



UNIVERSIDAD TECNOLÓGICA DE AGUASCALIENTES

TEMA:

"CallStack."

NOMBRE ESTUDIANTE:

Talamantes Castañeda Ángela María

MATRÍCULA:

191243.

CARRERA/AULA/GRADE/GRUPO:

IDGS ~ Aulas2, 9ºA.

PROFESOR:

Alberto Campos Hernández.

MATERIA:

Desarrollo Móvil Integral.

¿QUÉ ES?

También llamado “Pila de ejecución / Pila de contextos de ejecución”, el cual se usa para mapear, saber en dónde están ubicados y dónde anteriormente estaban.

¿CÓMO FUNCIONA?

Este si necesita añadir una pila, se debe de hacer desde arriba, y si esta se le tiene que quitar alguna, se deben de quitar las que están anterior o si se quiere agregar, esta debe ser desde la última que se agregó. (LIFO: Last In First Out).

Stack Trace: La secuencia de llamadas que se fueron dando durante la ejecución de un programa hasta que sucedió una excepción o un error inesperado (es muy útil cuando buscamos bugs en el código).

Scope: Contexto actual de ejecución, este está determinado por el conjunto de variables y funciones que se pueden acceder de esa función cuando se está ejecutando.

Contexto de ejecución (Scope): EL conjunto de variables que tiene acceso la función, el contexto (this), arguments u objeto global (window/global).

This: Hace referencia al objeto “dueño” de la función y el valor que lo contiene determina lo que se llama el contexto de la función.

Arguments: Objeto similar a los arrays, que tienen todas las funciones en su scope local (=> no están disponibles), este tiene todos los parámetros que recibe la función cuando fue invocada.

Función de la pila de ejecución se guarda:

- Nombre de la función.
- Archivo al que pertenece.
- Número de la próxima línea a ejecutar.
- Contexto de ejecución.

Lo que se apila es un frame/registro que contiene la función:

- Contexto de ejecución.
- Nombre de archivo al que pertenece.
- Número de la próxima línea a ejecutar que se va a ir actualizando a medida que la función se vaya ejecutando, cada que se ejecute una función, esta creará un nuevo registro y lo va a agregar a su pila de ejecución y el motor de JS trabajará con ese registro hasta que esta se termine de ejecutar. Y ahí es dónde saca el registro de la pila y continúa trabajando con el que estaba debajo el cual se puede acceder desde las secciones de CallStack y Scopes (en las herramientas de los navegadores para desarrolladores)