JS callbacks and event loop

Almas Maksotov

December 2019

1 Introduction

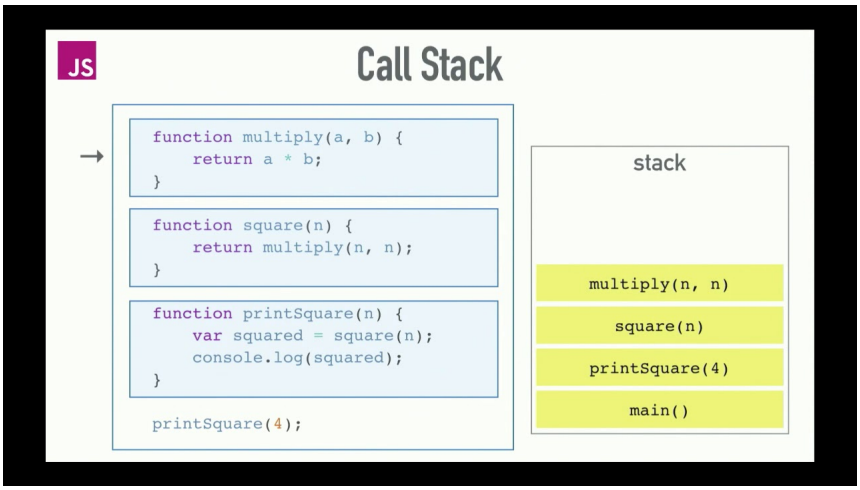
This are the notes for the video about JS event loop. video
Questions that are considered in this file:

1. How does JS works
2. What is callstack?
3. What is callback queue?
4. What is event loop?

JS is a single threaded non-blocking async. concurrent programming language. Lets break this down.

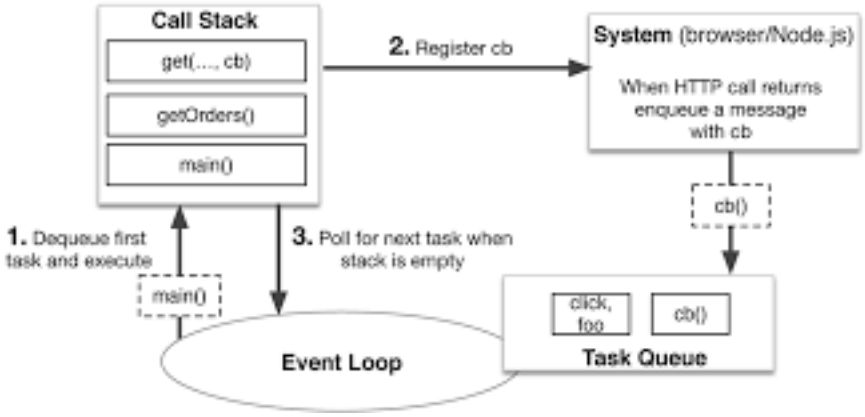
2 What is callstack ?

Lets start with what is callstack. Wiki tell us that it is - **In computer science, a call stack is a stack data structure that stores information about the active subroutines of a computer program. This kind of stack is also known as an execution stack, program stack, control stack, run-time stack, or machine stack, and is often shortened to just "the stack".** - from this we can infer that it is something that is related to execution of program. Let see the picture



The main idea is - any time we see the function call, first we insert it into callstack then our execution start. The function will stay in the stack until PC - program counter hits **return**. After, we will pop the function from the stack. JS has only one call stack. Other languages like - go java ... can have multiple of them and execute program in parallel. It is important to note that the callstack is the only way to execute function, meaning that it can be blocked by function which takes long to end. For example: for loop that makes 1 billion iterations or network call. If we block callstack with one of these operations, no other operations can start until the end. So that's why we need parallel processing in other languages. But is JS we have completely different architecture.

2.1 JS Architecture



So this is what developers of JS came up to solve the issue of blocking. The idea is to submit the high load task to API and keep executing the fast ones, and when API finishes task it informs the JS that task is finished with the help of event loop. Basically any JS environment has such API to submit task. That's why JS is called single threaded non-blocking.

3 What is callback queue?

Remember how when we called some functions we pass other functions as arguments. Those argument functions are called callbacks. And they have separate data structure called callback queue. Usually we pass such callbacks into high load functions that we've talked above, and when it ends API submits its callback into queue.

3.1 Then why do we need event loop?

The event loop has two simple jobs.

1. to track and inform whether call stack is empty
2. to submit task form event loop into call stack if it's empty

4 Conclusion

So this is the reason behind JS smoothness and why it is called non-blocking.