

Javascript

JavaScript es un lenguaje:

- De alto nivel
- Interpretado
- Event Driven
- Con un único hilo de ejecución
- Asíncrono, No bloqueante
- Concurrente

Motor V8

Motor de código abierto para Javascript desarrollado por Google

Es un programa escrito en C++, que toma el código en Javascript lo compila y lo ejecuta.

Es usado en el navegador Chrome, navegadores basados en Chromium, usado tambien en entornos de ejecución como NodeJS y Deno.

Otros navegadores tienen su propio motor:

- SpiderMonkey: Firefox
- Javascript Core: Safari
- Chakra: Internet Explorer y primeras versiones de Edge

Chromium:

Navegador web, libre y de código abierto de Google que es tomado como base para desarrollar otros navegadores, como:

- Brave
- Opera
- Vivaldi

Motor V8 Funcionamiento

Se compone de

- Call Stack
- Memory Heap
- Host APIs (Web/Node)
- Event Loop
- Task/Callback Queue

Call Stack

Javascript maneja una sola pila de llamadas, esta es una estructura tipo LIFO (Last-In, First-Out)

Las funciones son procesadas en el orden en que son llamadas.

Cada que se llama a una función, esta es añadida a la pila de ejecución.

Si dentro de esta función llamamos a otra, esta segunda función se añadirá a la pila, encima de la anterior y así sucesivamente. Según las funciones se vayan ejecutando son sacadas de la pila de ejecución.

Memory Heap

Es la parte de la memoria donde se guardan objetos y funciones

heap quiere decir montículo o montón es decir no hay un orden en como se guardan las cosas.

APIS

APIs escritas en Javascript y que vienen integradas en el navegador/runtime y que nos permiten implementar diversas funcionalidades.

Web : XMLHttpRequest / API Fetch / API de Geolocalización Node: fs / https / path / crypto

Motor V8 Funcionamiento

Se compone de

- Call Stack
- Memory Heap
- Host APIs (Web/Node)
- Event Loop
- Task/Callback Queue

Event Loop

Ciclo infinito que escucha si hay tareas pendientes en el Task/Callback Queue y las pasa a la Call Stack cuando este vacía

Siempre estará buscando tareas en el Task/Callback Queue y revisando si el Call Stack esta vacío

Task Queue/Callback Queue Tareas o funciones pendientes de ejecutar que serán ejecutadas cuando termine o se reciba el resultado de una operación.

Estas tareas se van colocando en esta cola en espera de ser pasadas a la Pila de ejecución, cuando esta este vacía.