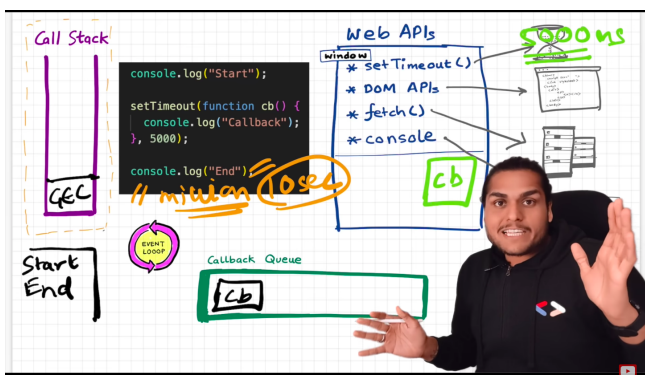


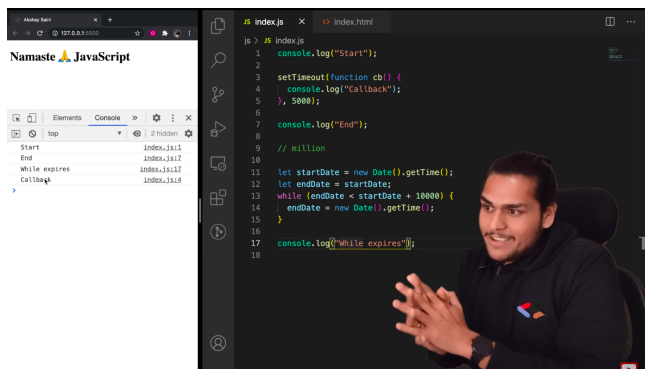
## Trust Issues with setTimeout()

Saturday, 19 August 2023 11:41 PM



- In above code a `fn` is created & pushed onto the callstack. After that 'start' is printed.
- Now JS engine registers the callback `fn` in web API environment & attaches a timer of 5secs to it.
- It goes on executing other lines of code, since JS does not wait for anyone and 'end' is printed.
- Now lets assume that there are 1000 lines of code after 'end' is printed which lets assume take 10secs to execute.
- Now one would think that after 5secs callback `fn`'s reference gets stored in the callback queue and will be pushed to the callstack by event loop immediately after 5secs. But that's not the case and that's why there are trust issues with `setTimeout`.
- Since the callstack is not empty & 1000s of lines of code is being executed, so event loop has to wait & finally after 10secs when 1000s of lines of code are done executing & callstack becomes empty only then, event loop pushes the `fn` of the callback `fn`.
- So, this `setTimeout`'s callback `fn` executes after 10secs instead of 5secs. This is called **concurrency model** in JS.
- That's why it is said that we should not block the main thread.

→ Browser Demo of above explanation:



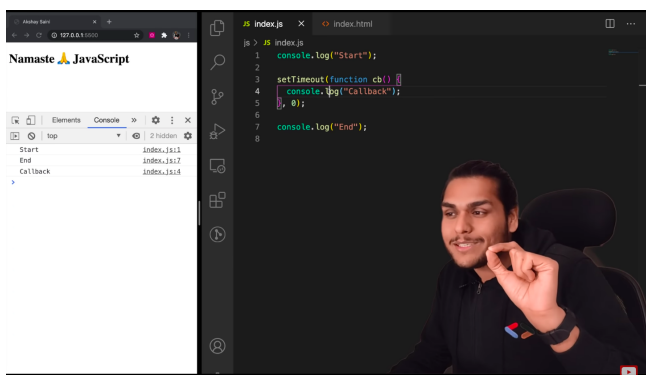
→ So here, line 11 to 15 represent a delay of 10secs.

→ So, JS engine will print 'start', then it'll print 'end' & finally after 10secs it

prints 'while expires' & immediately after that it prints 'callback'.

→ We can explain this behaviour using the above logic.

→ Another example:



→ So here the o/p is such because callback f<sup>n</sup> will have to go through the callback queue and event loop will only push the callback f<sup>n</sup> to the call stack once the call stack becomes empty, hence we get such o/p.

Q: What is the use-case of above code?

ans- We can use above code, if we want to 'defer' a piece of code until other lines of code get executed.

Note: So, due to the concurrency model, we can do asynchronous tasks in a single threaded language like JS.