Advance Crash Course

JavaScript

Learn What Matters

Part:-02

- What are the javascript main version?
- There are Two Types of Javascript version are has follows:-
  - ES5. (var)
  - ES6. (let, Const)

- The Difference:- Var & let & Const. ☑

  - Var old js mein tha
  - let const new js mein hai
  - Var function scoped hota hai
  - let braces scoped hota hai

  ⇒ Var apne parent fnction mein Kabhi bhi use ho sakta hai.

  For eg:-

```
function abcd() {
    for (var i=1; i<12; i++) {
        Console.log(i); // Print 1 to 11
    }
    Console.log(i); // Print 12.
}
```

Var function scoped hota hai

For Khtam to bhi nichey Ki line chalegi Kyuki Var Var function mein Kabhi bhi use ho skta hai

पृष्ठ सं.:
दि.: / /

```
abcd();
```

Result:- 1 9
2 10
3 11
4 12
5
6
7
8

Let :- braces scoped hota hai
For eg:-

```
function abcd() {
  for ( let i=1; i<12; i++) {
    console.log (i);
  }
  console.log (i);
}
abcd();
```

Let braces to speed hai the braces hi andar hi Rahega

Result:- Print 1 to 11

} shows error bcz i is not defined

Result:-
1 7
2 8
3 9
4 10
5 11
6

i is not defined:- Error.

- Window object ☑
  - Var adds itself to the window object.
  - let, const doesn't adds.

Note:- Js language mein kuchh cheeje nahi hai jo hum use kar sakte hai aur wo, cheeje hum js se nahi balki browser se milti hai, aise saare features jo js ka part nahi hai but fir bhi use kar sakte hai onhe hum dhoond sakte hai ek particular object mein jiska naam hai window.

For eg:-

```
Var a = 12;
let b = 12;
```

Now, Go to browser and Open console and Type window see on it u will see a=12 is add on it bcz let adds itself to window & b=12 will not add & u will not see b=12 on window bcz let & const doesn't add to the window object.

js mein kai saare features hai par kuchh features. jo hum use karte hai wo features wo nahi hai par fir bhi use kar paate hai kyoki wo features js language use kar leti hai window se, aur window hai ek box of features given by browser to use with js.

For eg:- alert, prompt, console, hai js ka part nahi hai
For eg:- for, if, else, while, array, function part hai js ka.

Abhi janne ke liye ki js ke konse part nahi hai Open Console on browser and type Window That's all you will see there which are not a part of js.

- Browser context api.

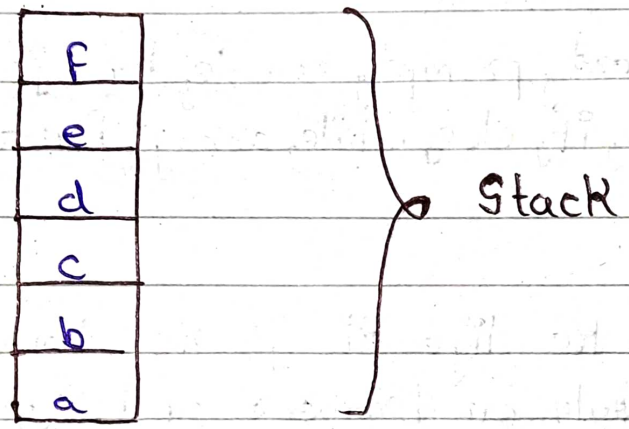
=> Browser tum ko feature deta hai or voh Window ke andr hai.
Browser mainly teen chijaye deta hai or voh Browser context api khelati hai.

- Window
- Stack
- Heap Memory.

Browser context api ke andr hi include hota hai window object. (window :- learn before gives features).

Stack :-

Chalo hum khate hai a, b, c, d, e, f ko chalo



To jis order mai aate hai usi order mai bhar jaate hai.

Heap memory :-

jitne bhi variables ya data hum baanaate hai unhe store kahi to karna padta hai uske liye hota hai heap memory.

For eg :-

1 + 2 + 3 + 4 + 5 + 6. (=21) aap answer do ge)
1+2 (3) (Computer add karega)
3+3 (6) (Computer add karega)
6+4 (10) & 15+6 (21)
10+5 (15) (Computer add karega).

ek calculation ko karte vakt jo bich ka intermiadat data hai jo jarurat hogi aage par final answer nahi hai.

1+2 ka result 3 toh kahi to store hogi na toh kaha ho rahi hai heap memory mai.

OR Final Answer aapko print hote hue dikh jaye ga that's it.

- Execution Context:-

Execution Context matlab jab bhi hum function chalaayege fnc apna ek khudka ek imaginary container bana lega jismein uski teen cheeje hogi:-

- Variable
- Functions inside that parent function.
- lexical environment of that function.

ye jo container hai imaginary ise hi hum execution Context kehte hai.

Chalo man lo ek function chalaya abcd()
function chaltey hai phele chij Execution Context (imaginary daaba (box)). usme kya hota hai.

- Variables.
- functions inside that parent function =>

(Yaani ki aapne function ke andr or ek function banaya hai).

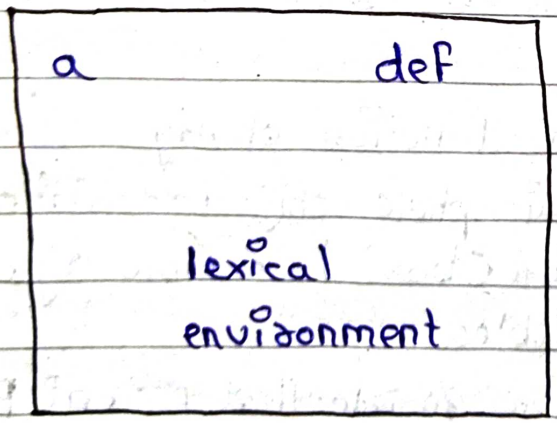
- lexical environment =>

(Yeh aapko baata hai kin cheejo ko use (access) kar shakte hai or kin cheejo ko nahi

for example:-

```
function abcd() {
  var a=12;
  function def() {
    var b=12;
  }
}
```

Aab mano ke maine abcd() ko call karaya toh ek daaba ban jaaye ga. Us daabe mai kya hoga => a, def, lexical environment
Yeh lexical environment baataa hai function kinee access kar shakta or kihey nahi. (Explaination Next)



=> daabe ka naam abcd - Execution Context.

Aapne function mai b use Kiya to voh nahi Chalega Kyu? Kyu? Kyu?
Function apni body Ko use Kar shakta hai or usKe aandr or Function ho toh use Kar shakta hai Lekin Kisi or Function Ke andr Ke element Ko Use nahi Kar shakta.

C Var Ka scoped nearest parent tak toh var b Ka scoped Kaha tak hoga (def) tak to uuse bhar use nahi Kar shakte Vahi to lexical environment Ka Kaam hai Ke aap use nahi Kar shakte.

Toh aapke pass a, def, lexical environment tha or usKi puri chij Ko Kya Khetey hai <u>Execution Context</u>

<u>Defination of Execution Context</u>:-

Execution context is a container where the function's code is executed and it's created whenever a function is called, it contains 3 things, Variables, functions and lexical environment.

Lexical environment :-

lexical environment hota hai ek chart jisme ye likha hota hai ke aapka particular function ki cheejo ko access kar sakta hai and kinko nahi, matlab ki it hold it's Scope and Scope chain.

- **How to copy Reference values :-**

```
Var a = [1, 2, 3, 4, 5];
Var b = [...a];  ( Yeh 3 dot i.e Spread operator ).
```

(Note :- aap 4, 5 dot nahi dal shakte yeh 3 dot is mai khuch kheylate hai or voh hai Spread operator)

```
b.pop();
```

Result :-

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

Toh abhi shami agaya ke array ko copy kese karte hai.

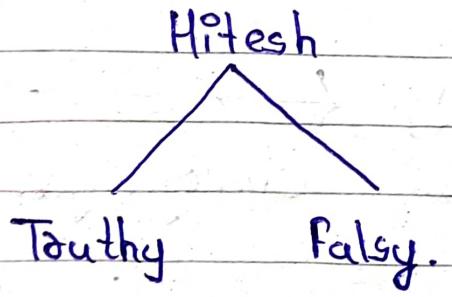
```
Var obj = { name:"harsh" };
Var Copyobj = {...obj};
```

- **Truthy or Falsy :-**

```
if (-1)
```
// It will work Question is How?

javascript mein kuchh bhi likho wo mainly do prakaar mein se kisi ek prakaar ke belong karti hai.

Homne likh diya Hitesh yeh do ko belong karegi that is :-



(Note :- Yeh truthy / Falsy, True / False se alag hai).

Falsy :- 0 False undefined null NaN document.all Truthy :- Truthy values Saari.

```
if (7) {   // Truthy hai or chalega Kyuki yeh value
    Console.log("Hi");   Kahi bhi Falsy se nahi
}                        hai.
else {
  Console.log("Hey");
}
```

Result:-

Hi

Example:- 2

```
if (0) {   // Falsy mai ata hai 0 (Zero).
    Console.log("Hello");
}
else {
    Console.log("Hi");
}
```

chalega ga.

Result:-

Hi

For each :-

For each sirf array pe chalta hai matlab ki jab bhi tumhaare paas ek array hoy tab use mein kaun aata hai for each loop.

For example :-

```
Var a [1, 2, 3, 4, 10]

a.forEach (Function (val) {
    Console.log (val+2);
})
```

// ek ek value hai val mai ayegi or usmai 2 add karayege.

2 add

Result :-

3
4
5
6
12

Bar Bar chalega or Console.log hoga.

Kya yeh main a main change hua hai ? Nahi!

For each kabhi bhi by default aapke array mein change nahi karta wo aapko changes karke deta

→

hai array Ki temporary copy par jiska wajah Se array humesha Same rehta hai.

For ex :-

```
Var a = [1, 2, 3, 4];
a.For Each ( Function (val) {
    Val+2;    // a Ka Copy aaya hai
});.
```

Toh a mai Koy Reflect nahi Hoga.

Result :-

a

a = [1, 2, 3, 4];

- **For in loop** :-

objects par loop Karne Ke liye hota hai For in loop.

```
Var obj = {
    name: "harsh";
    age: 24,
    city: "bhopal"
}

For (var Key in obj ) {
    Console.log (Key);
}
```

// Phele Variable banao phir in bhadme object Ka naam obj

Result:-

name

age

city

Example:- 2

```
Var obj = {
    name: "harsh"
    age: 24,
    city: "bhopall"

For (var Key in obj ) {
    Console.log (obj[Key]); //object Ke andar Key.
}
```

Result:-

harsh

24

bhopall.

Toh dono chaye toh. ?

```
Console.log (Key, obj[Key]); //Key se print hoga name,
                                age, city OR obj[Key]
                                se print hoga harsh,
                                24, bhopall.
```

Result:-

Name harsh

age 24

city bhopall.

- Do-while:-

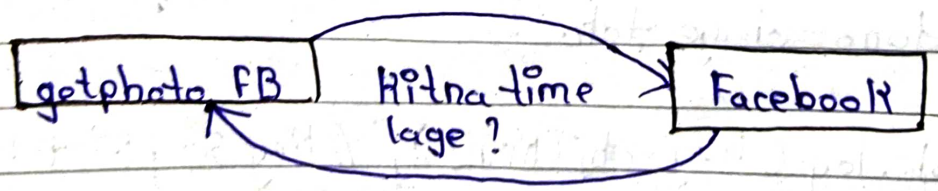
```
Var a=12;

do {
  Console.log("hey");
  a++;
}
while (a<15);
```

Result:-

③ Hey.

- <u>Callback Functions</u> :-

jab bhi koi aisa code jo baad me chalta hai aap likhoge, kyuki wo code baad mein chalta hai js ko ye pata nahi hota ke wo complete hua ya nahi, aise code ke completion par js ko btaya jaata hai ke wo Complete hogya aur aap use chala sakte ho, ye bataane ka kaam Callback ka hai.



Button (Get a photo from Facebook).

Kitna time lagega facebook se photo ko lanemai
Kuch time lagega par Kitna? Koy nahi bata sakta.

Jab bhi ap aasha code likh te ho jo Kahi
Ja Kar Kuch le Kar aayega Kahi bar (Third party).
hume nahi pata Kab tak le Kar ayega.
Voh bata ne Ka Kam hai <u>Callback</u> <u>Functions</u> Ka.

for example:-

aschronys Ka Par thai jo hum baad mai padege

```
Set Timeout( function () {
    Console.log("2 second baad chala");
},2000) // 2000 mili sec i.e 2s.
```

Callback Function.

Result:- 1,2 → Display hoga
2 Second baad chala.

aisa code jo baad me chalta hai use hum ek function dedete hai Ke bhaiya jab complete hojaana to ye function chala dena, aur wo function jo hum dete hai wo ek normal function hi hota hai aur use Kahte hai callback function.

- **First class Functions :-**

Js main ek concept hai jiska matlab hota hai ki aap function ko use kar sakte ho as a value

For example:-

```
Var a= Function(){};
```

Kya mai a main True, undefined, False, 1c, save kar sakte hai toh ha bilkul.

Kya hum a main function save kar sakta hu ?
ha! i.e `Var a = Function (){};`

Kyuki js mai functions hai first class functions. aap onhai normal value ki tarah trit kar sakte hai.

For ex:-

```
function abcd(a) {   a();
}
abcd( function () { Console.log("Hey"); })
```

Result:-

Hey.

- **How arrays are made behind the Scenes :-**

Var arr = [1, 2, 3, 4]

Result :- Typeof [ ]
'object'

Yeh array nay hai ? Nahi! It is object Here's the Proof :-

```
arr = {
    0: 1,
    1: 2,
    2: 3,
    3: 4,
}
```

JavaScript Convert Karti hai.

arr[-1] = 2;

Result :- arr
[1, 2, 3, 4, -1: 2] // Proof

Typeof [ ] => object
Typeof { } => object

Dono hai object hai.

To apko kese pata chalega konsa array hai
let's see:-

Array, then True

```
Array.isArray([])
True
```

object, then False.

```
Array.isArray({})
False
```

Yeh tariqa hai pata karneka ke kab humare pass Array hai or kab humare pass object hai.

- **How to delete object properties:-**

```
Var a = {
    name: "Harsh"
    age: 24
}

delete a.age;
delete a.name;
```

} Delete All Properties.

Result:- a{}
// blank //

End of

JavaScript Advance Crash Course.