

episode on

How javascript code is executed, call stack

phase (II)

phase (I) Memory creation phase

code execution phase

variable →

n : undefined

function →

squase : {....}

squase 2 : undefined

← whole code is stored

n : 2 [nothing to do with 2nd line]

Again execution context will be created due to function invoke

Memory component

code component

Again two phases

(I) Memory creation phase

num : undefined

ans : undefined

code execution phase (II)

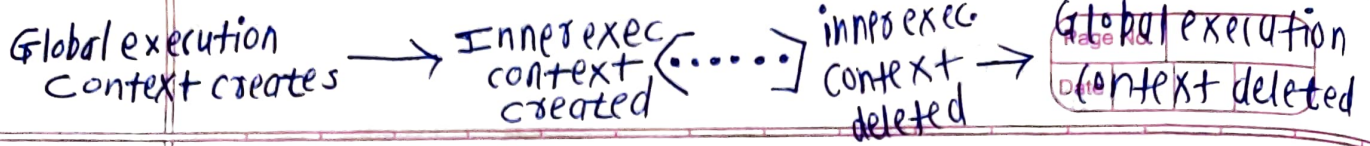
num : 2

ans : calculation i.e.

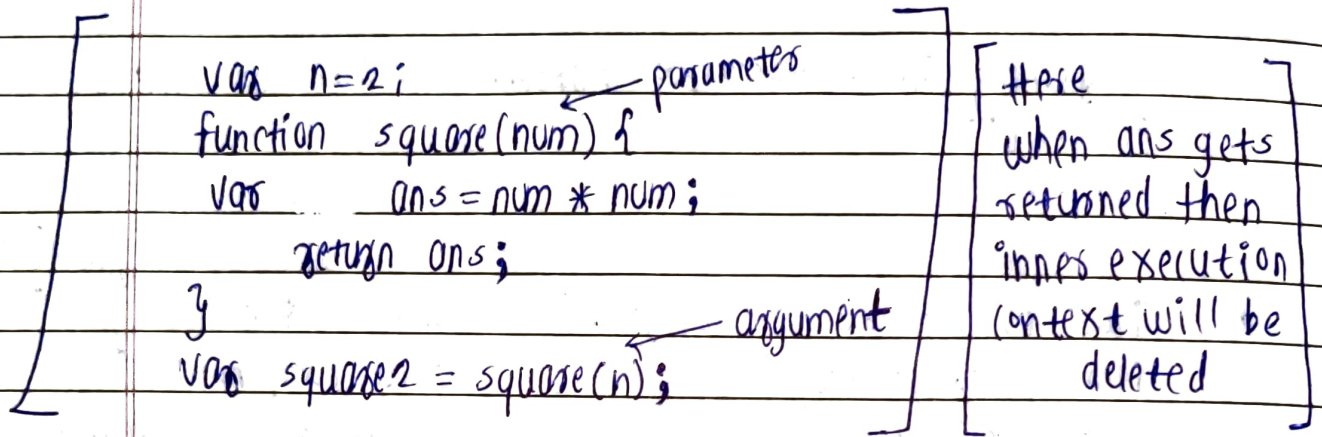
$num \times num = 2 \times 2 = 4$

[we get this value from argument n i.e. 2]

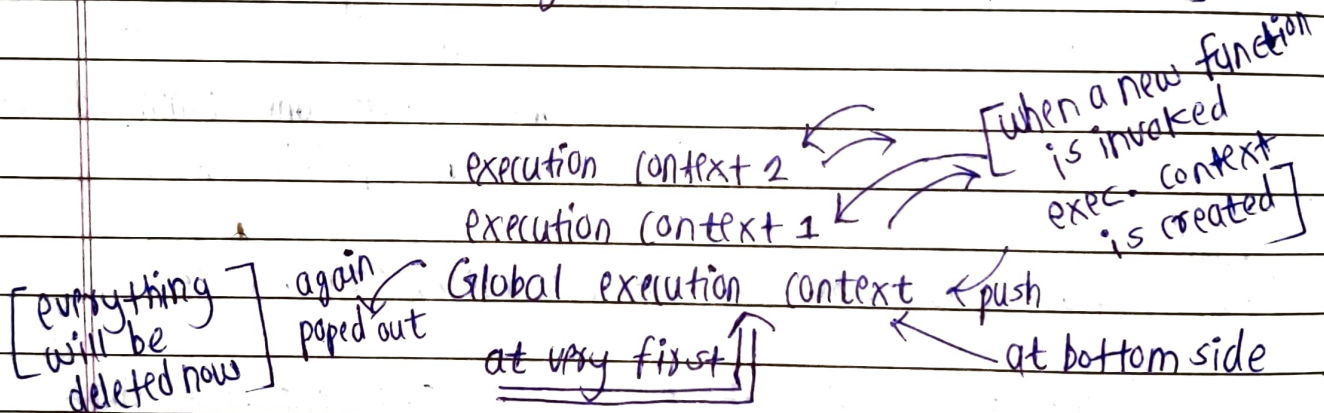
ans : 4



example which we consider in code execution:-



- call stack [all above critical things are managed by callstack]



- at end call stack will be empty.
- call stack maintains the "order of execution" of execution contexts.

- Call stack fancy names :-

- ① execution context stack
- ② program stack
- ③ control stack
- ④ runtime stack
- ⑤ machine stack

All names are call stack itself.

