



Event loop in JavaScript

Rise of **Asynchronous** behaviour



PRESENTED BY

Vinit Shahdeo

Software Engineer @ Postman

April 19, 2020

`npm run start`



```
// About me
```

```
import { Speaker } from 'HackOn';
```

```
let user = {  
  Name: 'Vinit Shahdeo',  
  Company: 'Postman',  
  Profile: 'Software Engineer', // Ex VITian  
  Twitter: '@Vinit_Shahdeo'  
},  
title: 'Event loop in JavaScript and rise of Asynchronous behaviour';  
_.assign(new Speaker(), user, new Event(title));
```

Js JavaScript

The language which feeds me... 🤪



Web



Server

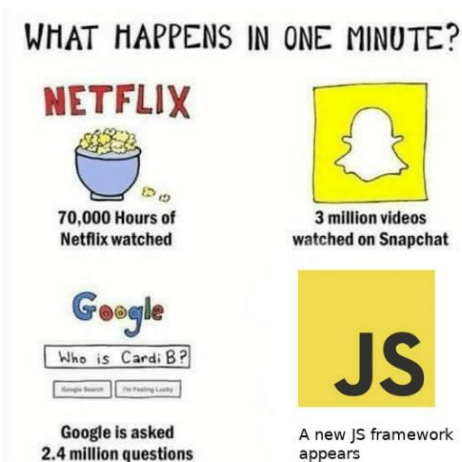


Mobile



Desktop

JavaScript is at the top in terms of
Active Repositories
and total pull requests in **GitHub**.



Looking at you Javascript

JavaScript is everywhere!

NETFLIX





JS is so popular!

Let's dive deep in?

JavaScript is

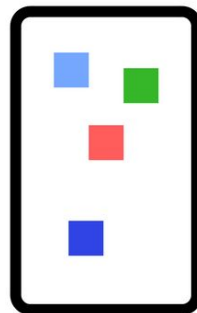
- High-level
- Dynamic
- Loosely typed
- Prototype-based
- **Single threaded**
- Interpreted programming language

JavaScript Engine

- An Interpreter which executes JavaScript code.
- It can be implemented as a standard interpreter, or **just-in-time** compiler that compiles JavaScript to bytecode in some form.



Memory Heap



Call Stack



JavaScript Engine



- **V8** is open source, developed by Google, written in C++
- Rhino — managed by the Mozilla Foundation, open source, developed entirely in Java
- Chakra (JScript9) — Internet Explorer
- Chakra (JavaScript) — Microsoft Edge
- JavaScriptCore — open source, marketed as Nitro and developed by Apple for Safari
- And many more

JavaScript is single threaded

- Single threaded simply means it has single call stack.
- **Call Stack** is where your stack frames are as your code executes
- The call stack is a **LIFO** queue (Last In, First Out).

Let's try to understand what the heck is call stack?

CallStack

watchMyYouTubeLive()

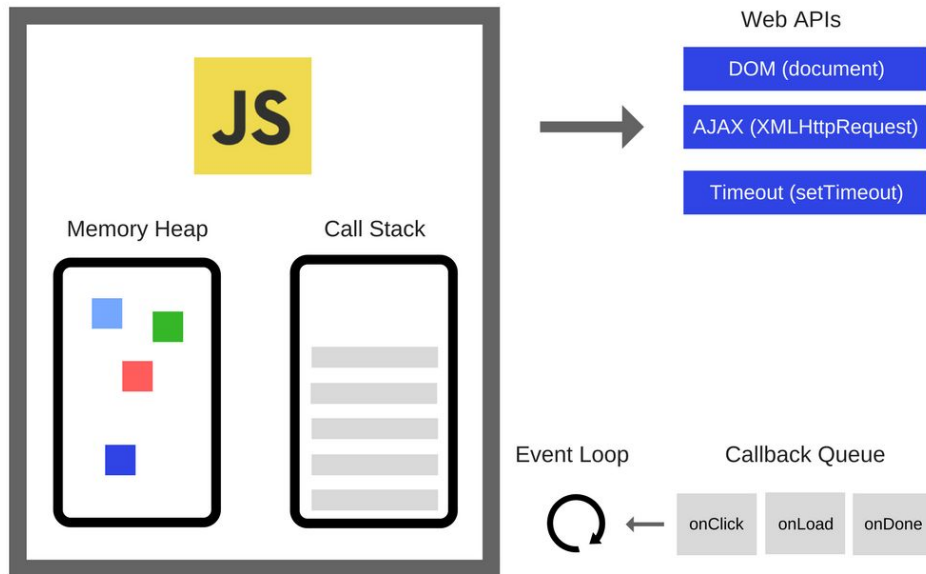
stayHome()

goCorona()

main()

JavaScript Runtime

(Basic Architecture)



- **Memory heap:** This is where the memory allocation happens. Objects are allocated in a heap.
- **Call Stack:** This represents the single thread provided for JavaScript code execution.
- **Web APIs:** Web APIs which are provided by browsers, like the DOM, AJAX, setTimeout and much more. These are not part of JavaScript language but they superpowers the asynchronous behaviour

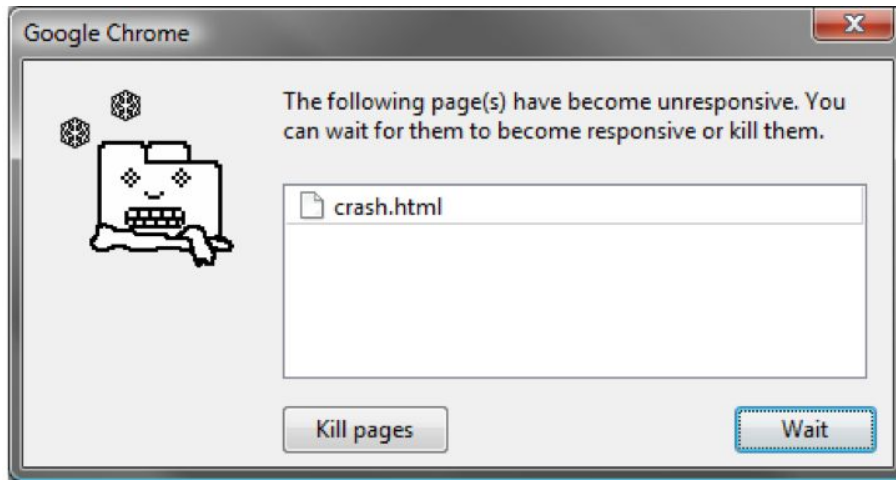
Web APIs

setTimeout()

setInterval()

Let's play with these..

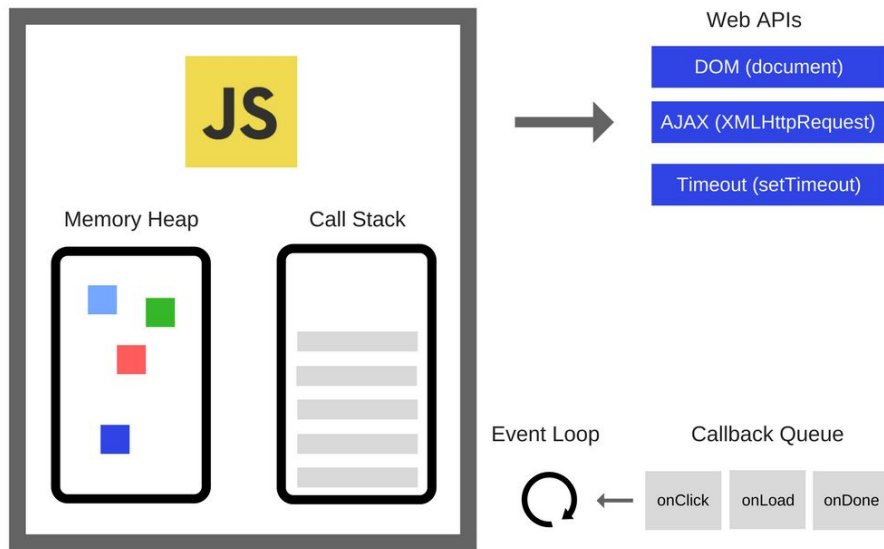
Blocking code



UI is unresponsive,
what's the solution?

Here comes,
the **asynchronous** callbacks

Architecture



Now,
what's **event loop**
and
callback queue?



```
// thank you :)
```

```
const THANK_YOU_MSG = `Thank you so much for being here!  
    you were an amazing audience`;
```

```
let sayThanks = (viewer) => { return THANK_YOU_MSG };
```

```
_.forEach(allViewers, function (participant) {  
    _.times(Number.MAX_SAFE_INTEGER, sayThanks(viewer));  
});
```



Vinit Shahdeo

vinitshahdeo



in relationship with console.log()

Adores JavaScript ツ › Works **@postmanlabs** › Project Admin
@GirlScriptSummerOfCode › Former President **@vinnovateit** ›
Former VP(Tech) **@CodeChefVIT** › VITian ☺

👤 Software Engineer **@postmanlabs**

📍 Bangalore, India

🔗 vinitshahdeo.github.io

👤 355 followers • 32 following

You can find the slides on my
GitHub (**@vinitshahdeo**).

**Feel free to discuss your doubts
on **Issue #1****



```
// Let's get connected
```

```
let socialMediaHandles = {  
  facebook: 'vinit.shahdeo',  
  twitter: 'Vinit_Shahdeo',  
  github: 'vinitshahdeo', // my home page; addicted to green dots  
  linkedin: 'vinitshahdeo'  
},  
contactDetails = {  
  email: 'vinitshahdeo@gmail.com',  
  social: socialMediaHandles  
};  
  
module.exports = contactDetails;
```

Need more help?



@Vinit_Shahdeo

Thank You!

Stay Home, Stay Safe!