Day21(functions)

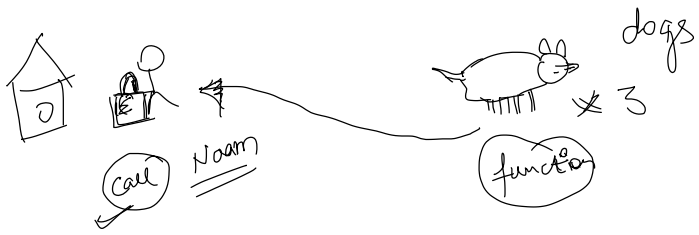
Friday, 31 March 2023 7:45 PM

functions

** → functions are heart of JS .

→ program , kuch kaam karwa raha hai jab hum usse , kaune ko bole .

fn ----] program



* ✓ func" declaration
* ✓ func" calling
* ✓ parameterised func"
* func"al expres
* 1st class funcs
* Anonymous func"
* higher ordere func"
* callback func"

define : function Naam name_func"() {

---- wowk todo

}

let num1 = 10

num2 = 20

let num3 = 30 — X ✓

arg

num3 = 30

function sum(num3) {

let num1 = 10;

let num2 = 20;

console.log(n1 + n2 + n3); ?

}

sum() → X

sum(30); ← "arguments"

func" → kaam → return

let nu (40)

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

30

num1
num2

clg(sum2()).

console

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

→ return X

JS → undefined

sum()

clg(sum())

undefined

HW

undef.

default

```
sam2( num1 , num2 = 50 ) {
    clg(num1);
    clg(num2);
}
sam2(40);
```

param.

avg.

sam2(40, 60)

overriding

avg. → Default parameter

HW grade calculator

```
func() {
    80 ↑ → O
    60 ↑ → A
    40 ↑ → b
    20 ↑ → c
    20 ↓ → f
}
```

20 45 60

sum()

10sam

string

sum(10, 'sam')

10 + 'sam' → concat

typecast

L

R

+ → 10sam

Num → typecasted → string

Imp

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

Interview

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

Let → let ans = 10;

No

ways

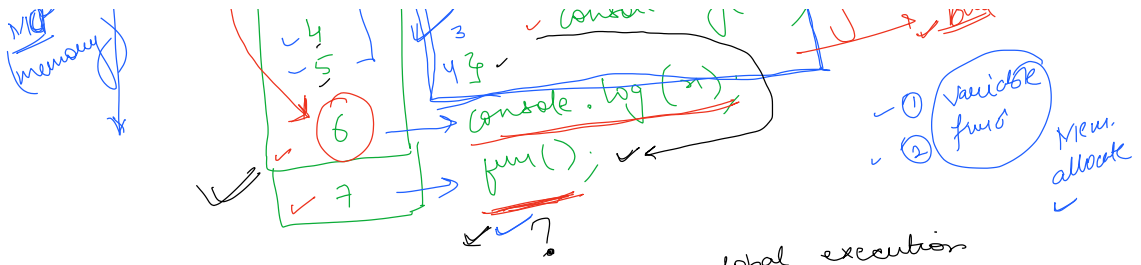
dec variable

(3) var / let / const

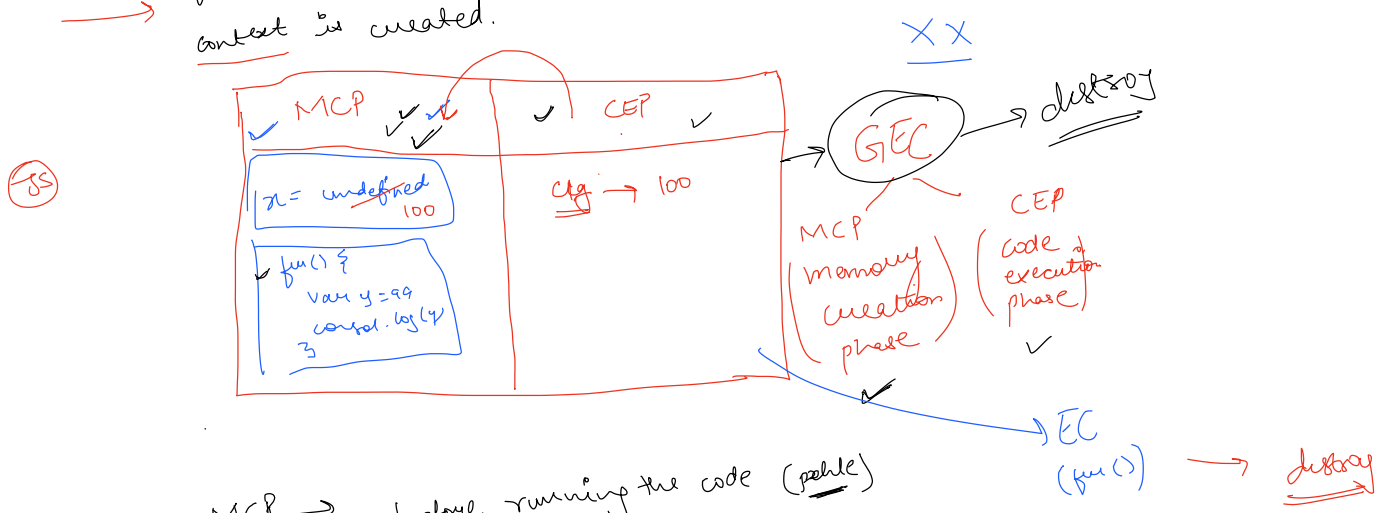
var naam = 10;
naaam = 200;
naam = 300;

```
var x = 100;
function fun() {
    var y = 99;
    clg(y)
}
```

X



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



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

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