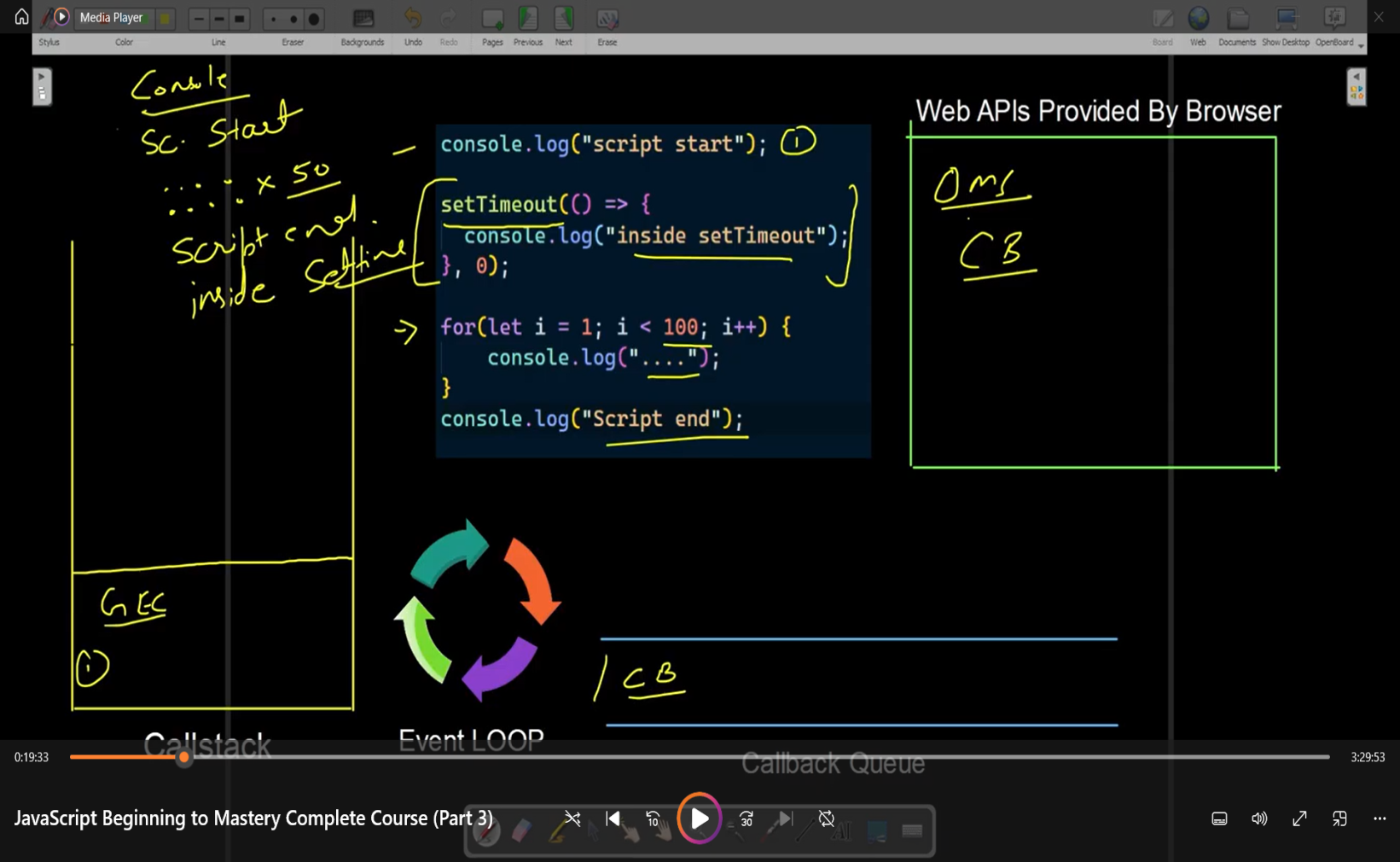


// webBrowser provides the setTimeout() function

// setTimeout() is not inside the js

In above code, when javascript encounter setTimeout then it will send the setTimeout to browser till then js will run its remaining code and after 1sec browser will pass the callback function(here it is arrow function inside setTimeout) inside callback queue. Now Event loop will check whether callstack is clear or not if it is clear event loop will send the callback function inside callstack. And then that callback function will be executed.



Time written inside the setTimeout function is the minimum time this time can be greater than minimum time but not lesser than.

In above code,

1st -> script start will be printed

2nd -> js will encounter setTimeout then js will send it to the browser and after 0ms browser will send setTimeout callback function to the callback queue. Now, event loop will stop the callback function and first check whether callstack is empty or not if the callback is empty then event loop will send the callback function to callstack for execution

3rd -> js will execute the for loop

4th -> script end will be printed

5th -> inside setTimeout will be printed

console.log("script start");

setTimeout(() => console.log("hello world"),1000);

setTimeout(() => console.log("Inside setTimeout"),0);

for(let i=0; i<100; i++){

    console.log("inside for loop");

}

const id = setTimeout(()=>{console.log("Inside setTimeout 3rd")}, 0);

console.log("setTimeout id is", id);

console.log("clearing timeout");

clearTimeout(id);

console.log("script end");

// output:

/\*

script start

100 inside for loop

setTimeout id is 11

clearing timeout

script end

undefined

Inside setTimeout

hello world

\*/