

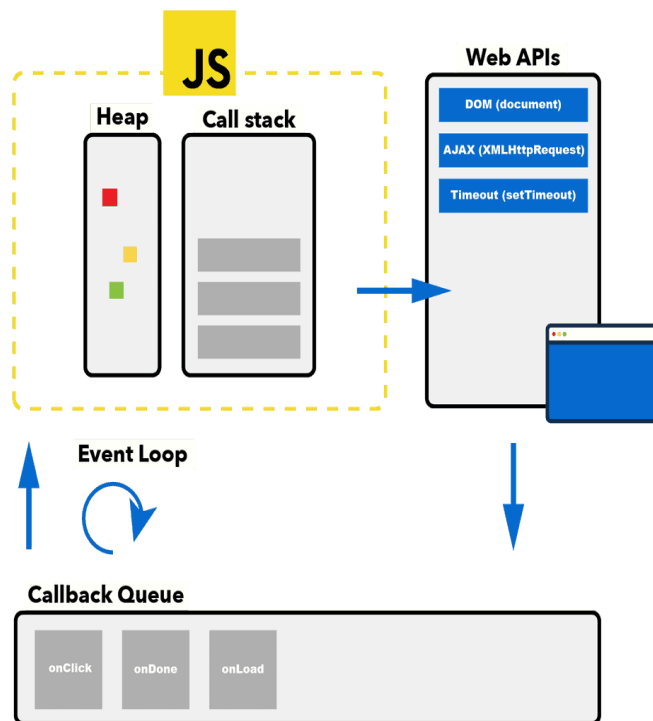
Event Loop en Javascript

Introducción

El Event Loop es un componente clave de la arquitectura de JavaScript que maneja el procesamiento asíncrono de eventos.

El Event Loop es un ciclo continuo que ejecuta una cola de tareas asíncronas en segundo plano y espera a que se completen.

El Event Loop es esencialmente un bucle que se ejecuta continuamente y verifica si hay tareas pendientes que deben procesarse. Si hay tareas pendientes, las procesa una por una y luego regresa al comienzo del bucle para esperar a que lleguen más tareas.



Representación a alto nivel de event loop.

Call stack (pila de llamadas)

Javascript puede ejecutar una tarea a la vez. **La pila de llamadas** es un mecanismo que ayuda a javascript a mantener un registro de las llamadas y su orden.

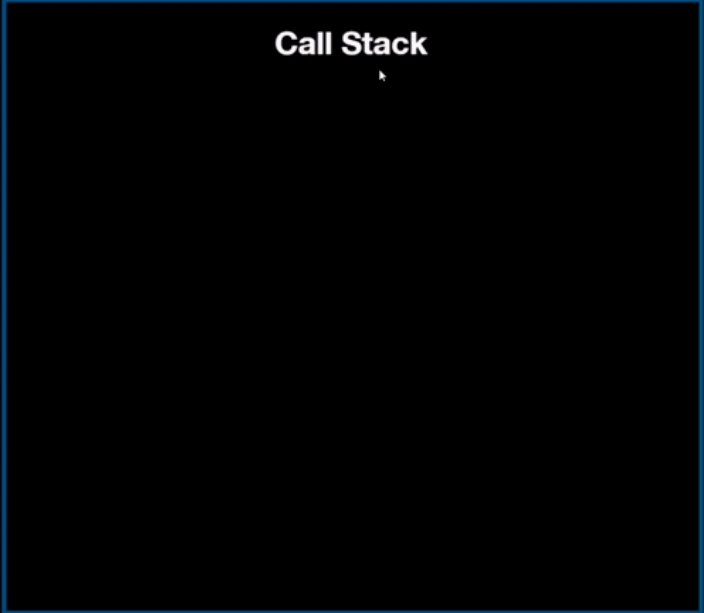
Cada vez que se agrega una llamada a una función, se agrega un ítem a la pila y cada vez que se llama se borra de la pila.

```
function add(a, b) {
  return a + b;
}

function average(a, b) {
  return add(a, b) / 2;
}

var x = average(10, 20);
```

Call Stack



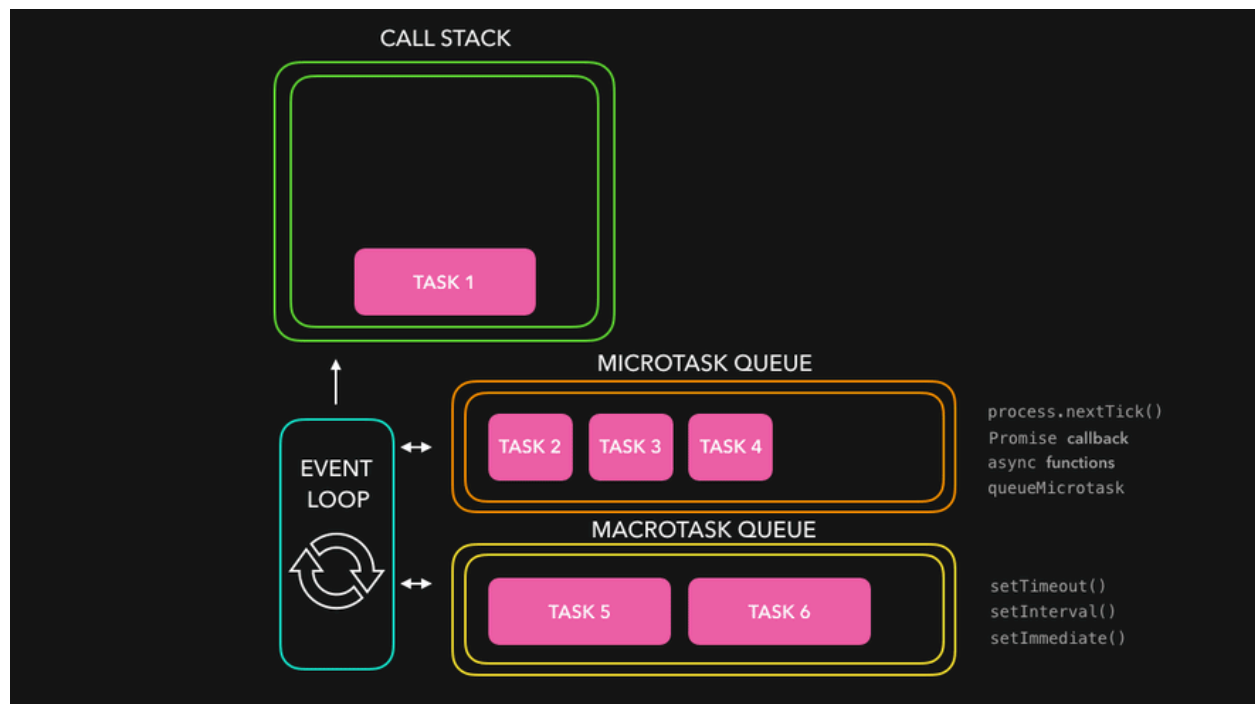
El Event Loop permite que JavaScript sea asíncrono, lo que significa que las tareas no bloquean la ejecución del programa. Cuando se inicia una tarea asíncrona, el control se devuelve inmediatamente a la línea siguiente de código sin esperar a que se complete la tarea. Una vez que la tarea se completa, se agrega a la cola de tareas pendientes y se procesa en la siguiente iteración del Event Loop.

El Event Loop de JavaScript se divide en dos fases: la fase de ejecución y la fase de espera. Durante la fase de ejecución, se procesan las tareas pendientes. Durante la fase de espera, el Event Loop espera a que lleguen más tareas. La fase de espera se divide en dos partes: la cola de microtareas y la cola de macrotareas.

Micro tareas y macro tareas

Las microtareas son tareas de alta prioridad que se procesan inmediatamente después de la fase de ejecución. Las microtareas incluyen promesas, observadores de mutación y eventos de proceso.

Las macro tareas, por otro lado, son tareas de baja prioridad que se procesan después de que se hayan procesado todas las microtareas. Las macro tareas incluyen eventos de temporizador, eventos de red y eventos de entrada.



Conclusión

En resumen, el Event Loop de JavaScript es un mecanismo que permite que el código se ejecute de forma asíncrona en segundo plano, sin bloquear el hilo de ejecución. Las tareas se procesan en una cola, lo que garantiza que se ejecuten en el orden correcto. El Event Loop es esencial para el procesamiento asíncrono en JavaScript y es fundamental para muchas de las características avanzadas del lenguaje.

Lectura recomendada:

<https://developer.mozilla.org/es/docs/Web/JavaScript/EventLoop>

Como funciona event loop (en Inglés) <https://www.youtube.com/watch?v=8aGhZQkoFbQ>