How does JS actually work?

A program allocate memory, parse and execute.

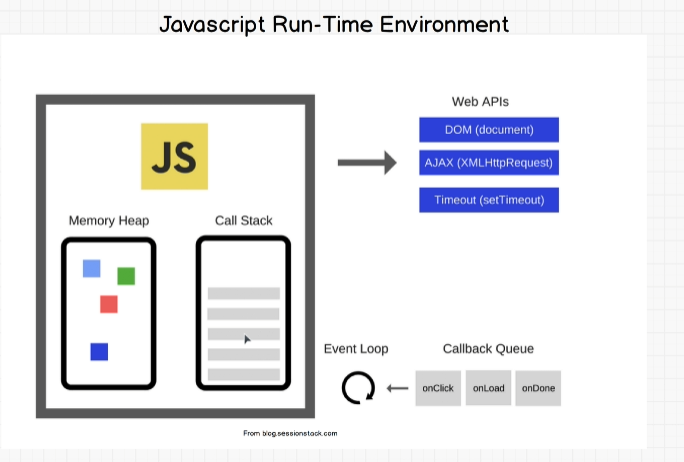
Common terms: memory leak and stack overflow

Memory Heap & Call Stack

Blocking, single threaded. Until we got asynchronous to be non-blocking

We need more than just JS engine.

JS run time environment is part of browser



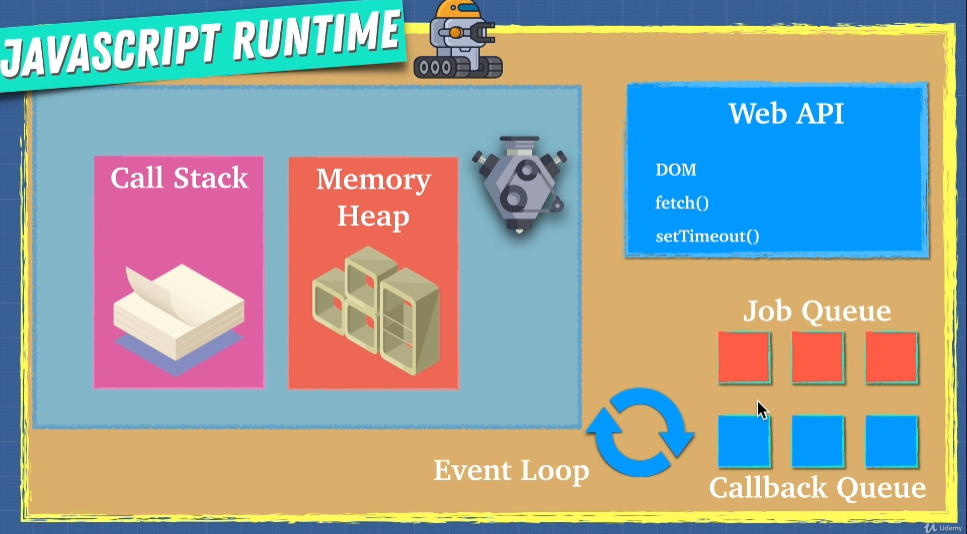
Promises

Promise is an object that may produce some value after some time. Fulfill (Resolved), rejected and pending. Promises is new in ES6

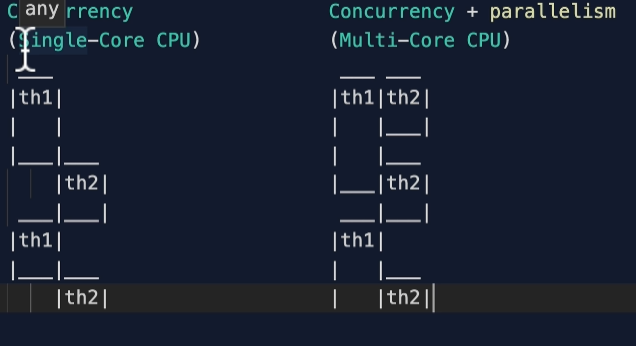
Before promise, we had callback. The syntax can cause callback pyramid of doom, hard to read

Job Queue

After ES6, came another piece of runtime – Job Queue



Threads, Concurrency and Parallelism



In JS we can only wait for callstack finish for those async action. We don’t really have parallelism in JS

But with node, we can achieve that