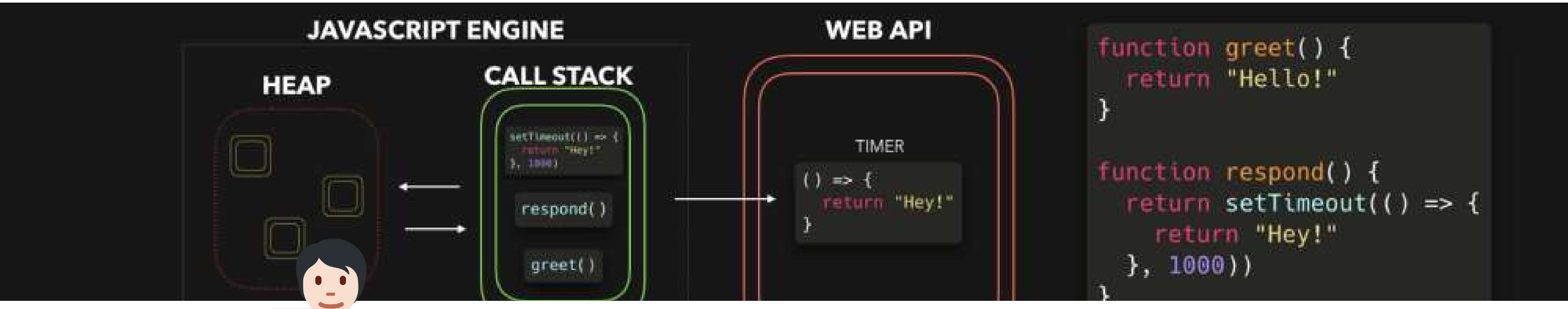


```
function greet() {
  return "Hello!"
}

function respond() {
  return setTimeout(() => {
    return "Hey!"
  }, 1000))
}
```

Call Stack Understanding

Hello everyone👋🏻
Let’s get started with `JS Nights with Tanay` 😎
(Press **Ctrl+Shift+L** on Windows to view this document in Dark Mode😎)

Table of Contents:

📌 INTRODUCTION

Example used by Tanay to demonstrate the difference between thread blocking and Non Thread Blocking : LINK

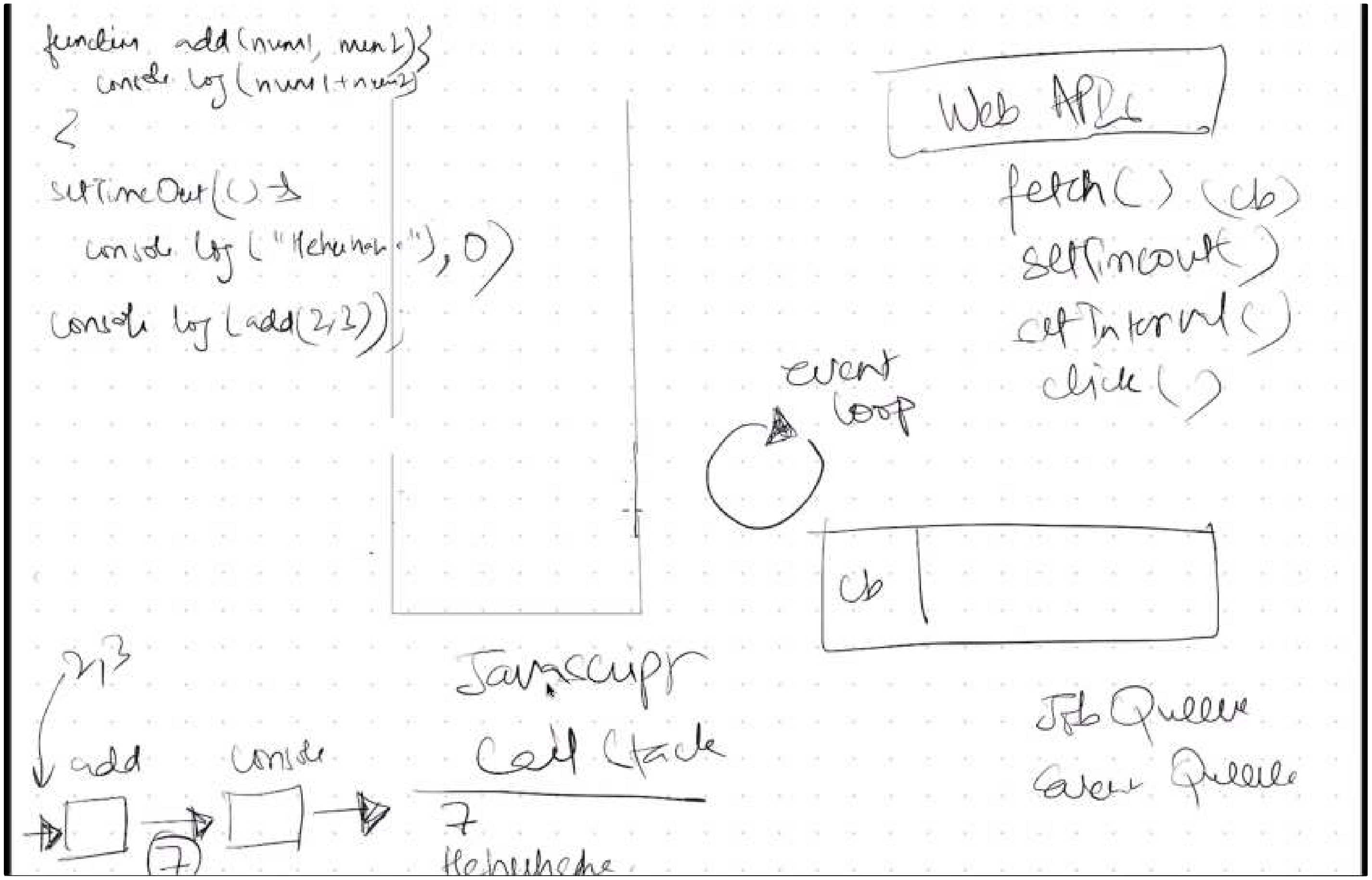
- Browser has a main thread. If you block it , it will become unresponsive to user interactions (such as giving input or responding to click) that in turn will set up a bad UX. So in order to avoid this we can leverage the tools and browser to provide better experiences on the browser while using the apps without any hurdles.
- In Level 0 we came across the use of Fetch while building the translator application. Javascript is a single threaded language that has a lot of good friends making it useful to execute several wonders.

  Blog Topic: How is Asynchronous different from multithreading?

📌 Why do you need ASYNC?

- Suppose you are an user opening Gmail and send a mail to someone and if the browser tells that you cannot do anything until the email gets successfully sent. Imagine how problematic can that become so for that purpose callbacks are used. Javascript's ability of delegation can be very similar to Alfred doing Mr Wayne's tasks and delegating it to Mr Fox, Rachel in order to make sure Mr Wayne gets what he wants.
- Callback is do something when something is done.

  STRUCTURE OF A JS RUNTIME WITHIN BROWSER



- Why call stack needs to be empty: because Javascript is single threaded and can only execute one task at a time thus stack needs to be empty before doing next task. Example : 2 people talking and if third person starts talking in between it will get messed up so it is better to finish the tasks within the call stack and once it gets empty take reference from it.

- Synchronous web api passes through the call stack and Asynchronous ones will go to the web apis section.

- What if there is a heavy task instead of console.log in add function ?
  Then the call stack will remain blocked and that’s why it is advised to shift the intensive operations to web workers

Link for JS Visualiser: https://www.jsv9000.app/
Blog Topics: Web Workers, Performance Optimizations on Web Dev