

How JavaScript Works Behind the Scenes

Amr A Khllaf

January 27, 2025

Contents

1	An High-Level “Overview” of JavaScript	1
1.1	Basic Definition	1
1.1.1	javascript is a high level, object-oriented programming , multi-paradigm language	1
2	JavaScript: is a high-level, prototype-based object-oriented, multi-paradigm, dynamic language, interpreted or “just-in-time” compiled language with first-class functions, single-threaded, non-blocking event loop, and a garbage collector for automatic memory management.	2
2.1	High-level:	2
2.2	Garbage collector:	2
2.3	Interpreted or “just-in-time” compiled language with first-class functions:	2
2.4	Multi-paradigm:	2
2.4.1	Paradigm	2
2.5	Prototype-based object-oriented:	2
2.6	Dynamic language:	2
2.7	First-class functions:	3
2.8	Single-threaded:	3
2.8.1	Concurrency Model :	3
2.9	Non-blocking event loop:	3

1 An High-Level “Overview” of JavaScript

1.1 Basic Definition

- 1.1.1 javascript is a high level, object-oriented programming , multi-paradigm language
-

2 JavaScript: is a high-level, prototype-based object-oriented, multi-paradigm, dynamic language, interpreted or “just-in-time” compiled language with first-class functions, single-threaded, non-blocking event loop, and a garbage collector for automatic memory management.

2.1 High-level:

It is a high-level language, which means it is closer to human language than machine language. It is easy to read and write. It is not necessary to remember the system architecture or manage memory.

2.2 Garbage collector:

It is a garbage-collected language, which means it has a garbage collector that automatically allocates and deallocates memory. This means that you don't have to worry about memory management.

2.3 Interpreted or “just-in-time” compiled language with first-class functions:

It is an interpreted language, which means that it is executed line by line. However, it is also a “just-in-time” compiled language with first-class functions, which means that it can be compiled into machine code at runtime. This makes it faster than other interpreted languages.

2.4 Multi-paradigm:

It is a multi-paradigm language, which means that it supports different programming paradigms, such as object-oriented, functional, and imperative programming.

2.4.1 Paradigm

An approach and mindset of structuring code, which will direct your coding style and technique.

1. Object-oriented programming (OOP)
2. Functional programming (FP)
3. Procedural programming (PP)

2.5 Prototype-based object-oriented:

It is a prototype-based object-oriented language, which means that it uses prototypes to create objects. This is different from class-based object-oriented languages, such as Java or C++, which use classes to create objects.

2.6 Dynamic language:

It is a dynamic language, which means that it is flexible and can change at runtime. This means that you can add or remove properties from objects, change the type of variables, and even

change the behavior of functions.

2.7 First-class functions:

It is a language with first-class functions, which means that functions are treated as first-class citizens. This means that functions can be assigned to variables, passed as arguments to other functions, and returned from other functions.

2.8 Single-threaded:

It is a single-threaded language, which means that it can only execute one task at a time. This is different from multi-threaded languages, such as Java or C++, which can execute multiple tasks at the same time.

2.8.1 Concurrency Model :

- how the JavaScript engine handles multiple tasks happening at the same time, we need that because JavaScript is single-threaded (This is mean that it can only execute one task at a time).
- So What about the long -running tasks? like fetching data from the server, or reading a file from the disk, or waiting for a user to click on a button, or waiting for a timer to finish.
- This wil be handled by using the **Event Loop** and **Callback Queue**.

2.9 Non-blocking event loop:

It is a non-blocking event loop, which means that it can handle multiple tasks at the same time without blocking the main thread. This is done by using asynchronous functions, such as `setTimeout` or `fetch`, which allow the main thread to continue executing while waiting for the asynchronous function to finish.