# Java Execution Context

- JS runs your program in two phase

{} → Global EC
JS code
↳ this

in Browser (this = Window object)
[this ke andar GEC rakh diya jata hai

{}
JS have 3 EC (2 mainly)
↳ Global EC
↳ function EC
↳ Eval EC

- JS is single threded
- The two phases are
  → Memory Creation Phase (allocate Mem)
  → Execution Phase

eg

```
let vall = 10
let Val 2 = 3

function add Num (Num1, Num2) {
        let total = num1 + num 2
        return total
};

let result1 = add Num (Val1, Val2)
let result2 = addNum (10, 2)
```

① → runs thru. → Global Exeation
↑ allocate at this

② Memory Phase
vall → undefined, val2 → ud, ..
addNum → defination
result1 → ud, result2 → ud

③ Execution Phase
vall ← 10, val2 ← 3, [func addNum → kuch nahi hoga]
But when result1 call hoga toh

addNum → new Variable Environment + Execution thread
new execution of contex called

{ jitni Baar func^n call honge utni baar yeh Box Create hoga } as ③ Execution Phase

val1 ← 10 , val 2 ← 3

add Num → new var enviro + Exe thred [GEC1]

→ (3.1) Memory Phase

val1 → undefined

val2 → "

total → "

→ (3.2) Execution Context

num1 ← 10

num2 ← 3

total ← 13

return to (1)

this total is return to its parent executional context

[Bola jata hai ki Global execution context mein return krta h]

Delete

Now [GEC1] → Deleted

result 1 ← 13

; Now again for result 2

add Num → NVE + ET → M Phase, EC

result 2 ← 12 ;

- Call Stack

```
let func one () {
        func two() (
                func three () {
                }
        }
}
```

in stack it works as

```
| three() | → ①
| two()   | → ②
| one()   | → ③
| global Exe |
```