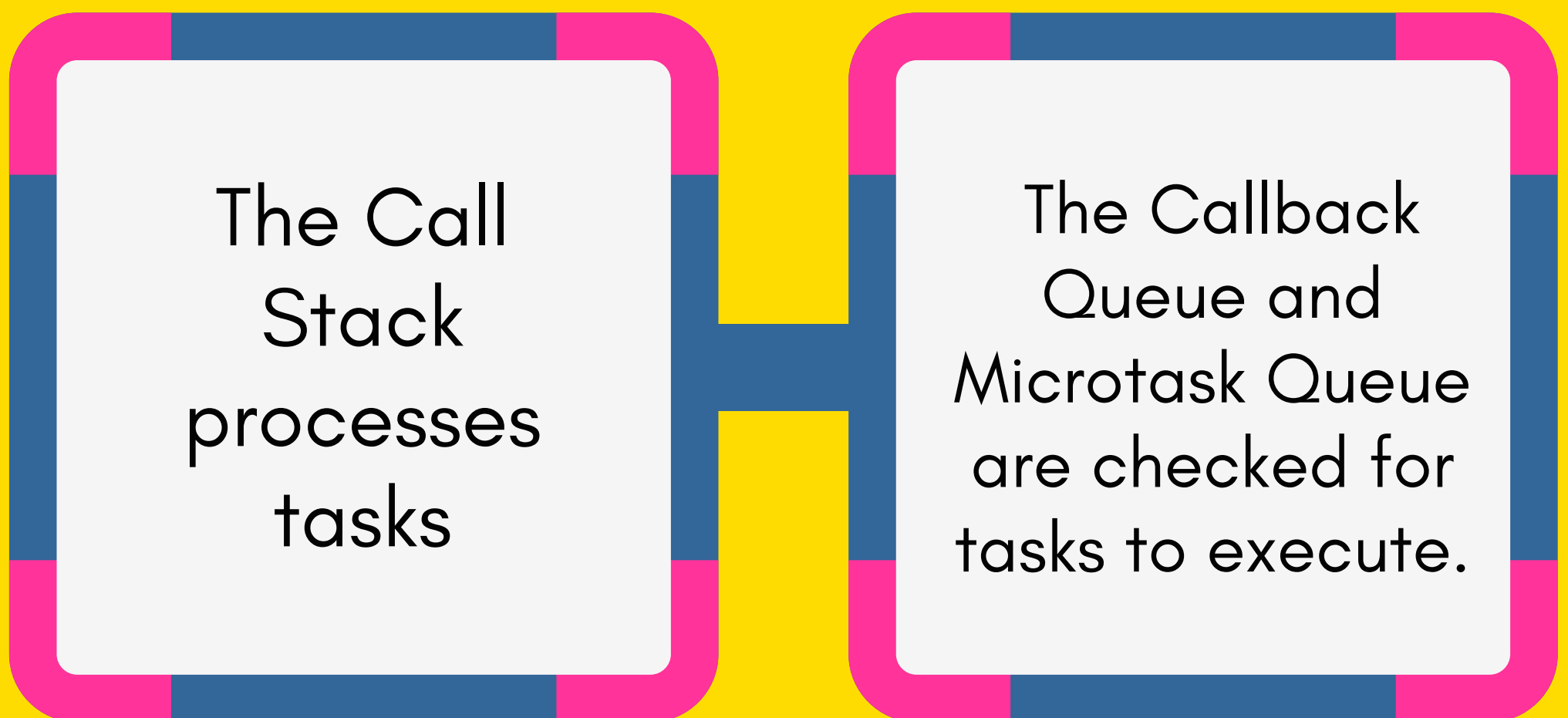


Event Loop in JavaScript



What is the Event Loop?

- The Event Loop is a mechanism in JavaScript that ensures:

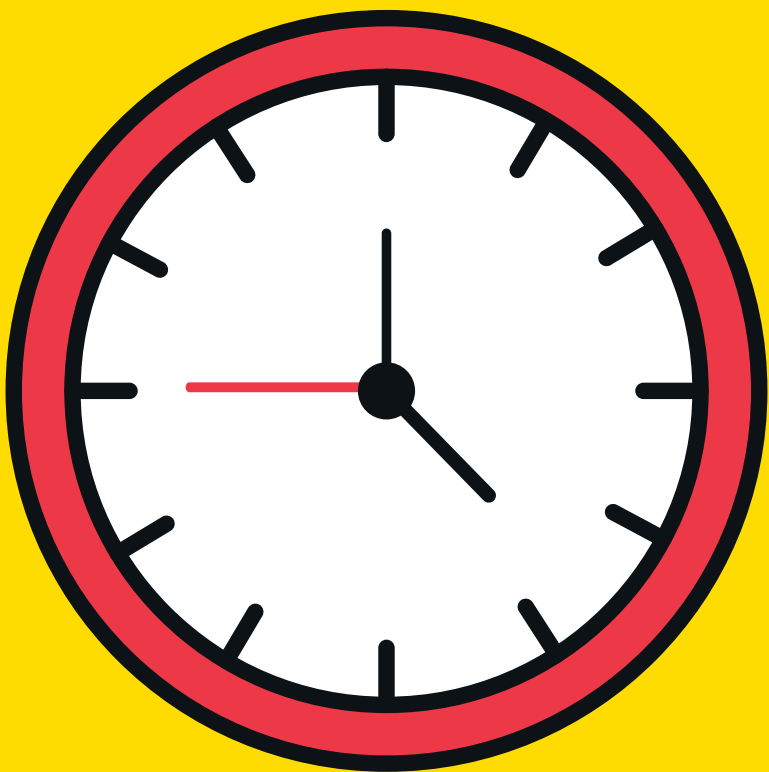


- It keeps JavaScript non-blocking and single-threaded!



How It Works?

- JavaScript executes synchronous code in the Call Stack.



- Asynchronous tasks like `setTimeout` are sent to the Web APIs.



How It Works?

- Once complete, their callbacks go to the Callback Queue (or Microtask Queue for promises).



- The Event Loop checks if the Call Stack is empty and pushes tasks from the queues.



Example

```
JS

console.log("Start");

setTimeout(() => {
  console.log("Timeout");
}, 0);

Promise.resolve().then(() => {
  console.log("Promise");
});

console.log("End");
```



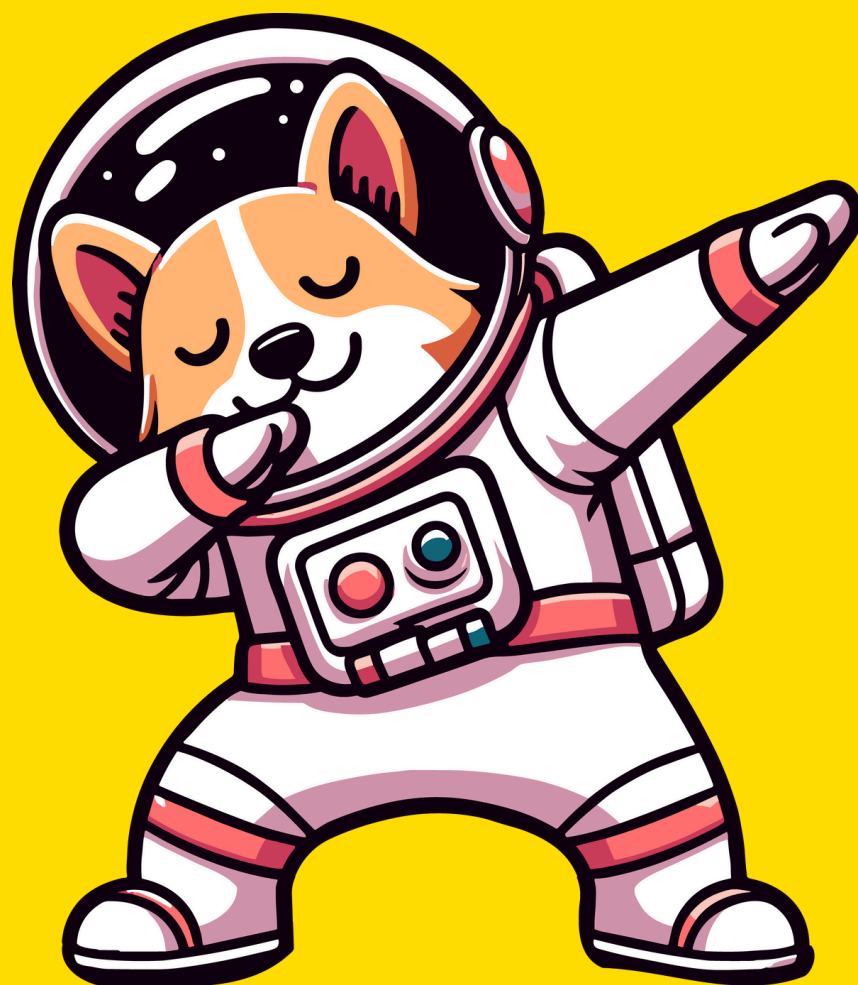
Why This Order?

- **console.log("Start") and console.log("End"):** Synchronous tasks in the Call Stack.
- **Promise callback:** Goes to the Microtask Queue (higher priority).
- **setTimeout callback:** Goes to the Callback Queue (lower priority).



Summary

- **Call Stack:** Executes synchronous code.
- **Web APIs:** Handles asynchronous tasks.
- **Callback Queue:** Processes tasks like `setTimeout`.
- **Microtask Queue:** Processes promises and `MutationObserver`.



Keep Exploring Javascript with us!

Share this with a friend who needs it and
make sure to practice these in scribbler.



Scribbler.live

Free and Open Interface to
experiment JavaScript