Day21(functions)

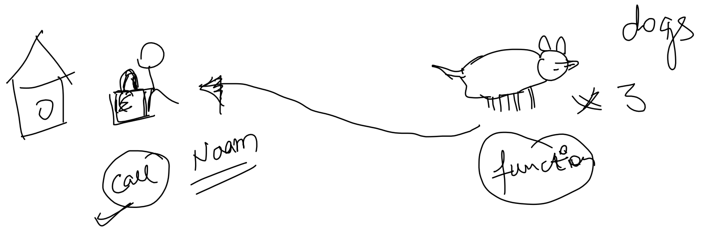
Friday, 31 March 2023 7:45 PM

functions

** → functions are heart of JS ♥.

→ program , Kuch Kaam Karva raha hai
Jab hum usse , Kaune ko bole .

fn - - - - -] program ✓



- * ✓ func^n declaration
- * ✓ func^n calling
- * ✓ parameterised func^n
- * func^{n}al express
- * 1st class funcs
- * Anonymous func^n
- * higher order func^n
- * callback func^n

define : function name_func^n () {   (Naam)

- - - - - work to do

}

let num1 = 10
num2 = 20
let num3 = 30 — X ✓

num3 = 30

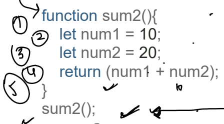
```
function sum(num3){
    let num1 = 10;
    let num2 = 20;
    console.log(n1 + n2 + n3);
}
```

?

sum() → X

sum(30); ← "auguments"

func^n → Kaam → return



```
function sum2(){
    let num1 = 10;
    let num2 = 20;
    return (num1 + num2);
}
sum2();
```

① ② ③ ④ ⑤

10 20

30

num 1

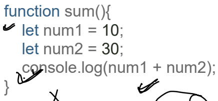
nu 2

clg(Sum2()).

console

let nu

40



```
function sum(){
    let num1 = 10;
    let num2 = 30;
    console.log(num1 + num2);
}
```

X

→ return X

JS → undefined

sum()

clg(Sum(),)

undefined

40

undef.

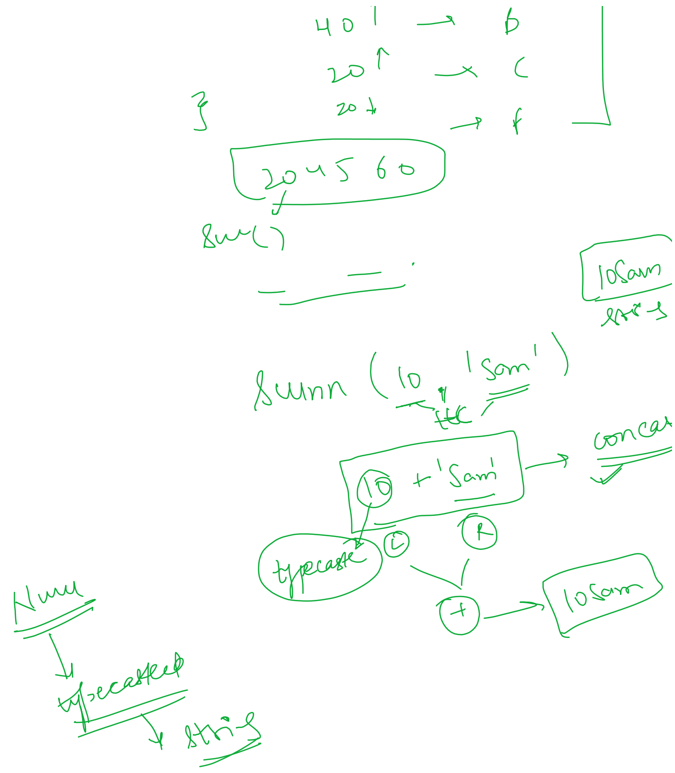
Default

```
sam2( num1 , num2 = 50 )
    clg (num1);
    clg (num2);

}

sam2( 40);
```

sam 2 (

HW grade calculator

```
func^n() {
    80 ↑  →  O
    60 ↑  →  A
```



Imp

Interview

Whenever i store the entire function inside a variable then it is called :

1. functional expression ✓
2. first class function ✓
3. first class citizen ✓
4. All of the above ✓
5. None

yes

Let → let ans = 10; ✓

No

3 var = let = const

ways

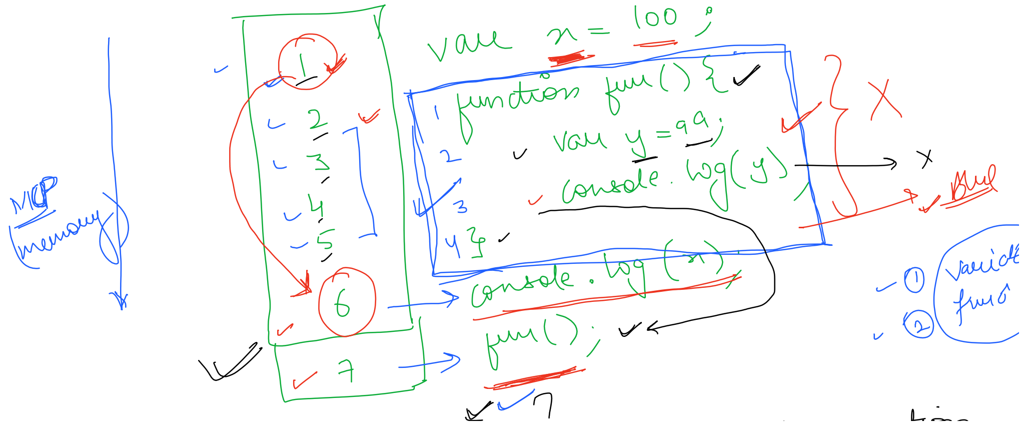
var naam = 10;

let naaam = 200;

const naam = 300;

✓

dec. variable

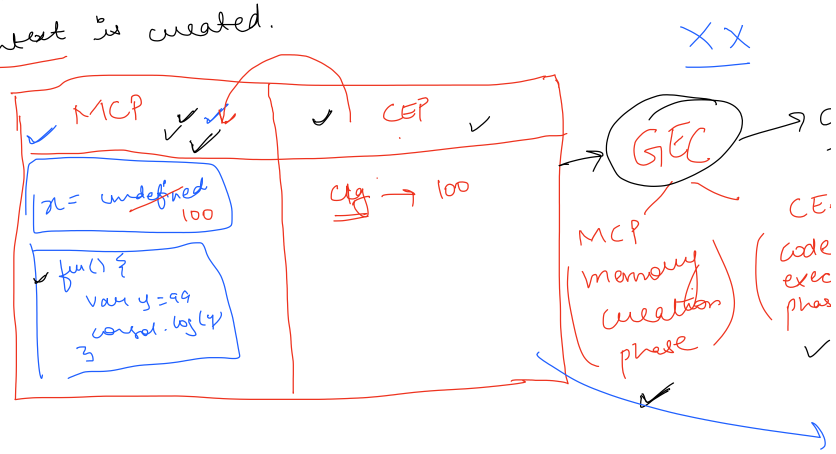


```
var x = 100;
function fun(){
    var y=99;
    console.log(y)
}
console.log(x)
fun();
```

Satya

Whenever a JS code is run a global execution context is created.

Whenever a JS code is run a global execution context is created.



Whenever a Global Execution Context is created along with that a Global object is also created.

MCP → before running the code (pehle) variables, func's → memory

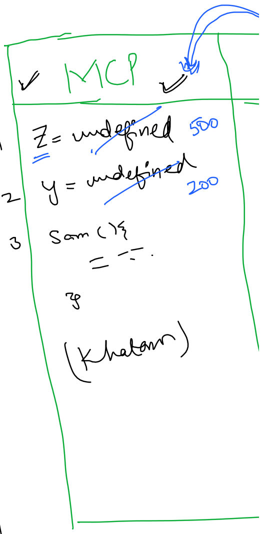
CEP → 1-1 karke ab aapka code run hoga.

MCP => Inside the MCP all the variables and function they all get the memory separately in MCP.

CEP => In CEP the code runs line by line one after the other.

———X———X———X———X——

```
①  Let z = 500;
②  Let y = 200;
③  function Sam(){
④     Let g = 10;
⑤     console.log(g);
⑥  }
⑦  console.log(y);
⑧  sam();
⑨  console.log(z)
```



Whenever a function is called a new execution context is created with the name of the function.

Memory Creation Phase m jitne bhi variables hai unhe initialize kiya jayega default value undefined ke sath.

Aur function ko pura body ke sath memory dedi jayegi.Aur jab bhi function call hoga tab ek naya GEC banega function ke naam se .