

Javascript Execution Context

when new file is created

{ } → Global Execution Context

← this { } ← this

* node execution context is diff than browser execution context

↳ window ← this obj

* JS → Single threaded

↳ Global execution context

↳ function execution context

↳ Eval execution context

How JS CODE RUNS

{ }

→ memory creation phase / creation phase
→ execution phase

only memory is declared

```

1 let val1 = 10
2 let val2 = 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)

```

① Global Execution ← occur only once

↓
this

② Memory Phase

val1 → Undefined
 val2 → undefined
 addNum → definition
 result1 → undefined
 result2 → undefined

③ Execution Phase

val1 ← 10
 val2 ← 5
 addNum →

result1 = 15

new Variable
 Environment
 +
 Execution thread

→ as fn is called
a new execution
context is created

↓

after work is
deleted

memory phase

val1 → undefined
val2 → undefined
total → undefined

Execution Context

num1 → 10

num2 → 5

total → 15

↳ return to global execution context

line - 8.

result2 = 12

new variable envt

+

thread

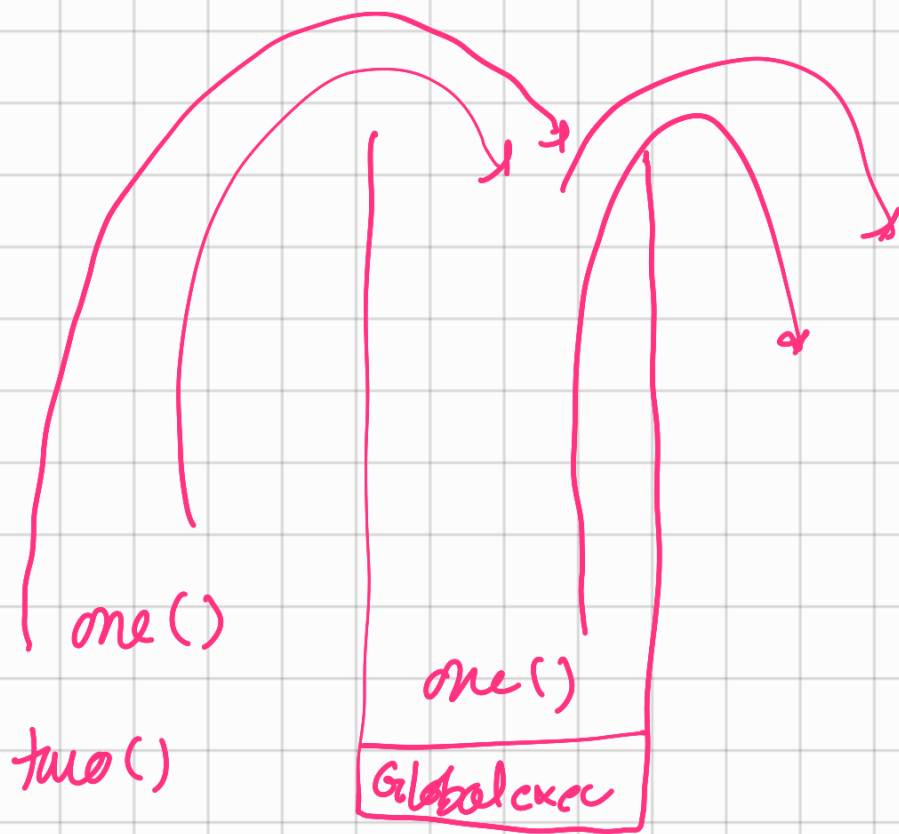
memory phase

val1 = undefined
val2 = undefined
total = undefined

execution phase

num1 = 10
num2 = 2
total = 12

CALL STACK



LIFO (last in first out)

