**Asynchronous JavaScript - Part 1… -** [Marc Kirk](https://medium.com/@byteslovesbits?source=post_page-----a6419a391b8b--------------------------------) Feb 20, 2022

Let’s go back in time and look at the world’s first web-page. It was relatively dull text. Back then, it was all about sharing static-documents or text-content. If you were a web-developer, life was simple, life was good. However, as time passed, people demanded more from the web than mere static-content. They demanded a dynamic experience! — an experience where the client-server model of computing would make things more expressive and exciting.



Figure 1 — Tim Burners-Lee’s first ever website

Fast forward to the present day and for most people, the web-browser **is the internet**. The web-browser is a ubiquitous, all singing, all dancing window to the world. It is dynamic and exceptionally powerful. It represents decades of software development and refinement.

From the browser’s humble beginnings displaying text, we now use our web-browsers to play games, edit videos, track the economy, feed the cat, crash a drone, monitor our health and facetime people. The list is endless and we do all this ‘stuff’ asynchronously!

***Asynchronous Definition:****processes that operate independently.*

Being asynchronous is just a fancy way to say we can do multiple things at the same time. Think of watching your favourite movie. You don’t have to watch the video, then listen to the audio. We can watch and listen at the same time. In essence, you have video and audio processes that combine to produce a better experience.

**JavaScript — the Programming Language of the Web**

Of all the programming languages — it is JavaScript that has cemented its legacy as the programming language of the web. JavaScript has evolved from a relatively toy language to become an extremely powerful, expressive, multi-paradigm language that dominates the client-side. Also, it is making significant progress on the server-side through **node.js**. If you need one language to do most things, the argument for JavaScript is strong.

JavaScript is baked into all browsers through various engines such as Chrome’s V8 and Firefox’s SpiderMonkey. If you would like to quickly experiment with JavaScript, go to your browser’s developer tools where you will find a REPL (Read Evaluate Print Loop) within the console section.

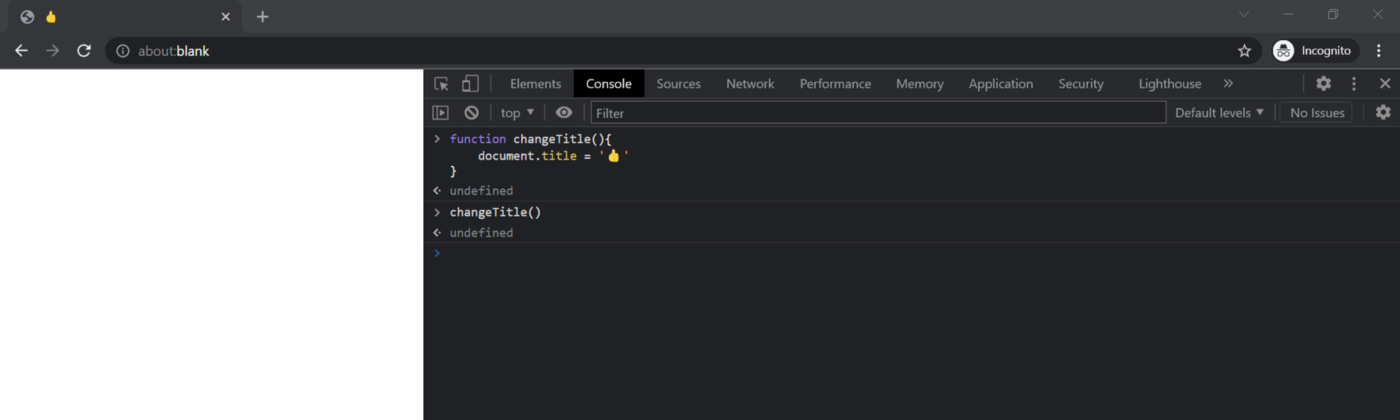


Figure 2 — Experimenting with Chrome’s V8 Engines REPL in Chrome’s Developer Tools. Here, I access the DOM to change the documents title.

Despite JavaScript’s widespread acceptance as the programming language of the web, it may come as a shock that:

**JavaScript is not asynchronous!**

What? You are probably a little confused now. I too was confused. But don’t fear. In [part 2](https://medium.com/@byteslovesbits/asynchronous-javascript-part-2-f300a31c24c2) of this article, I will explain how a *synchronous*, *blocking*, *single-threaded* language like JavaScript can magically become asynchronous.

*Spoiler — JavaScript becomes asynchronous through web APIs and something that is known as the event loop.*

Diagram

Description automatically generated

Figure 3 — the JavaScript Event Loop

If you like what you read, then please share and sub…