How JS handles Async operations ?

→ Javascript is a Single threaded language.

→ Javascript by default only supports Synchronous code execution.

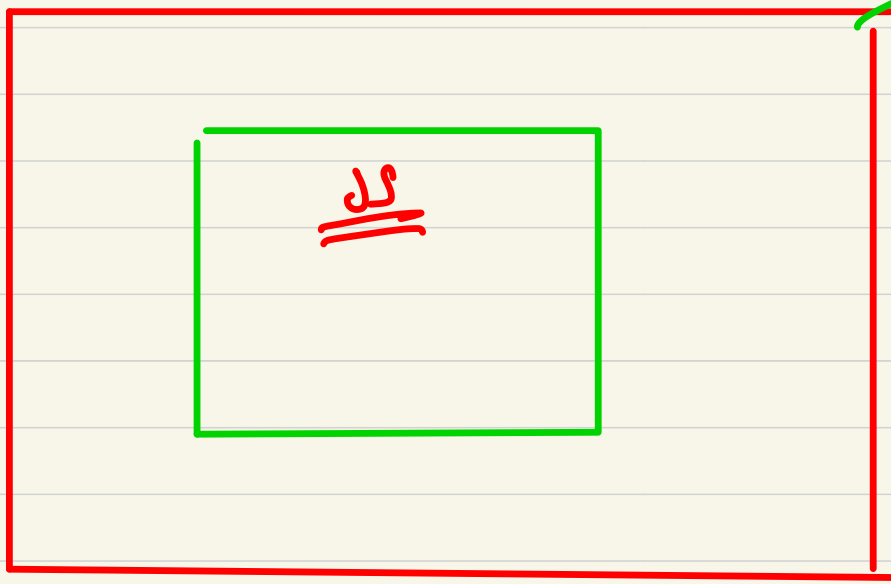
Note → The above property of Sync code execution only works for operations natively known to Javascript.

Node

Ryan Dahl

2009

Js Runtime

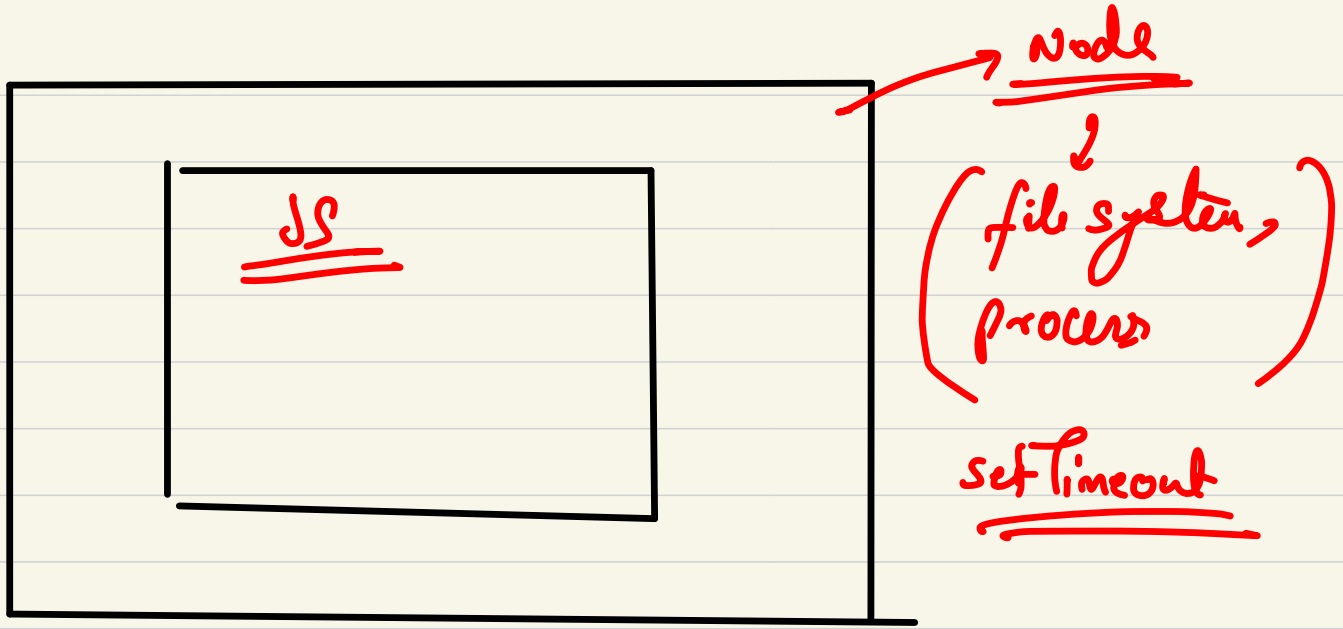


Browser

Dom APIs

setTimeout

setInterval


JS
Node
(file system, process)
setTimeout

Now we know that runtime also provides functionalities that can be leveraged by JS.

But how JS handles them.

⇒ We have a few questions to answer →

Q-1 We can easily do tasks that take a lot of time to complete without blocking the code flow. How??

```
function process() {
    console.log("Start");
    setTimeout(function exec() {
        console.log("Executed some task")
    }, 3000);
    for(let i = 0; i < 10000000000; i++) {
        // some task
    }
    console.log("End");
}

process();
```

Condition → false

timer → 3sec

runtime

3sec → 8sec

Call stack

event loop

event queue

the condition to start executing tasks from event queue is that,

1) the call stack should be empty i.e. no function in the call stack is left to be executed

2) the global code is also done.

```
function process(){
    console.log("start");
    for(var i=1;i<3;i++){
        setTimeout(function exec(){
            console.log("executed after some time");
        },3000);
        console.log("inside for loop");
    }

    for(var j=0;j<1000000;j++){

    }

    console.log("end");
}
process();
```