

Asynchronous Execution → step by step

1

1. `console.log("Starting the process");`
2. `setTimeout(() => {`
3. `console.log("2 sec's timer");`
4. `}, 2000);`
5. `setTimeout(() => {`
6. `console.log("0 sec timer");`
7. `}, 0);`
8. `console.log("finishing the process");`

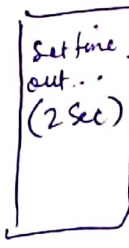
Explanation

Step 3

Call Stack



node api



Event 1.

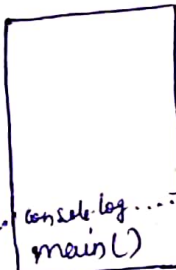
Event loop



Callback queue

Step 1:

Call stack



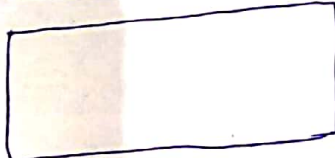
Node API (Event Tracking)



Printed

Event loop

Callback queue:



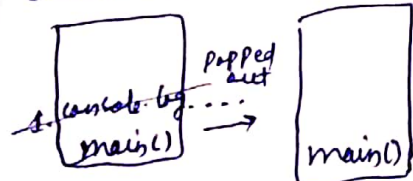
Explanation:-

line 1 pushed to call stack and gets executed as it is non-IO operation, printing

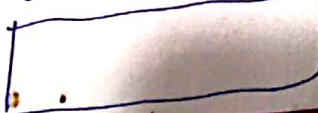
O/P → Starting the process
↓
Executed, so, popped out from call stack

Step 2:

Call stack

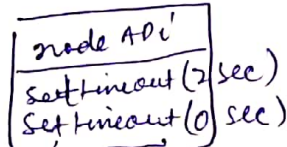


Event loop (callback queue)

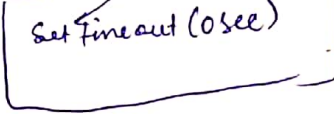


Step 4:

Call stack



Event loop

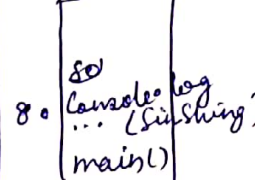


callback queue

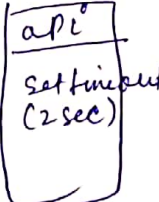
Since call stack is not fully empty, callback queue's event won't be moved to call stack by event loop.

Step 5:

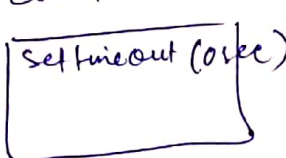
Call stack



node api

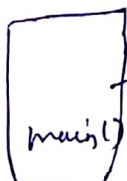


Event loop

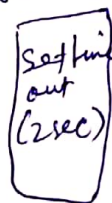


Step 6:

Call stack



node api

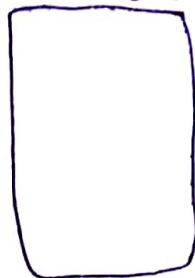


console.log (finishing the process) popped out as it is executed

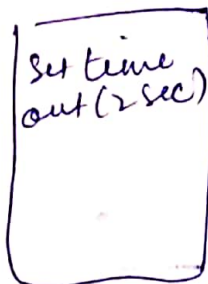
O/P → Starting the process
finishing the process

Step 7

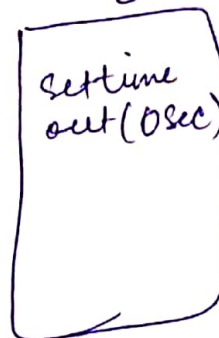
Call stack



node API(Event)



Event loop
callback Queue



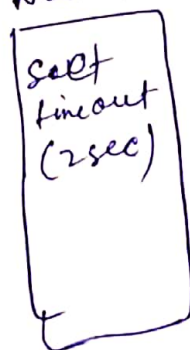
main() pops out from call stack, as program reached to line 8, but process does not end, as it happens in other programming language. Now, Event loop part is starting

Step 8:-

Call stack



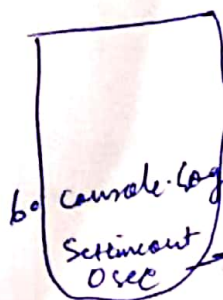
node API(Event)



pushed by Event loop as call stack was emptied completely, Event loop.

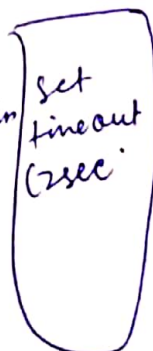
Step 9:-

Call stack

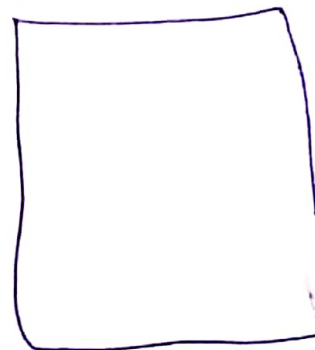


pops out first after execution
pops out after console.log pops out

node API

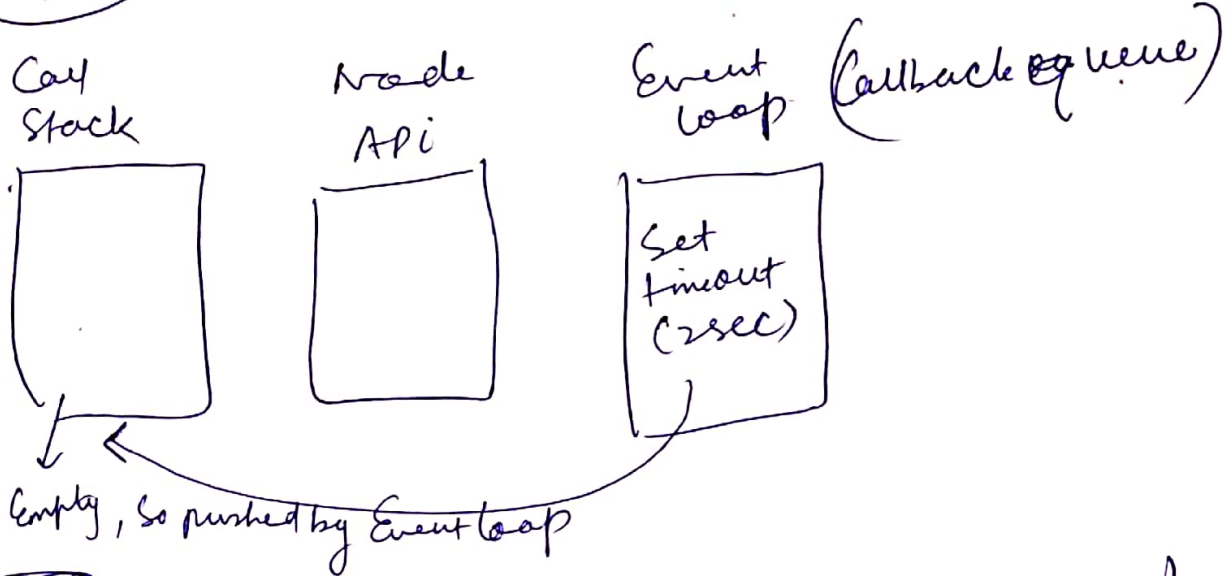


Event loop

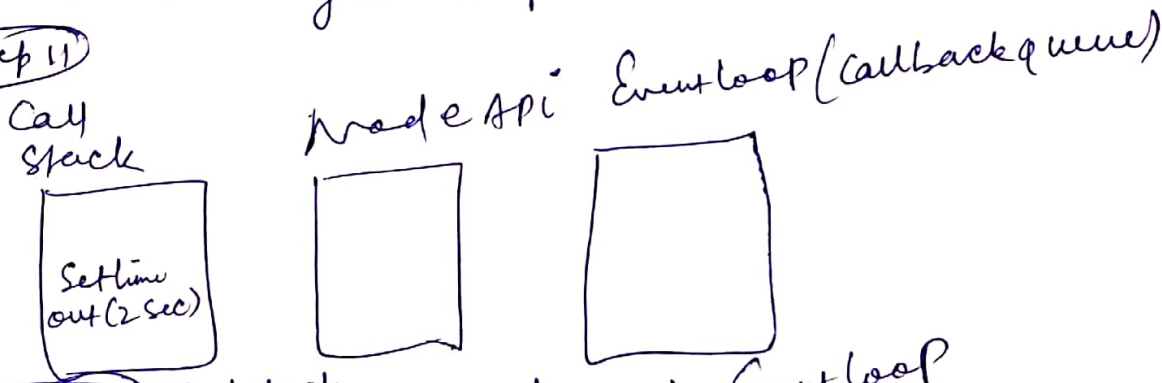


O/P starting the process
finishing the process
0 sec timer

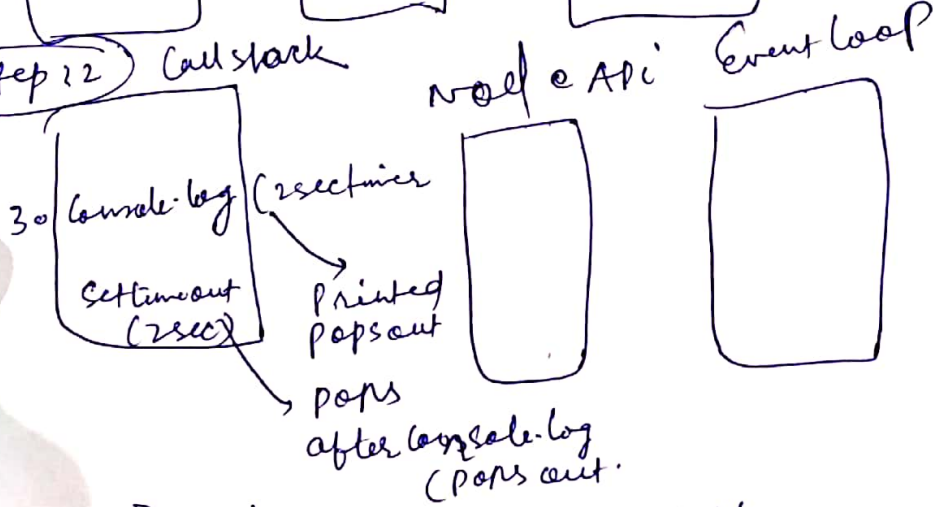
Step 10



Step 11

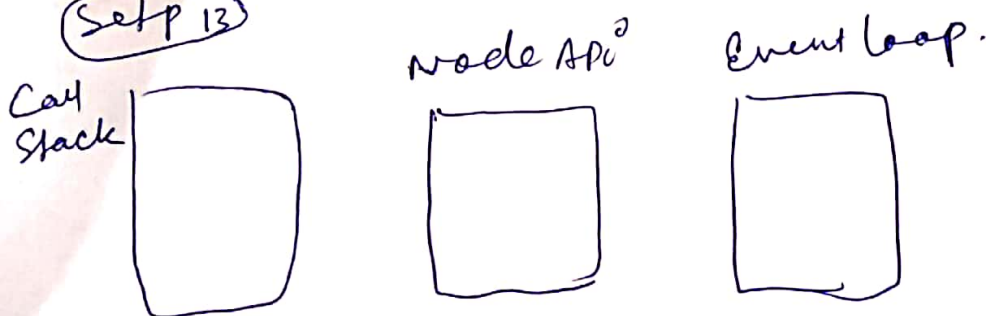


Step 12



Step 13 O/P: Starting the process
finishing the process
0 sec timer
2 sec timer.

Step 13



End of Program.