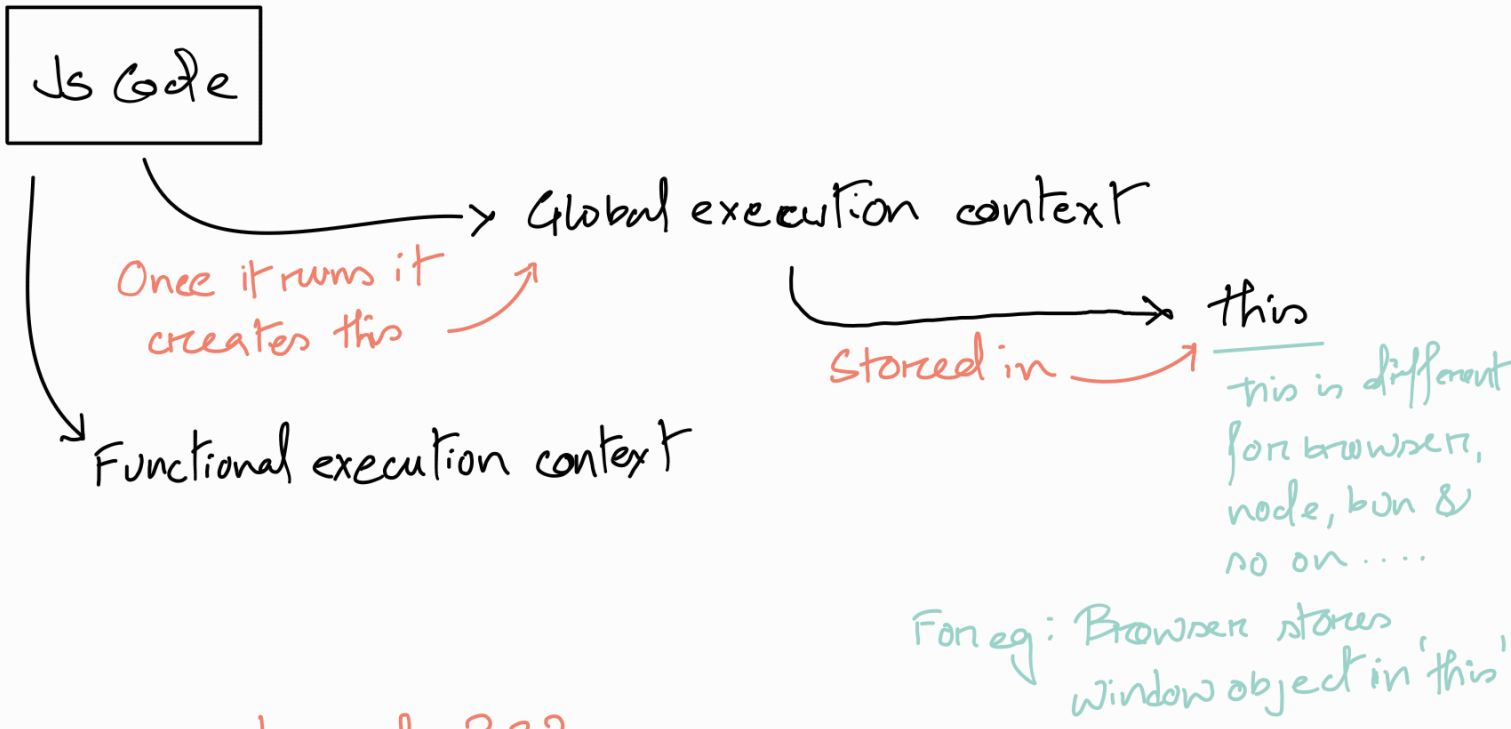


How JS executes code ???

- Memory creation phase
- Execution phase

```
let val1 = 10
let val2 = 20

function addNums(num1, num2){
    total = num1 + num2
    return total
}

let result1 = addNums(val1, val2)
let result2 = addNums(30, 40)
```

Step 1: Global execution → this

Step 2: Memory allocation

val 1 → Undefined
val 2 → Undefined
addNum → func^n defn
result 1 → Undefined
result 2 → Undefined

Step 3: Execution phase

val 1 → 10
val 2 → 20
addNum →
result 1 → 30
result 2 → 70

New variable env + Execution thread

→ This will again make one memory phase & execution phase for every func^n

Memory phase

val1 → undefined
val2 → undefined
total → undefined

Execution phase

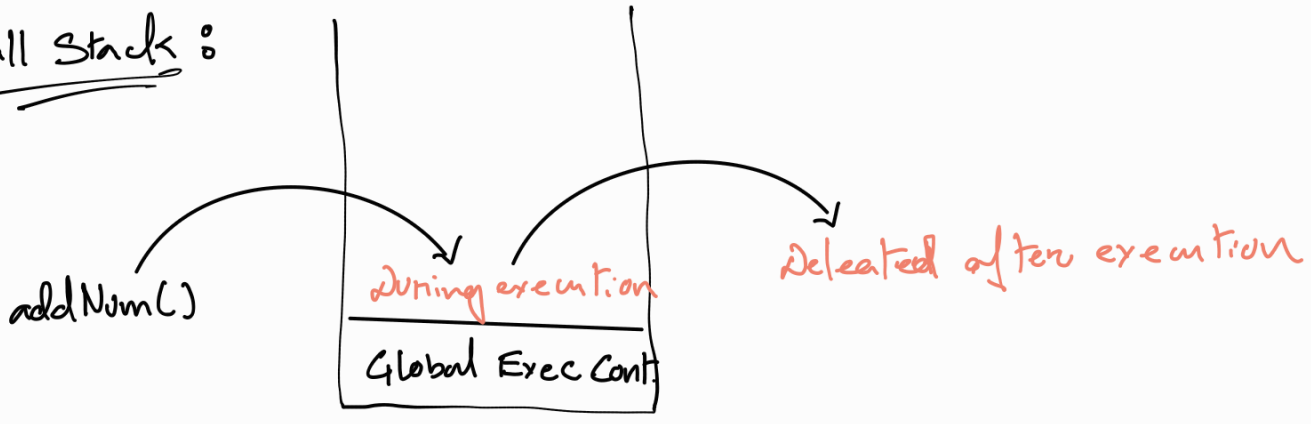
num1 → 10
num2 → 20
total → 30;

this goes back to the global execution phase

AND

→ This box gets deleted & a new box is created for the second call of addNums()

call Stack :



If one function calls another function, then they follow LIFO method in order to get deleted.