

Module 5b: Asynchronous JavaScript - Part2

In this Module:

Writing asynchronous code is a super important for Javascript developer and done on daily basis.

- Generators
- Event Loop, thread in JS

Generators in JavaScript - What, Why and How

"Generators are (kind of) pausable functions in JavaScript. Another word for them is co-routines. They are used (among other things) to manage async operations, and play very well with promises. "

<https://www.youtube.com/watch?v=ategZqxHkz4>

Event Loop:

Event Loop explanations can be condensed down this two videos:

First of them is a lighter one, and second is a bit more in-depth. Please watch both of them carefully, as understanding the event loop is essential for JavaScript programmer.

What the heck is the event loop anyway?

"JavaScript programmers like to use words like, "event-loop", "non-blocking", "callback", "asynchronous", "single-threaded" and "concurrency". We say things like "don't block the event loop", "make sure your code runs at 60 frames-per-second", "well of course, it won't work, that function is an asynchronous callback!" If you're anything like me, you nod and agree, as if it's all obvious, even though you don't actually know what the words mean; and yet, finding good explanations of how JavaScript actually works isn't all that easy, so let's learn! With some handy visualisations, and fun hacks, let's get an intuitive understanding of what happens when JavaScript runs."

<https://www.youtube.com/watch?v=8aGhZQkoFbQ>

Jake Archibald: In The Loop - JSConf.Asia 2018

"Have you ever had a bug where things were happening in the wrong order, or particular style changes were being ignored? Ever fixed that bug by wrapping a section of code in a setTimeout? Ever found that fix to be unreliable, and played around with the timeout number until it kinda almost always worked?"

This talk looks at the browser's event loop, the thing that orchestrates the main thread of the browser, which includes JavaScript, events, and rendering. We'll look at the difference between tasks, microtasks, requestAnimationFrame, requestIdleCallback, and where events land. ”

<https://www.youtube.com/watch?v=cCOL7MC4PI0>

Concurrency model and the event loop

*“JavaScript has a concurrency model based on an **event loop**, which is responsible for executing the code, collecting and processing events, and executing queued sub-tasks. This model is quite different from models in other languages like C and Java.”*

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/EventLoop>