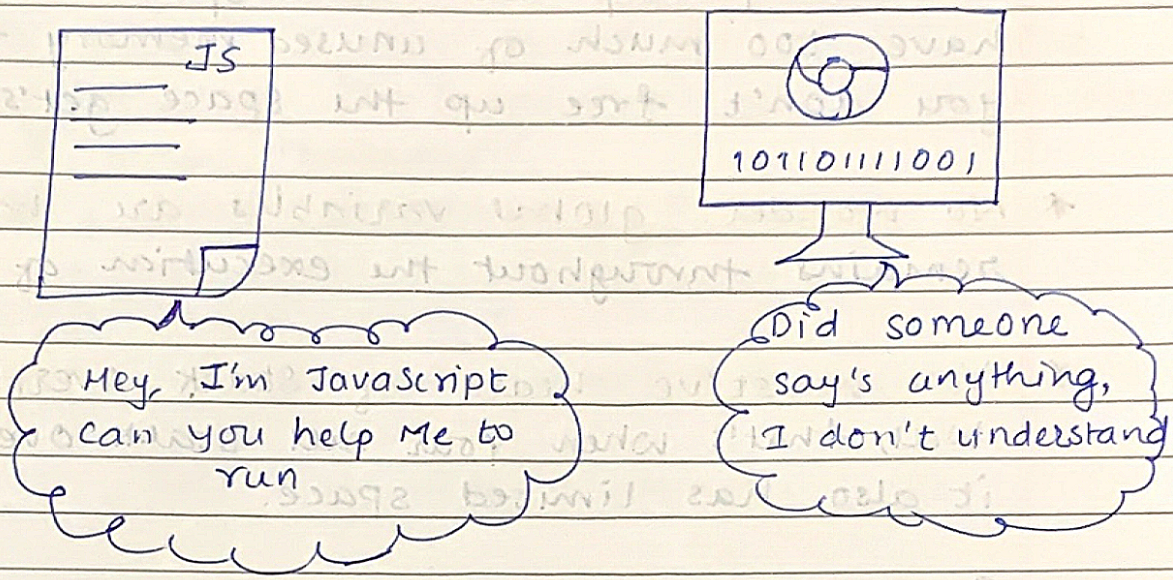
* JavaScript

* JavaScript Engine:



→ Okay... So the browser doesn't understand JavaScript.

→ What it understands is bit's (1's & 0's)

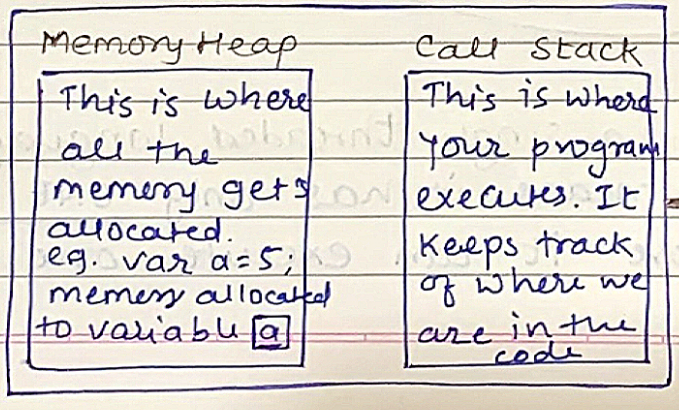
- Then, Who can help us here?

Yes!!! The JavaScript Engine.

There are a lot of JavaScript engines out there written by really smart people.

For e.g.: V8 engine is written in C++ (yes they're programmed too and can be in a different Language).

Okay, So what's inside this JavaScript Engine?



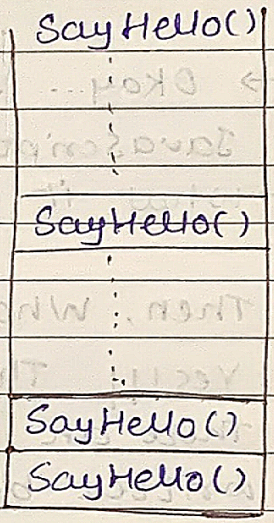
* So ever heard of a memory leak?
A memory heap has limited space. When you have too much of unused memory that you don't free up the space get's filled.

* No wonder, global variables are bad. (They remains throughout the execution of the code).

* You must've heard of stack overflow?
Well, that's when your call stack overflows as it also has limited space.

for e.g.

```
function SayHello ()
{
    console.log ('Rahul Aher');
    SayHello();
}
```



Well, that went into an infinite recursion and we have stack overflow

* Javascript is a single threaded language?
Well, that means it has only ONE CALL STACK and therefore it can execute
time.

Page No.
Date

- Okay.... But why singer threaded?

→ It is quite easy and no complications.

- Okay... Wait! I have heard of asynchronous programming, if Javascript can do that, how is it singer threaded?

let's take an example!

```
setTimeOut ( () => {
        console.log ("setTimeOut is async");
}, 2000);
```

↳ Wait for 2 seconds

setTimeOut is given to us by web API (if gives us various API's) It's technically not a part of Javascript.

* Guess the output.

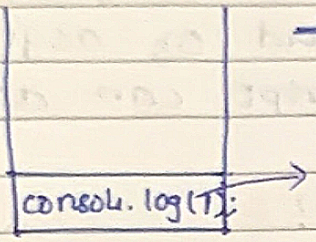
setTimeOut()

```
console.log ('1');
setTimeOut ( () => {
    console.log ('2');
}, 2000);
console.log ('3');
```

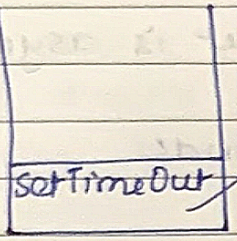
Output :- 1
3
2

Since setTimeOut wait's for 2 seconds it's printed in the end

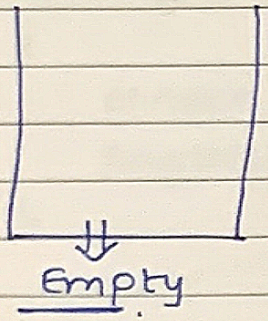
* Behind The scenes:

① 

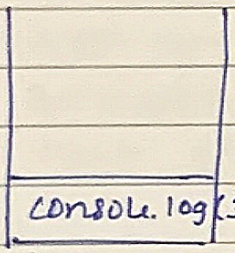
→ Call stack gets executed and get's popped so stack is now empty and out put is 1.

② 

OH! This is given by the web API let me send it to web API.

③ 

Web API
I have setTimeOut with me and I should execute it after 2 sec.

④ 

→ output will print 3

So far :- 1
3

Web API is still waiting

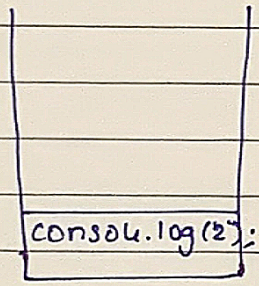
⑤ After 2 seconds are over
Web API ← on It's console.lo[illegible]ld be executed. This is basic[illegible]

that is executed after 2 seconds.
WEB API will send this to callback Queue saying there's a callback ~~que~~ please proceed.

callback 1	callback2	.....

Callback Queue

→ This queue basically keeps track of all callbacks that need to be executed.
Now, there's something called as Event Loop which keep's cheking if stack is empty.
well, now it's empty so Event loop will take a call back from callback queue and put it in the stack



> prints 2
So finally we have
1
3
2

Recap of setTimeOut

① Pushed to stack ⇒ ② Passed to Web API ⇒ ③ Wait's for 2 sec

⑤ Event loop check if stack Empty & pushes to stack. ← ④ pushes callback to call back queue