Web API & Event Loop

1. Web API

Definition: Web APIs are provided by the browser (or Node.js in server environments) to allow JavaScript to perform tasks like HTTP requests, timers, DOM manipulation, and event handling.

Examples:

# setTimeout(), setInterval()

# fetch()

# addEventListener()

# DOM manipulation methods like document.querySelector()

2. Event Loop

- Call Stack

# Executes synchronous code in a Last In, First Out (LIFO) order.

# Functions are pushed onto the stack when invoked and popped when completed.

- Web APIs

# Handle asynchronous tasks (e.g., setTimeout, HTTP requests).

# Once completed, they send callbacks to the callback queue.

- Callback Queue

# Stores callbacks from completed asynchronous tasks.

# Callbacks are processed in order and moved to the call stack when it’s empty.

- Event Loop

# Monitors both the call stack and the callback queue.

# When the call stack is empty, it pushes the first callback from the queue onto the stack for execution.

- Microtasks vs. Macrotasks

# Microtasks (e.g., Promise callbacks) are processed before macrotasks.

# Macrotasks (e.g., setTimeout, I/O events) are processed after microtasks in the queue.