Execution Context, Call stack

Janascript Execution Context

$\boxed{\{\}}$ → Global EC
↑ this

browser → window,
node env → { } blank rehta है।

⇒ Janascript is single threaded, इसके अंदर हर चीज एक process होती है।

EC → Global Execution Context
→ Functional Execution Context.
→ Eval Execution Context (Mogoose)
→ (property of Global object).

How code runs

$\boxed{\{\}}$ → Memory Creation Phase {allocation of memory}
→ Execution phase

Steps:

① Global Execution
↓
this (pura code) ←

```
1  let val 1 = 10
2  let val 2 = 5
3  function addNum(num1, num2) {
4  let total = num1 + num2
5  return total
6  }
7  let result1 = addNum(val1, val2)
8  let result2 = addNum(10, 2)
```

② Memory creation phase

val 1 → undefined
val 2 → undefined
addNum → defination
result 1 → undefined
result 2 → undefined

→ first cycle, always होगा

③. Execution phase :-

val 1 ← 10
val 2 ← 5
} upto line 6

add Num →

result 1 = 15

sandbox

new variable env
+
execution thread

add no. के लिए वापस से step ② और step ③ perform होगा

memory phase

val1 → undefined
val2 → undefined
total → undefined

(deleted after its work)

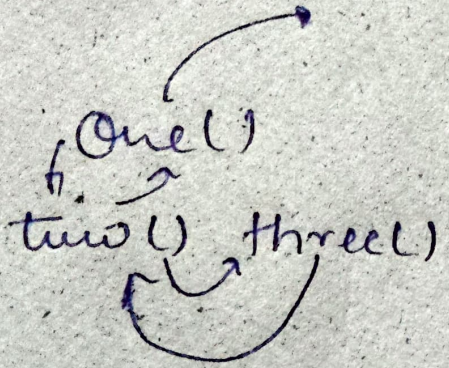
Execution phase

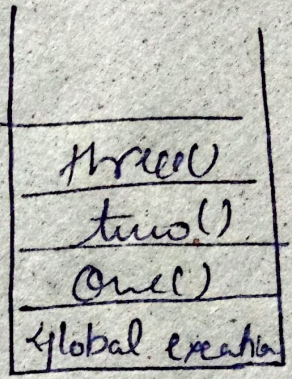
num 1 → 10
num 2 → 5
total → 15.

→ return to Parent execution context

Same repeats for result ②

function Call stack :-





Lifo

Stack