**JAVASCRIPT**

**¿Qué es JavaScript?**

Javascript es un lenguaje de scripting multiplataforma y orientado a objetos (Interpretado). Se caracteriza también por ser un lenguaje liviano. Javascript ejecuta el código en manera de un solo subproceso, esto infiere que no realiza acciones simultáneamente si no, solo maneja y corre el código un proceso a la vez.

**¿Cómo funciona JavaScript?**

Javascript Engine es un software que interpreta código JavaScript y que a su vez ejecuta un script acorde a las instrucciones dadas. Todos los navegadores tienen un motor JavaScript.

**Motores en los navegadores:**

**V8 Engine** : creado por Google, es open source y utilizado por el navegador de Google Chrome.

**Chakra Engine :** Creado por Microsft, es utilizado por su nuevo navegador web Internet Explorer 9

**SpiderMonker Engine :**  Creado por la fundación Mozilla, es utilziado por su navegador web Mozilla Firefox.

**Carakan Engine :** Creado por Opera Software, utilizado en su navegador Opera

**Composición motor JavaScript:**

El motor se compone de dos componentes principales:

**Memoria ( Memory Heap )** encargado de la asignación de memoria.

**Pila de Llamadas a Funciones ( Call Stack )** aquí es donde se encuentra las funciones a medida que se ejecuta el código.

**Runtime ( Tiempo de Ejecución )**

setTimeout() son APIs las cuales no son proporcionadas por el motor V8, ¿De donde vienen?

Adicional al motor de JavaScript, tenemos las Web APIs que son provistas por los navegadores web, como DOM, AJAX, setTimeout y mucho más.

Luego también tenemos el Event Loop y cola de funciones ( Queue Callback )

**Pila de Llamadas de Funciones ( Call Stack )**

JavaScript es un lenguaje de programación de un único subproceso, lo que significa que tiene un solo Call Stack, por lo tanto solo puede hacer una cosa a la vez.

**¿Qué es Call Stack?**

Es una estructura de datos que registra básicamente en que parte del programa estamos. Si entramos en una función, la colocamos en la parte superior de la pila. Si regresamos de una función, salimos de la parte superior de la pila. Eso es todo lo que puede hacer.

**Overflowing,** se denomina cuando el navegador obtiene muchas call stack esto significa que esta realizando muchas acciones una y otra vez, el navegador cuenta con un número limite, si el numero de call stack supera este número, el navegador generará un error que normalmente se representa así:

**Uncaugth RangeError: Maxium call stack size exceded**

**¿**Qué sucede si se tiene funciones Call Stack que toman un gran cantidad de tiempo ser procesadas**?**

El problema es que mientras la Call Stack tiene funciones para ejecutar, el navegador no puede hacer otra cosa: **Se bloquea**, puesto que es solo puede realizar un **call stack** a la vez, esto generá problema si se quiere crear buenas UI ( User Interfaces ).

**¿**Cuál es la solución a esto?

Utilizar **asynchronous callbacks**, Llamados asíncronos.

**Además de esto, JavaScript cuenta con Hoisting y Scope, mirar en el archivo acerca de Hoisting y Scope.**

**Es importante estos dos conceptos para entender un poco más la manera de interpretación de código JavaScript por parte del motor.**