye you should use array.

```
function abcd (val, val1, val2) {
            console.log(this, val, val1, val2))
    }
var obj = {age: 24}

abcd.apply(obj, [1,2,3])
```

Bind: Yeh turant nahi chalta hai. Matlab, Matlab yeh ki jab event listeners react me kuch click karu tab chale then we use bind.

Page No.

Date ___/___/_____

Saathi

```
function abcd(){
    console.log(this);
}

var obj = {age:24}

var bindedfnc = abcd.bind(obj);
binded fnc();
```

Asynchronous Javascript ⇒

Synchronous ⇒ Ek hi baad ek kaam hoga, yaani jab tak pehla khatam nhi ho jata tb tk dusra suru nhi hoga.

Asynchronous ⇒ Saare kaam ek saath suru aur jiska ans pehle aaye usko answer dede.

How to identify Asynchronous JS ⇒

If we use ⊛ setTimeout, setInterval, promises, fetch, axios, XMLHttpRequest etc, then it will be async.

Callback ⇒ Callback function hota hai aur yeh function async code me answer aane par chalta hai.

⊛ setTimeout ⇒

```
console.log("hey");  → Sync
setTimeout(function(){  Async
    console.log("hey2");
}, 2000)
console.log("hey 3");  → Sync
```

Out ⇒

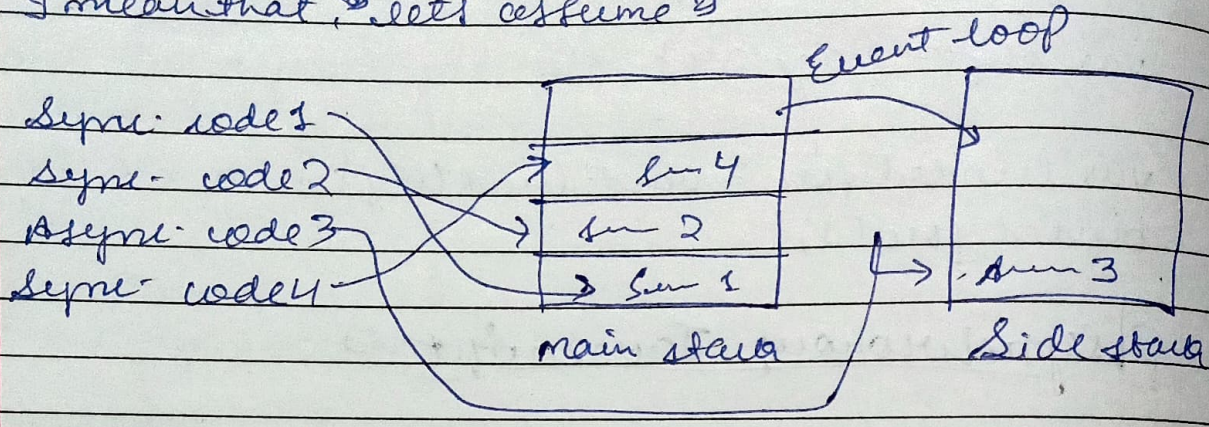
hey
hey 3
hey 2

Date __/__/____

Saathi

Javascript is not asynchronous

I mean that, let's assume



Out: Sync code 1
Sync code 2
Sync code 4
Async code 3.

why?

Async humesha SS me jata hai aur Sync MS me. Pehle sync ke stack one by one execute hote hain aur jab MS khali hota hai fir SS se data liya jata hai.

Add on: Event loop ka yeh kaam hota hai ki when SS se data MS me lekar aata hai.

→ Single threaded hai isliye nhi.

Date __/__/____

Saathi

Promises ⇒ Promises me teen cases aate hain i.e. Pending, Accepted or Rejected.

⇒ Agar ho gaya toh (.then) chalega varna (.catch) chalega.

```
var ans = new Promise((res, rej) => {
    if (true) {
        return res();
    }
    else {
        return rej();
    }
})

ans
    .then(function() {
        console.log("resolve hogaya tha");
    })
    .catch(function() {
        console.log("reject hua tha");
    })
```

Output ⇒ resolve hogaya tha.

Async await ⇒ async await hum (.then) ko remove karne ke liye use karte hain.

```
async function abcd() {
    let raw = await fetch('https://randomuser.me/api
');
    let ans = await raw.json();
    console.log(ans);
}
abcd();
```

Page No.

Date __/__/____

Saathi

Document Object model (DOM) ⇒

4 pillars of Dom ⇒

1) Selection of an element.
2) Changing HTML
3) Changing CSS
4) Event listener.

1) How to select an element ⇒

```
document.querySelector(".tan/#tan")
console.log(.tan/#tan)
```

2) Changing HTML ⇒

```
var a= document.querySelector("h1")
a.innerHTML = "Changed HTML"
        or
document.querySelector("h1").innerHTML = "Changed"
```

3) Changing CSS ⇒

```
var a= document.querySelector("h1")
a.style.backgroundColor = "black"
a.style.color = "red"
```

4) Event listener ⇒

```
var a = document.querySelector("h1")
a.addEventListener("click", function(){
    a.innerHTML= "Changed"
    a.style.color = "yellow"
})
```

Date ___/___/_____

Saathi

Ques 17 WAP to turn bulb on & off from some duration.

Multiple events :-

```
<h1> Hello 1 <h1>
<h1> Hello 2 <h1>
<h1> Hello 3 <h1>
```

JS :-

```
var h = document.querySelector("h1")
console.log(h)
```

It will only print Hello 1. but if we have to print all of them then we use :-

```
var h = document.querySelectorAll("h1")
h.forEach(function(e){
    console.log(e)
})
```

Now it will print all hello 1, 2 & 3.

textContent :- textContent is used to change the content of the HTML file.

```
var a = document.querySelector("h1")
a.textContent = "<div> Hello </div>"
```

Output :- Hello