Single Threaded ← Javascript Execution Context

→ Global Execution Context
→ Function EC
→ Eval EC

code → Global EC ← (this)
↓
(window in browser)

{} → Memory Creation Phase
→ Execution phase

```
let val1 = 10
let val2 = 5
function addNum (num1, num2) {
    let total = num1 + num2
    return total
    }
let result1 = addNum (val1, val2)
let result2 = addNum (10, 2)
```

→ Execution phase

① Global Execution (Environment) → (this) is allocated

② memory phase → only space is given (not value)

val1 → undefined
val2 → undefined
function addNum → definition (refrence)
result1 → undefined
result2 → undefined

First Cycle (It will always happen)

③ Execution phase

val1 ← 10
val2 ← 5
addNum →

For result1

result1 = 15

(context)
New Executional Environment
New variable Environment
+
Execution thread

after execution, it get deleted

Memory phase
val1 → undefined
val2 → undefined
total → undefined

Execution Context
num 1 → 10
num 2 → 5
total → 15

return total
↳ returns the value of total to "global context"

addNum →

For result2

result2 = 12