

Js Advanced concepts

Andrei Neagoie

2021-07-25

Contents

1	Js foundation	5
1.1	JavaScript Engine	5
1.1.1	Interpreter vs. compiler	5
1.1.2	Babel and TypeScript	6
1.2	Inline caching	6
1.3	Hidden classes	6
1.4	WebAssembly	6
1.5	Call Stack and Memory Heap	6
1.6	Single-thread	7
1.7	Runtime	8
1.8	Node.js	8
2	Js foundation part 2.	11

Chapter 1

Js foundation

1.1 JavaScript Engine

JavaScript (JS) engine is a translator between *JS language* and *computer native language* - 0,1 [1]. There are tons of JS engines [4]. One of the most famous is google V8 engine. The creator of first JS engine is Brendan Eich cofounder of the Mozilla project, the Mozilla Foundation, and the Mozilla Corporation [3].

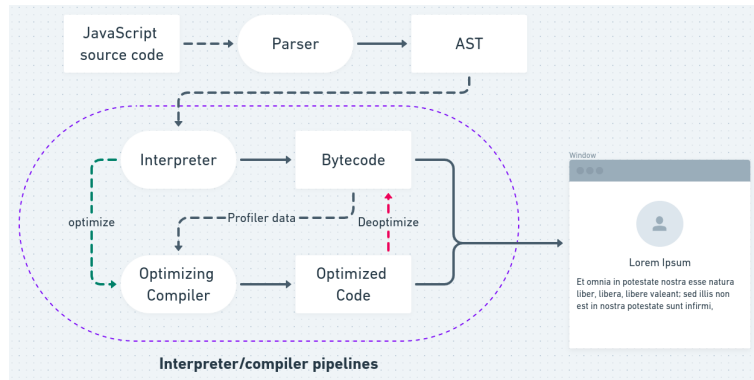


Figure 1.1: JavaScript engine description [1].

1.1.1 Interpreter vs. compiler

Interpreter read and translate code line by line. JS mostly works like interpreted language. In computer science, an interpreter is a translator (computer program) that repeatedly reads instructions (one at a time) and translates them to machine code. It then executes instructions written in a programming or scripting language, without requiring them previously to have been compiled

into a machine language program. An interpreter generally uses one of the following strategies for program execution:

1. Parse the source code and perform its behavior directly;
2. Translate source code into some efficient intermediate representation or object code and immediately execute this;
3. Explicitly execute stored precompiled code made by a compiler which is part of the interpreter system.

Compiler is a computer program that translates computer code written in one programming language (the source language) into another language (the target language). The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language, object code, or machine code) to create an executable program. There are many types of compilers which produce output in different useful forms. A compiler that can run on a computer whose CPU or operating system is different from the one on which the code it produces will run is called a cross-compiler.

1.1.2 Babel and TypeScript

Babel is a Javascript compiler that takes your modern JS code and returns browser compatible JS (older JS code).

Typescript is a superset of Javascript that compiles down to Javascript.

Both of these do exactly what compilers do: Take one language and convert into a different one!

1.2 Inline caching

1.3 Hidden classes

1.4 WebAssembly

1.5 Call Stack and Memory Heap

The (Call stack) shows order of executed commands in JS and primitive variables. In a (stack) data structure that stores information about the active subroutines of a computer program. This kind of stack is also known as an execution stack, program stack, control stack, run-time stack, or machine stack, and is often shortened to just "the stack". Although maintenance of the call stack is important for the proper functioning of most software, the details are normally hidden and automatic in high-level programming languages. Many computer instruction sets provide special instructions for manipulating stacks.

Stack overflow JS is garbage collected language \Rightarrow automatically clean Call stack. If the program reaches the capacity of the Call Stack (stack data structure), it stops work immediately.

```
1 // recursive function
2 const inception = () => {
3     inception();
4 };
5
6 // calling recursive function with stack overflow
7 // RangeError: Maximum call stack size exceeded
8 inception();
9
```

Listing 1.1: recursive function with exceed Call stack

(Memory leak) is the unclean (with garbage collector) memory.

```
1 let array = []
2 for (let i = 5; i>1; i++){
3     array.push(i-1);
4 }
5 // Global variables
6 var a = 1;
7 var b = 1;
8 var c = 1;
9
10 // Event listeners
11 var element = document.getElementById('button');
12 element.addEventListener('click', onClick);
13
14 // setInterval
15 setInterval(() => {})
16
17
```

Listing 1.2: Memory leak

The (Memory Heap) is used for storing dynamic data (objects). The memory heap The (heap) is a different space for storing data where JavaScript stores objects and functions. Unlike the stack, the engine doesn't allocate a fixed amount of memory for these objects. Instead, more space will be allocated as needed. Allocating memory this way is also called dynamic memory allocation.

1.6 Single-thread

JS is a single-threaded programming language. It means only one set of instructions can be executed at time (one Call Stack). Because of this JS is



Figure 1.2: All the values get stored in the stack since they all contain primitive values [2].

synchronous.

1.7 Runtime

(webAPI) -> provided by web browser -> (window) -> fetch() -> make http calls

1.8 Node.js

is JS **runtime** (c++ program). LIUBUV = Web api for node.js and *global* is more or less same as *window*.

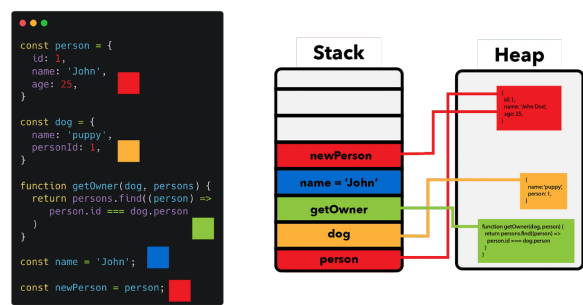


Figure 1.3: In this picture, we can observe how different values are stored. Note how person and newPerson both point to the same object [2].

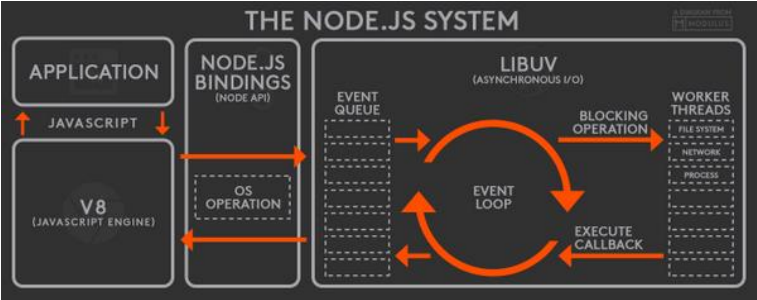


Figure 1.4: Single thread architecture

Chapter 2

Js foundation part 2.

Acronyms

JS JavaScript. 5, 7, 8

List of Figures

1.1	JavaScript engine	5
1.2	Call stack	8
1.3	Memory heap vs Call stack	9
1.4	Single thread	9

Resorces

- [1] Leonardo Freitas. *Inside the JavaScript Engine. A quick introduction of how it works and some tips*. May 2020. eprint: <https://levelup.gitconnected.com/>. URL: <https://levelup.gitconnected.com/inside-the-javascript-engine-896c64cb7623>.
- [2] Felix Gerschau. *JavaScript's Memory Management Explained*. URL: <https://felixgerschau.com/javascript-memory-management/>.
- [3] wikipedia.org. *Brendan Eich*. URL: https://en.wikipedia.org/wiki/Brendan_Eich.
- [4] wikipedia.org. *List of ECMAScript engines*. URL: https://en.wikipedia.org/wiki/List_of_ECMAScript_engines.