

How Javascript works

A closer look behind the scene

Front End Gurus session 2

Run time and browser components

- **Javascript Engine**
- Javascript Runtime Environment
- The call Stack
- Concurrency and Event loop

What type of language is
Javascript?

What type of language is Javascript?

Interpreted programming language

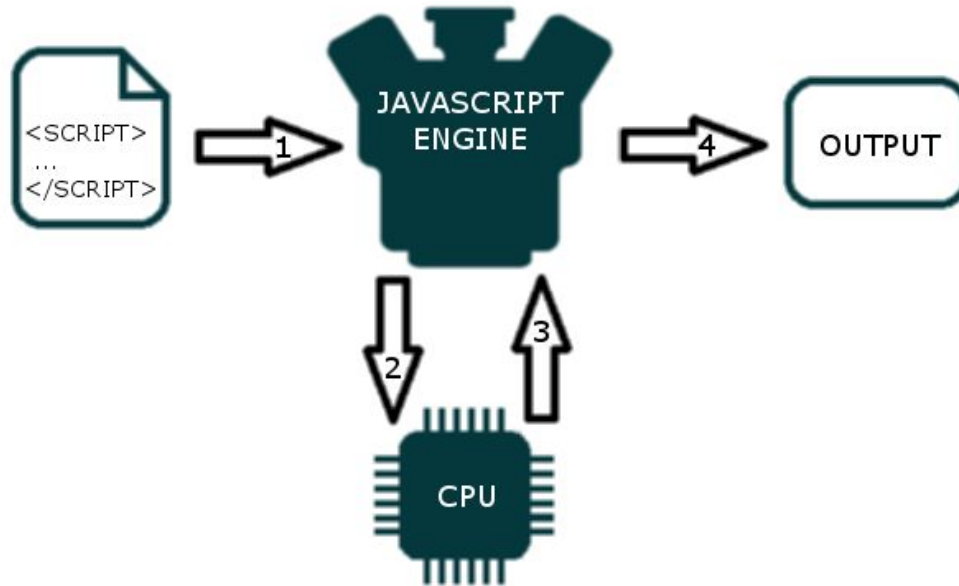
Interpreted programming language

Source code is not compiled into binary code
before execution

BUT

How can a computer understand from a plain text
script?

Interpreted programming language



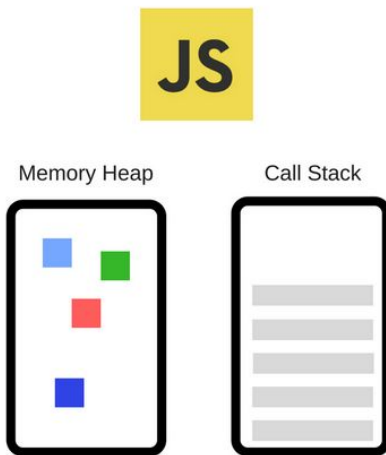
Engine	Brower
V8	Google Chrome and NodeJS
Spider monkey	Mozilla Firefox
JavaScriptCore	Safari
Chakra	Microsoft Edge Browser

Browser engines

Javascript engine

The Engine consists of two main components:

- * Memory Heap — this is where the memory allocation happens
- * Call Stack — this is where your stack frames are as your code executes



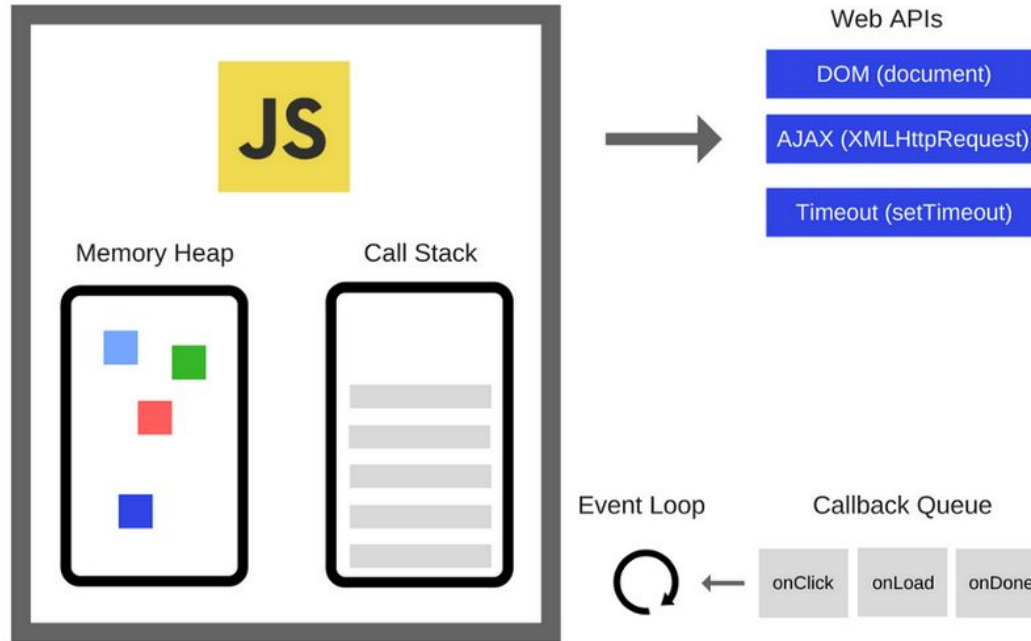
Run time and browser components

- Javascript Engine
- **Javascript Runtime Environment**
- The call Stack
- Concurrency and Event loop

Javascript runtime

JRE is responsible for making JavaScript asynchronous. It is the reason JavaScript is able to add event listeners and make HTTP requests asynchronously.

Javascript runtime



Run time and browser components

- Javascript Engine
- Javascript Runtime Environment
- **The call Stack**
- Concurrency and Event loop

The call stack

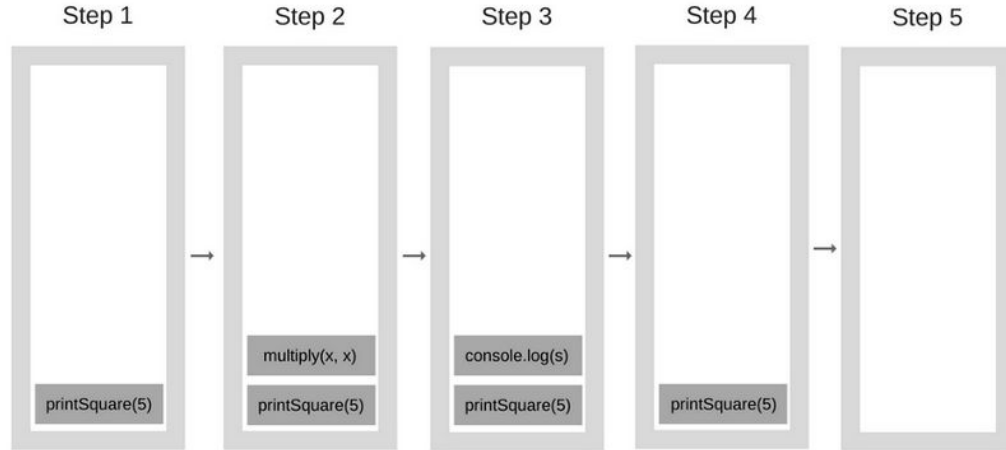
The Call Stack is a data structure which records basically where in the program we are. If we step into a function, we **put it on the top** of the stack. If we return from a function, we **pop off the top of the stack**. That's all the stack can do.

The call stack

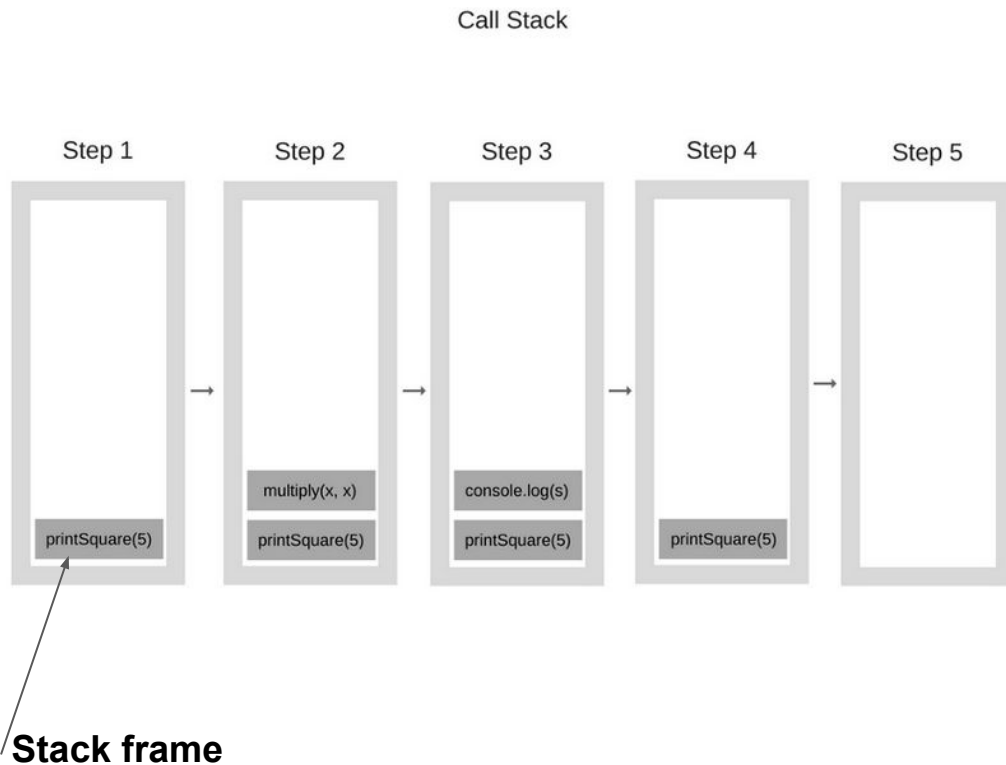
```
function multiply(x, y) {  
    return x * y;  
}  
function printSquare(x) {  
    var s = multiply(x, x);  
    console.log(s);  
}  
printSquare(5);
```

The call stack

Call Stack



The call stack



Run time and browser components

- Javascript Engine
- Javascript Runtime Environment
- The call Stack
- **Concurrency and Event loop**

Concurrency & Event Loop

What happens when you have function calls in the Call Stack that take a **huge amount of time** in order to be processed?

Example: Complex image transformation

Problem?

Concurrency & Event Loop

Browser is blocked & stuck

Stops fluid UI

Browser raise errors: “Do you want to terminate web page?”

BUT

How can Javascript as single threaded language being able to handle concurrency - event loops and multithreading?



Resources:

- <https://blog.devgenius.io/how-javascript-works-behind-the-scenes-88c546173f32>
- <https://www.freecodecamp.org/news/compiled-versus-interpreted-languages/>
- <https://blog.devgenius.io/concurrency-multi-threading-multi-processing-asynchronous-programming-and-event-loop-1b8df9fa6c20>
- <https://thecodest.co/blog/asynchronous-and-single-threaded-javascript-meet-the-event-loop/>