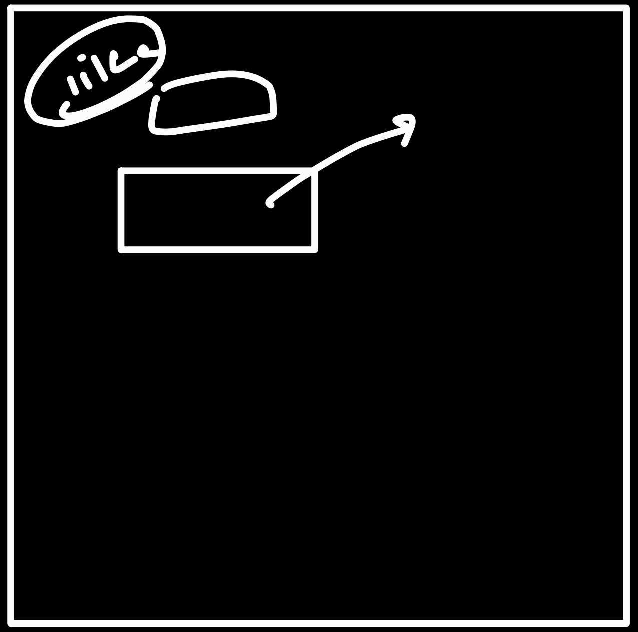
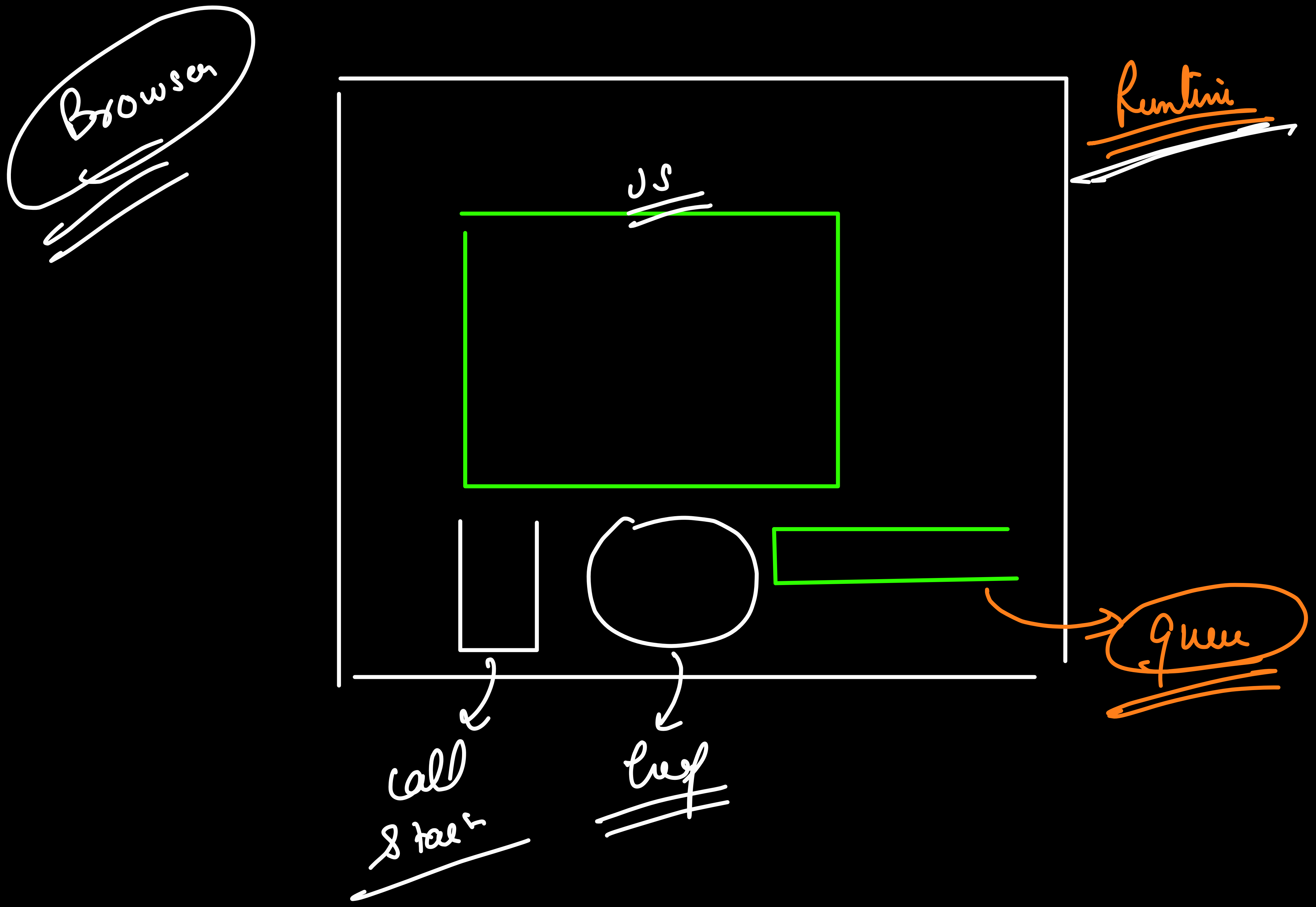
Async Programming With JS

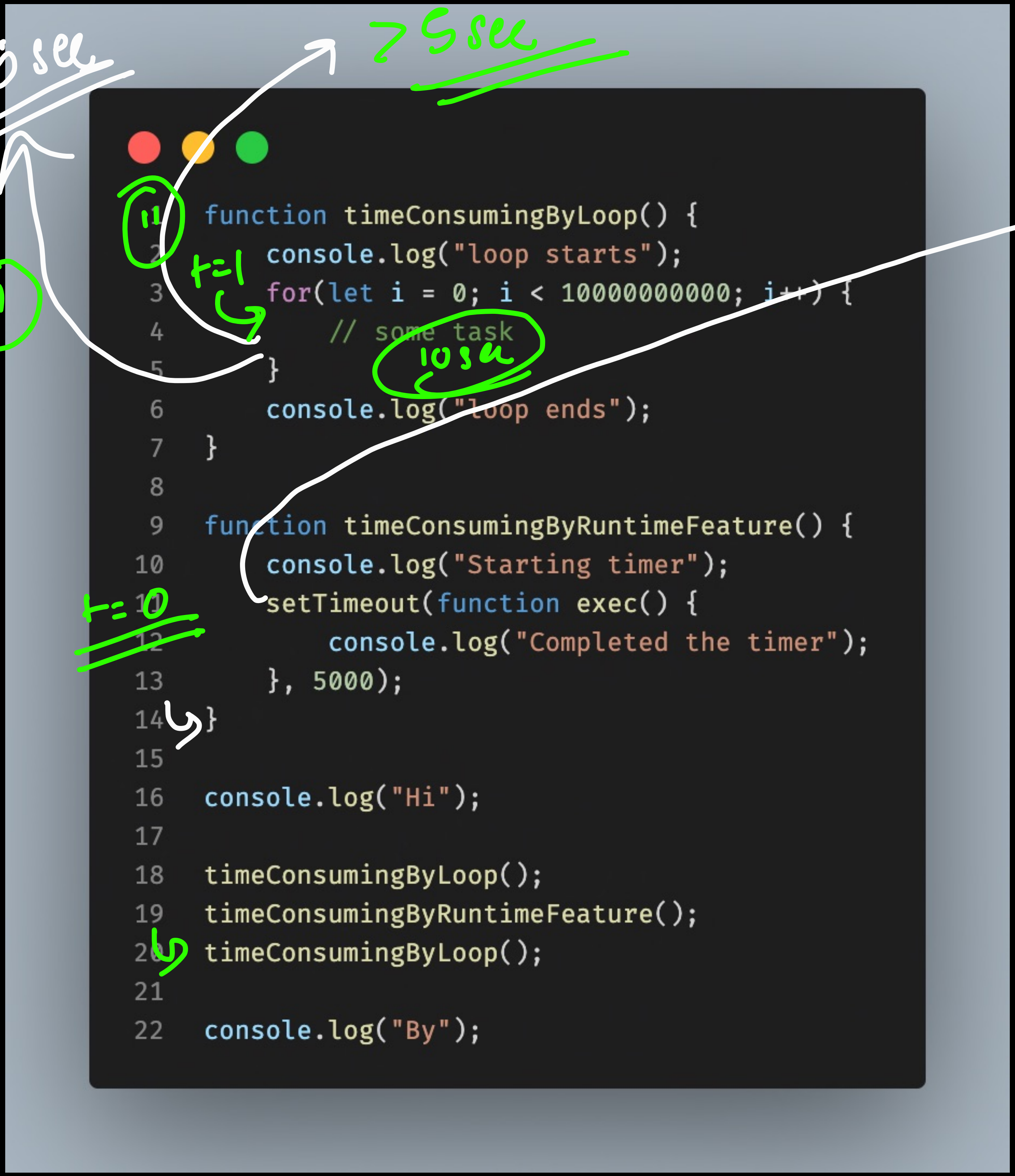
1) JS is sync in nature

2) JS is single threaded.

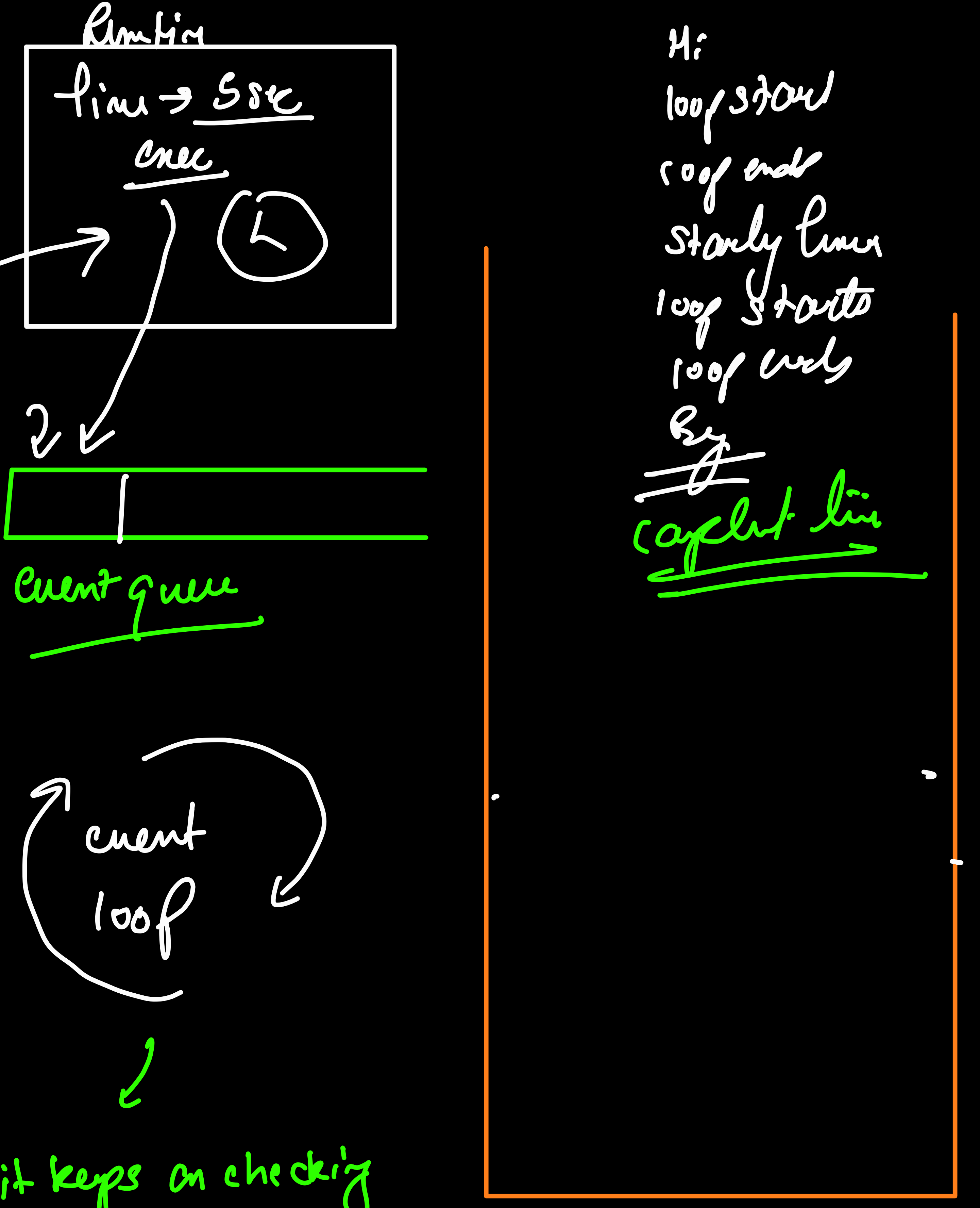
if we execute valid ecmascript code viz given by the standards




Browser
Runtime
JS
call Stack
loop
queue

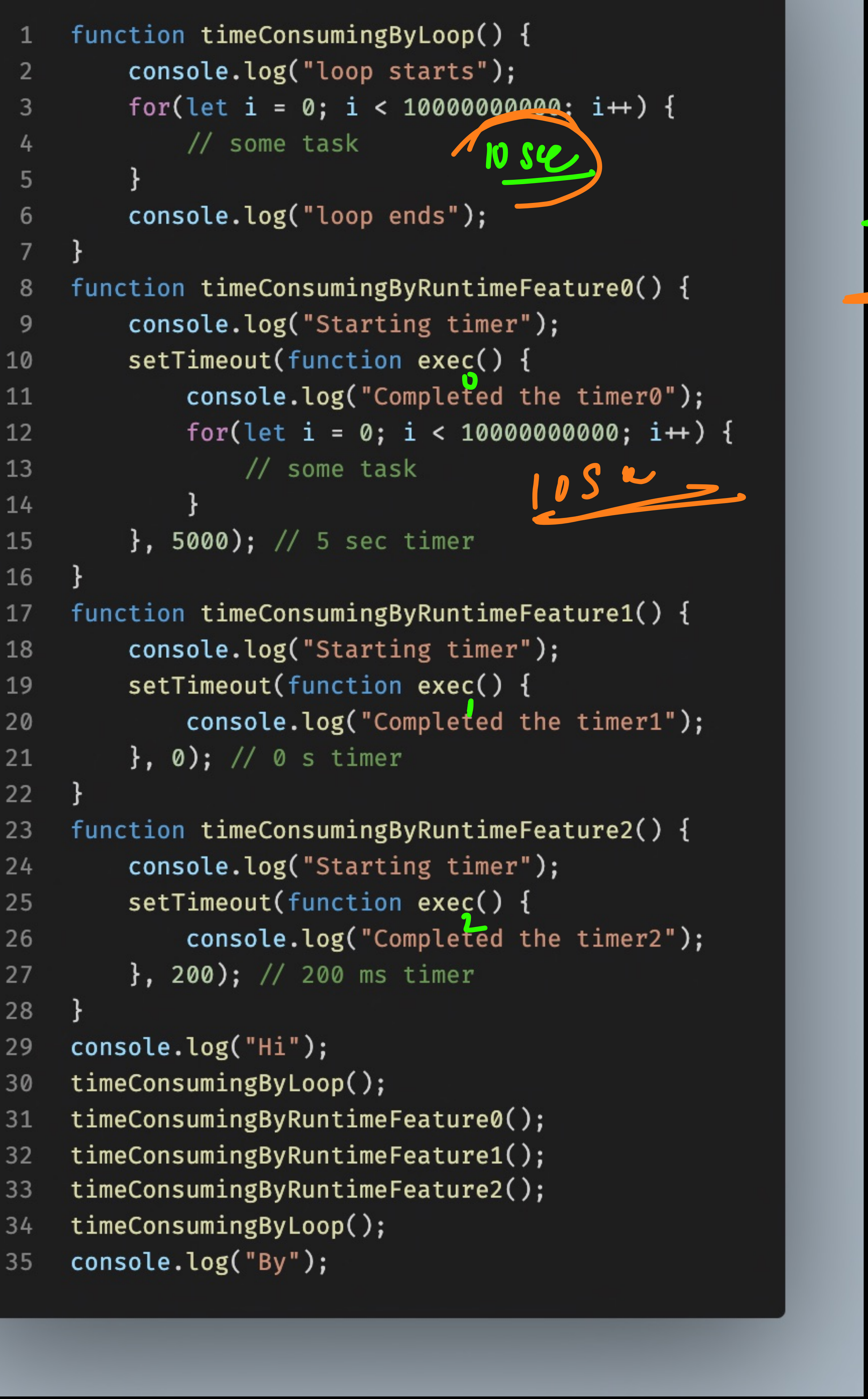


```
function timeConsumingByLoop() {
    console.log("loop starts");
    for(let i = 0; i < 10000000000; i++) {
        // some task
    }
    console.log("loop ends");
}

function timeConsumingByRuntimeFeature() {
    console.log("Starting timer");
    setTimeout(function exec() {
        console.log("Completed the timer");
    }, 5000);
}

console.log("Hi");

timeConsumingByLoop();
timeConsumingByRuntimeFeature();
timeConsumingByLoop();

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

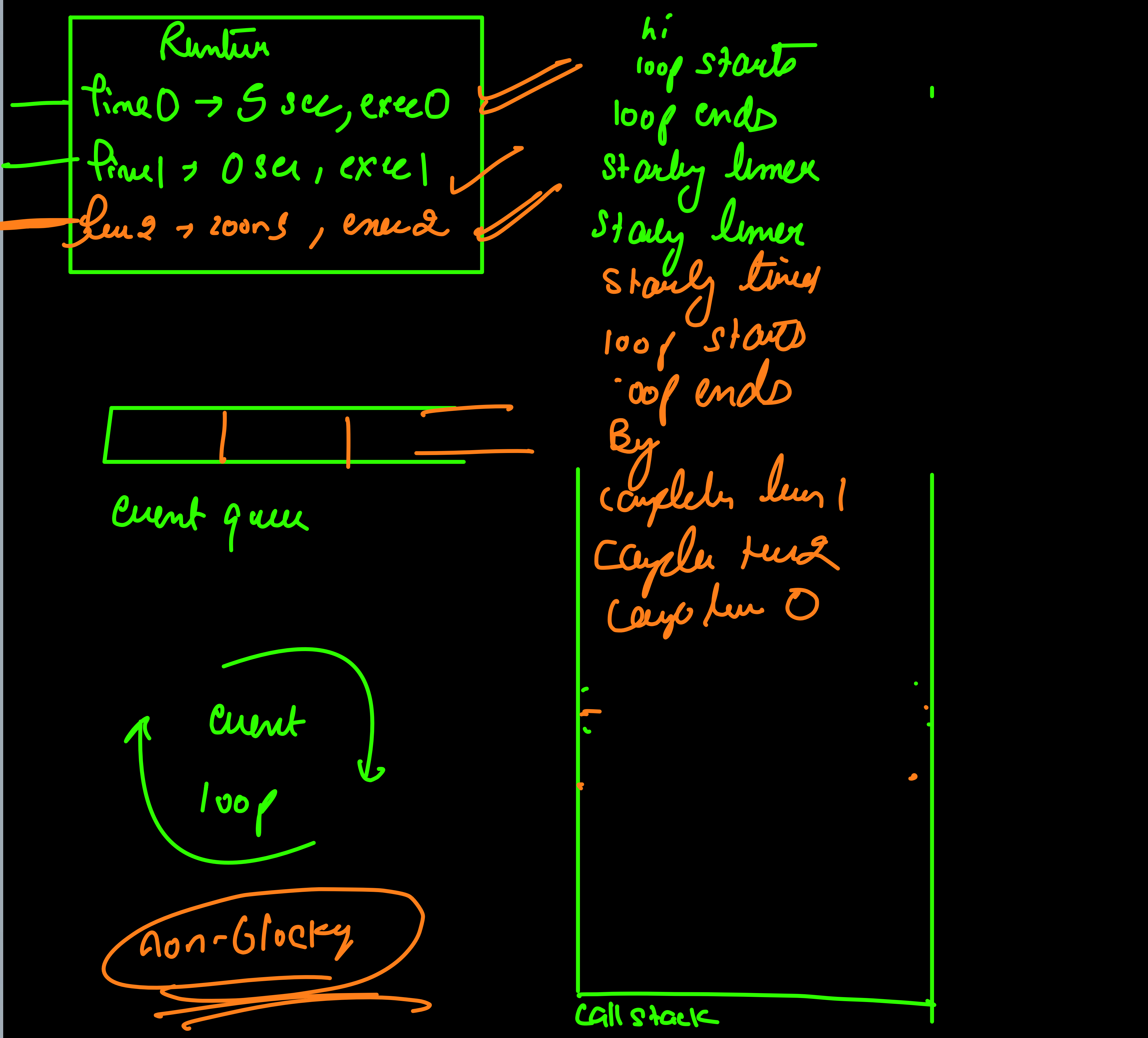


Hi
loop start
loop ends
Starting timer
loop starts
loop ends
By
completed tim

it keeps on checking whether the call stack is empty or not & no global code is left.



```
function timeConsumingByLoop() {
    console.log("loop starts");
    for(let i = 0; i < 10000000000; i++) {
        // some task
    }
    console.log("loop ends");
}
function timeConsumingByRuntimeFeature0() {
    console.log("Starting timer");
    setTimeout(function exec() {
        console.log("Completed the timer0");
        for(let i = 0; i < 10000000000; i++) {
            // some task
        }
    }, 5000); // 5 sec timer
}
function timeConsumingByRuntimeFeature1() {
    console.log("Starting timer");
    setTimeout(function exec() {
        console.log("Completed the timer1");
    }, 0); // 0 s timer
}
function timeConsumingByRuntimeFeature2() {
    console.log("Starting timer");
    setTimeout(function exec() {
        console.log("Completed the timer2");
    }, 200); // 200 ms timer
}
console.log("Hi");
timeConsumingByLoop();
timeConsumingByRuntimeFeature0();
timeConsumingByRuntimeFeature1();
timeConsumingByRuntimeFeature2();
timeConsumingByLoop();
console.log("By");
```



hi
loop starts
loop ends
starting timer
starting timer
starting timer
loop starts
loop ends
By
completed timer1
completed timer2
completed timer 0