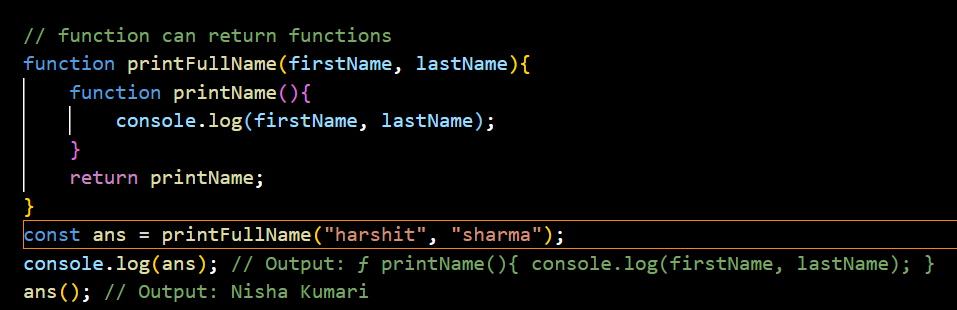
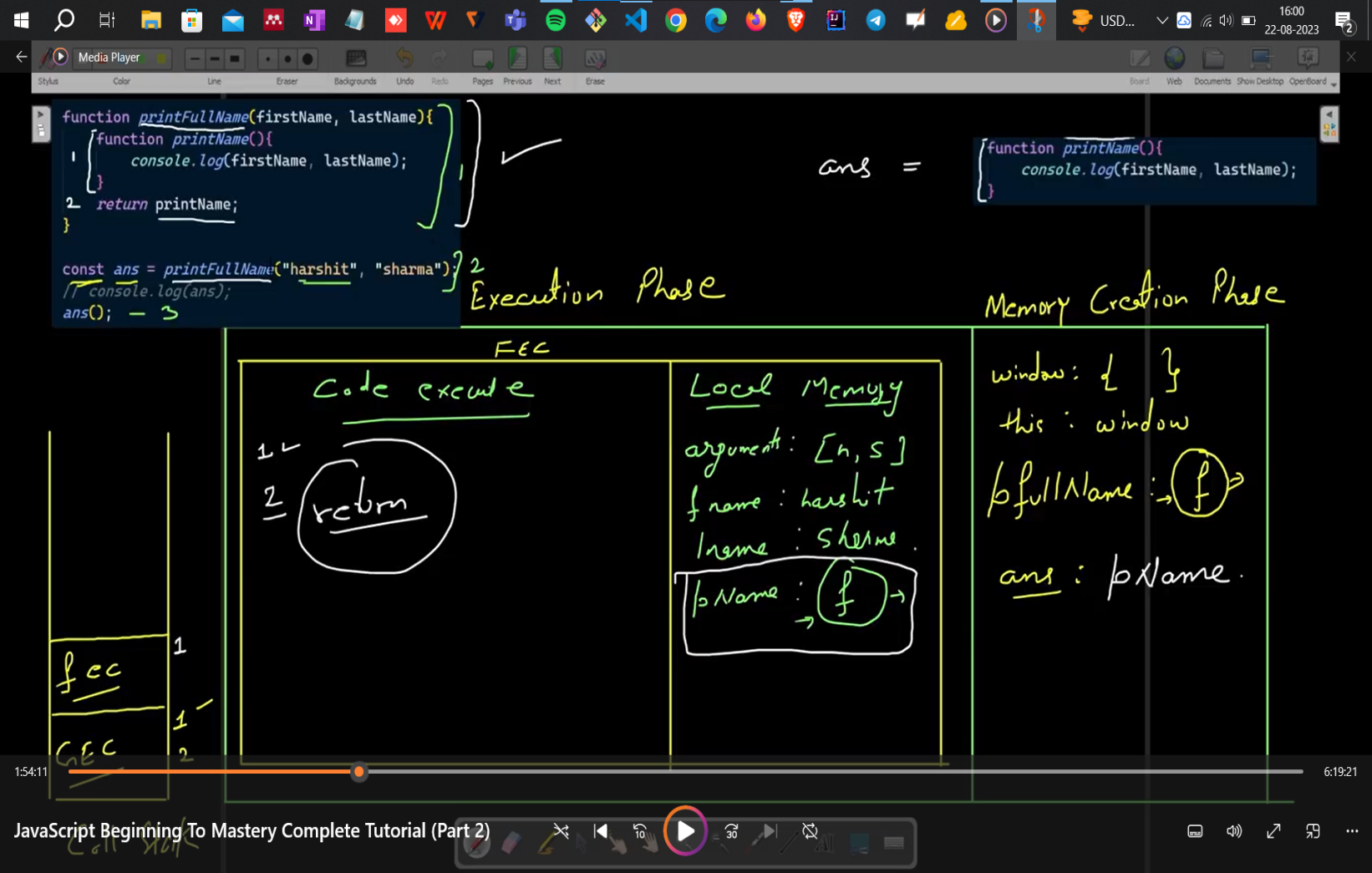
**CLOSURE**



Here 1st line of code is already stored in global memory i.e., function “printFullName”

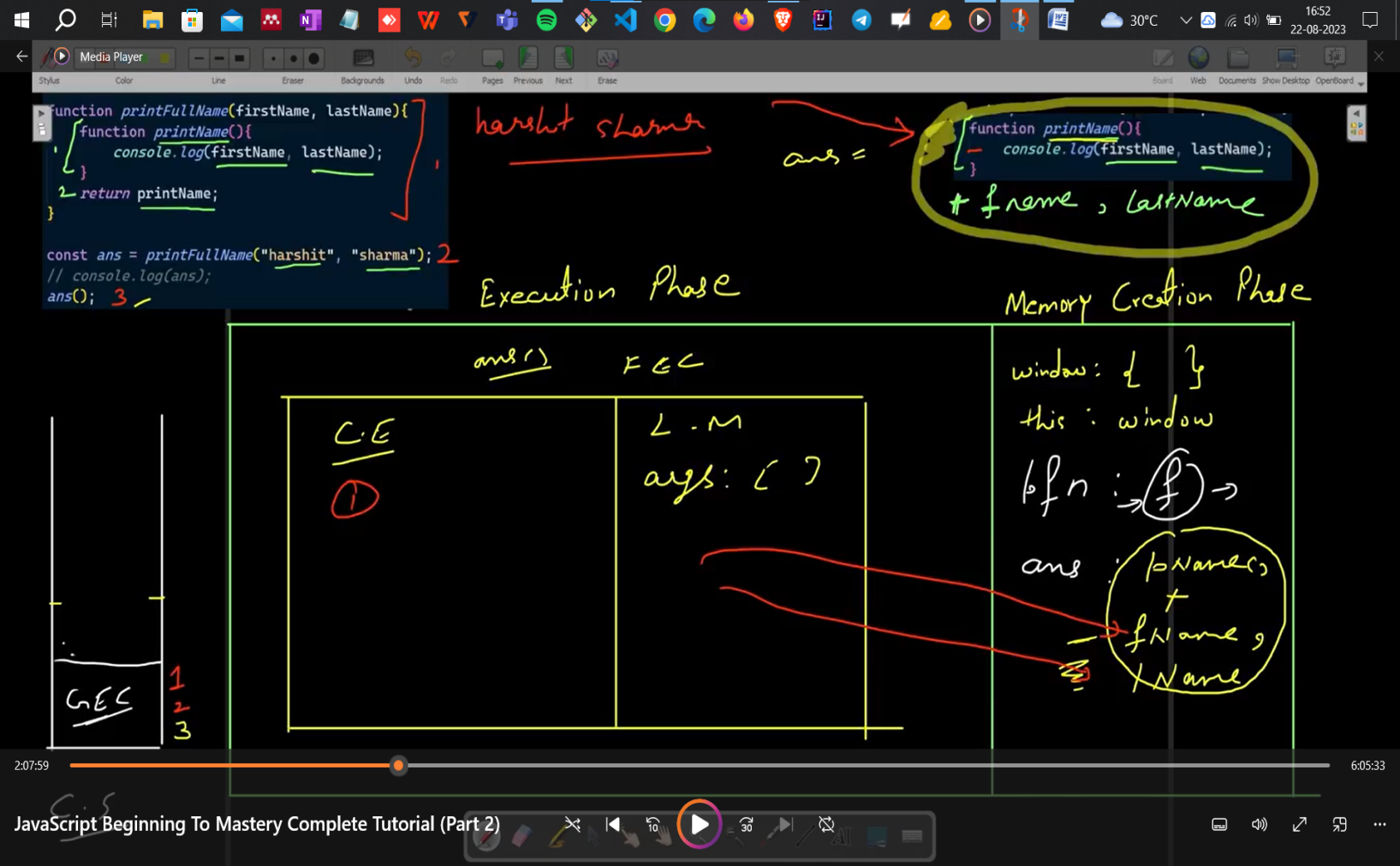
2nd line of code is ans : uninitialized and this 2nd line of code is calling the function “printFullName”. So, FEC of printFullName will be created.

‘ans’ return printName function which is in the 2nd line of code of FEC of “printFullName”.

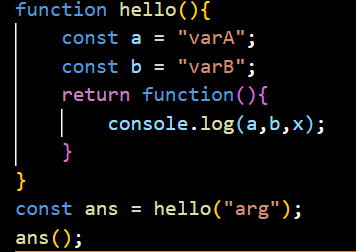


Now there is no use of FEC, it will again shift to GEC

In 3rd line of code again FEC of “ans” will be created in this firstName and lastName neither in local memory of ans() nor in global memory of global execution context.



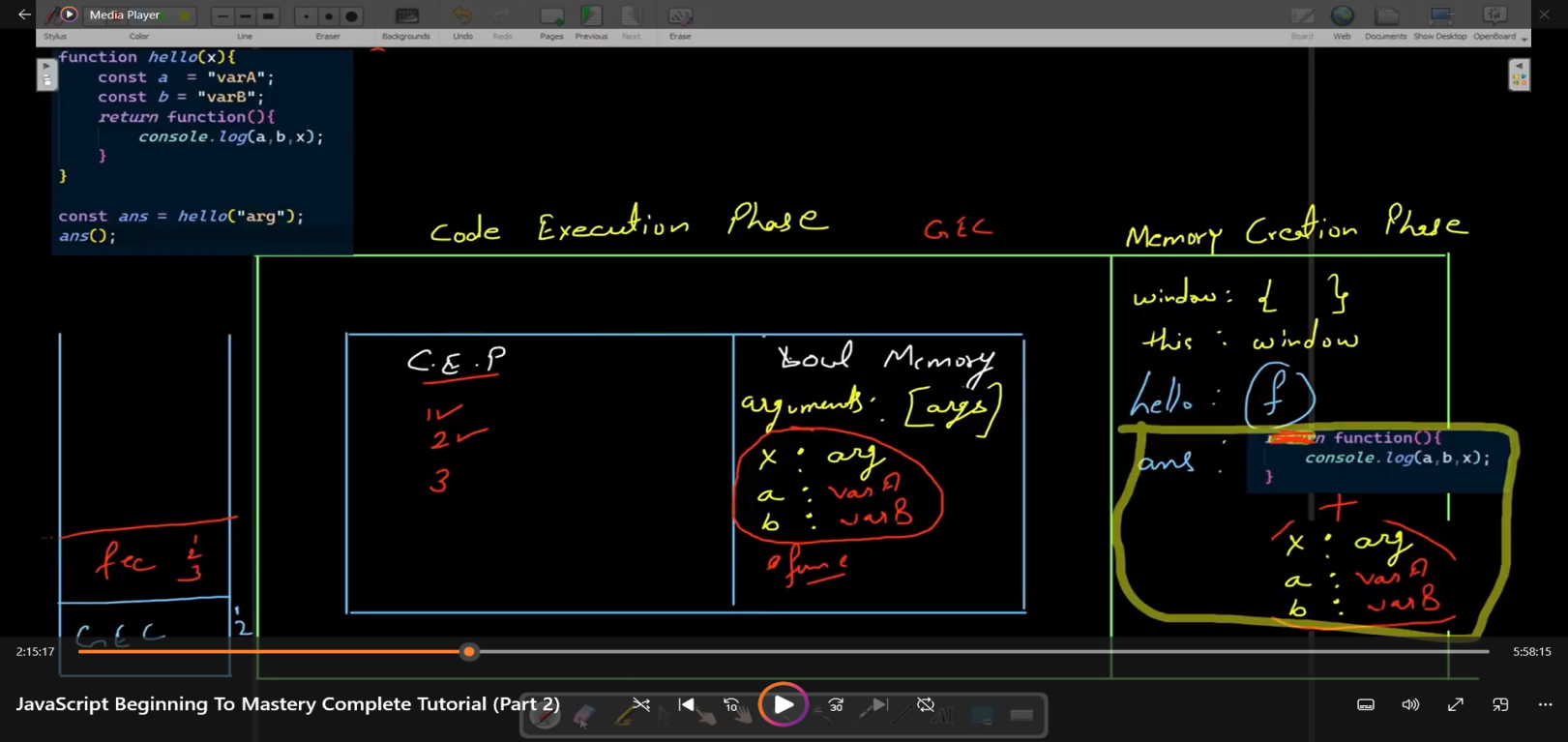
Closure Example 1st



Here in GEC of code of execution, 1st line of code is hello : f

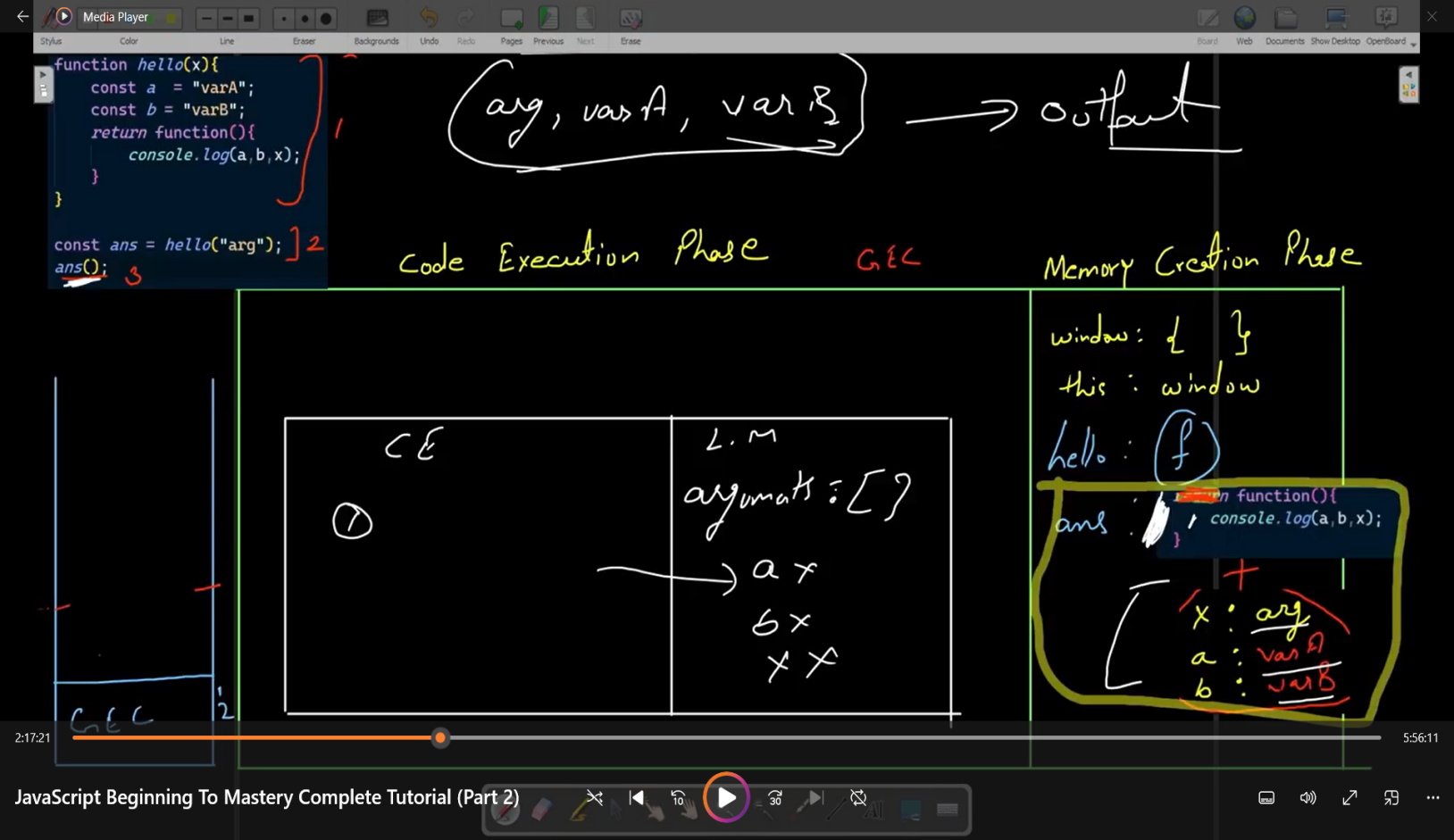
2nd line of code is “ans” is calling function “hello” then FEC of hello will be created.

In local memory of FEC of ‘hello’, ‘a’ and ‘b’ will be uninitialized first

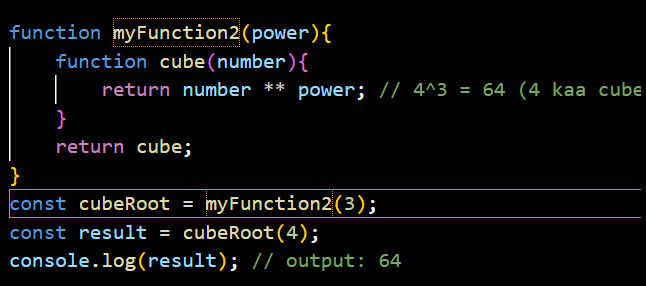


Now in 3rd line of code of GEC, we will call ‘ans()’ and in ‘ans’ innerfunction is stored where a,b,x is not present

3rd line of code of GEC -> function execution context



Closure Example 2nd



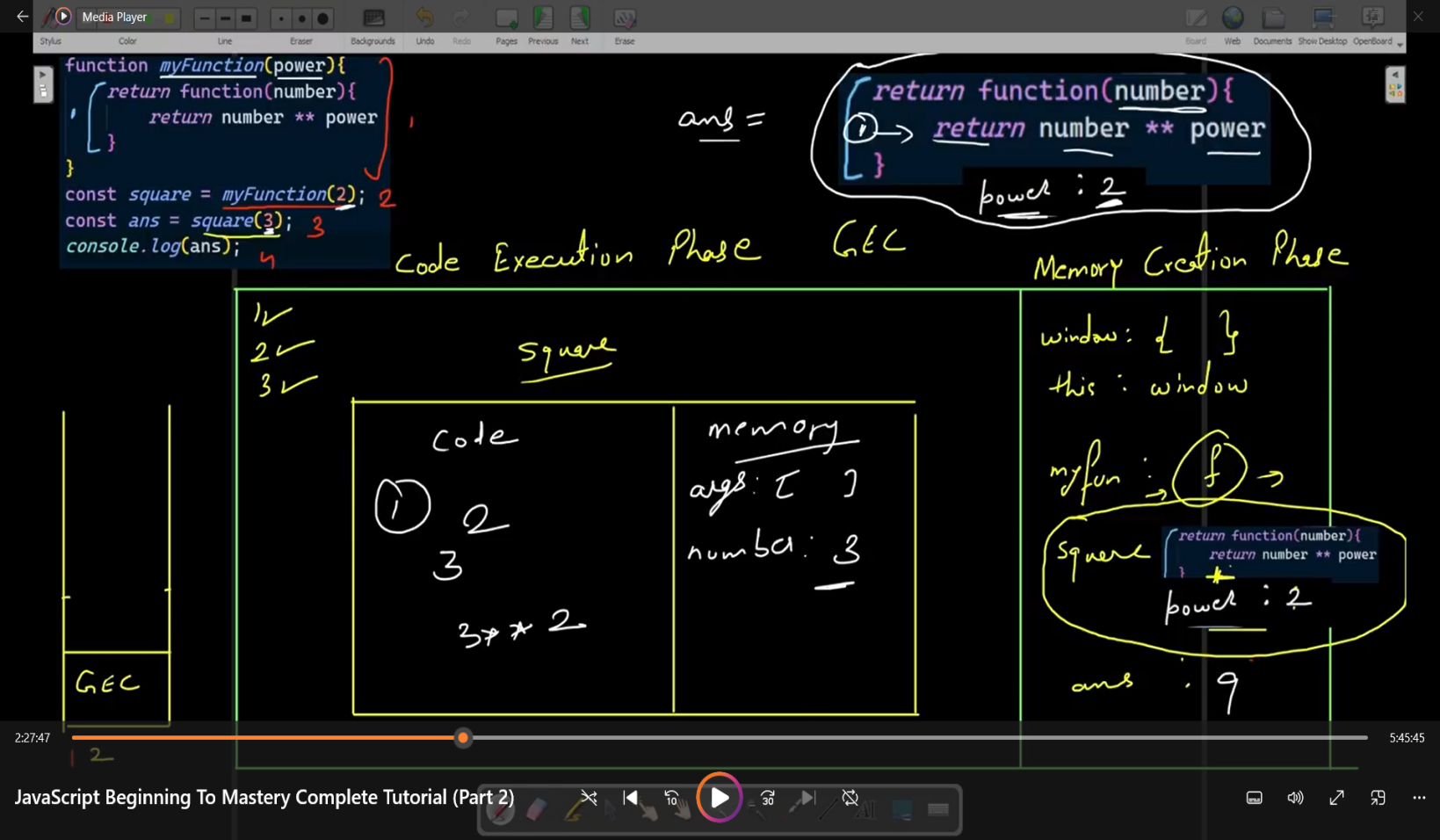
We can write the myFunction2 by using arrow function

const myFunction = power => { const cube = number => number \*\* power; }

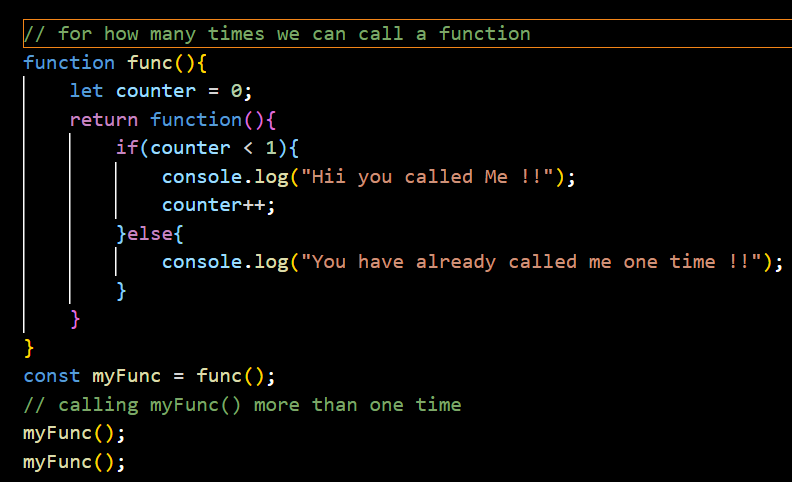
**OR**

const myFunction = power => number => number \*\* power

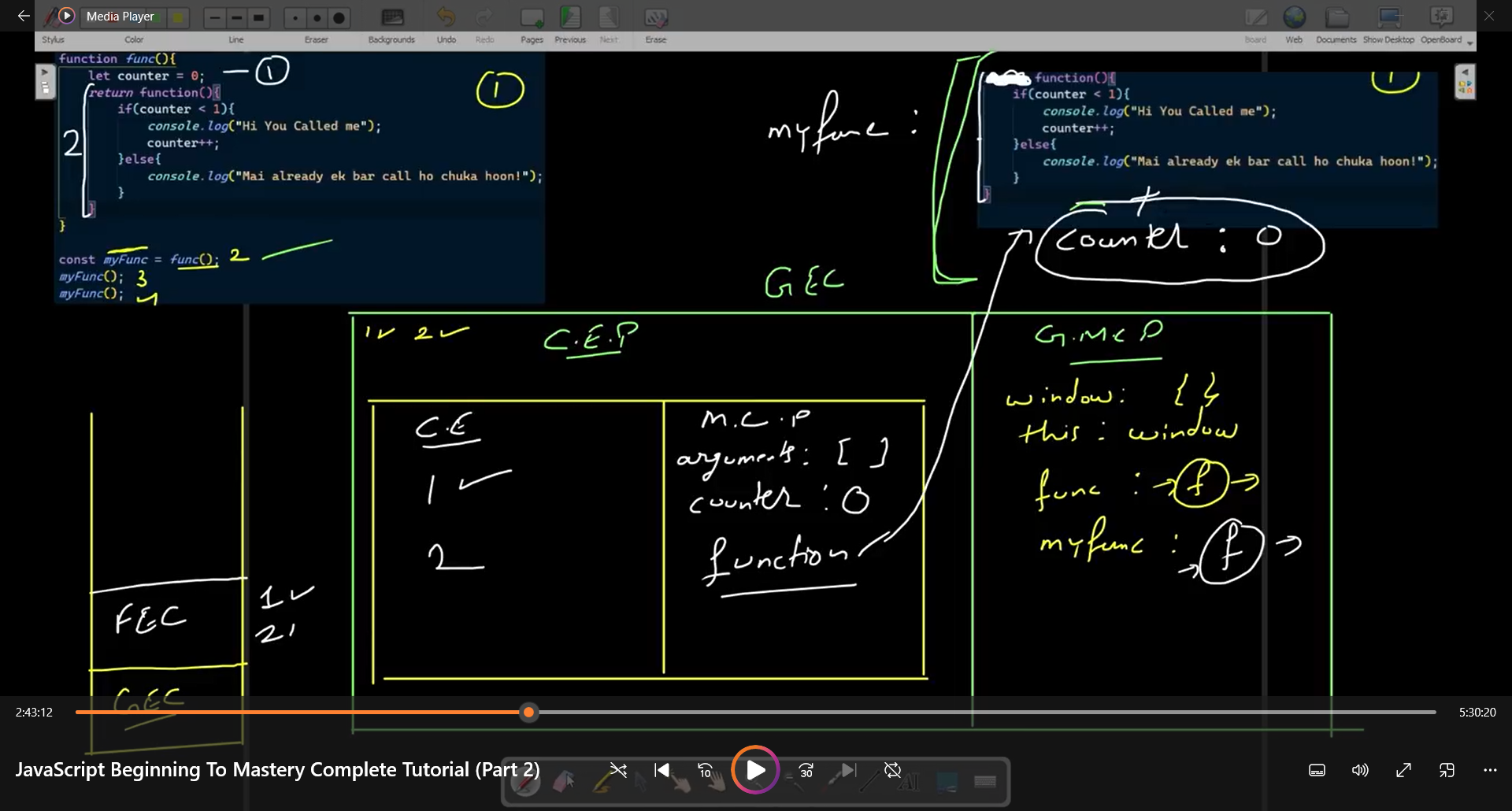




Closure Example 3rd

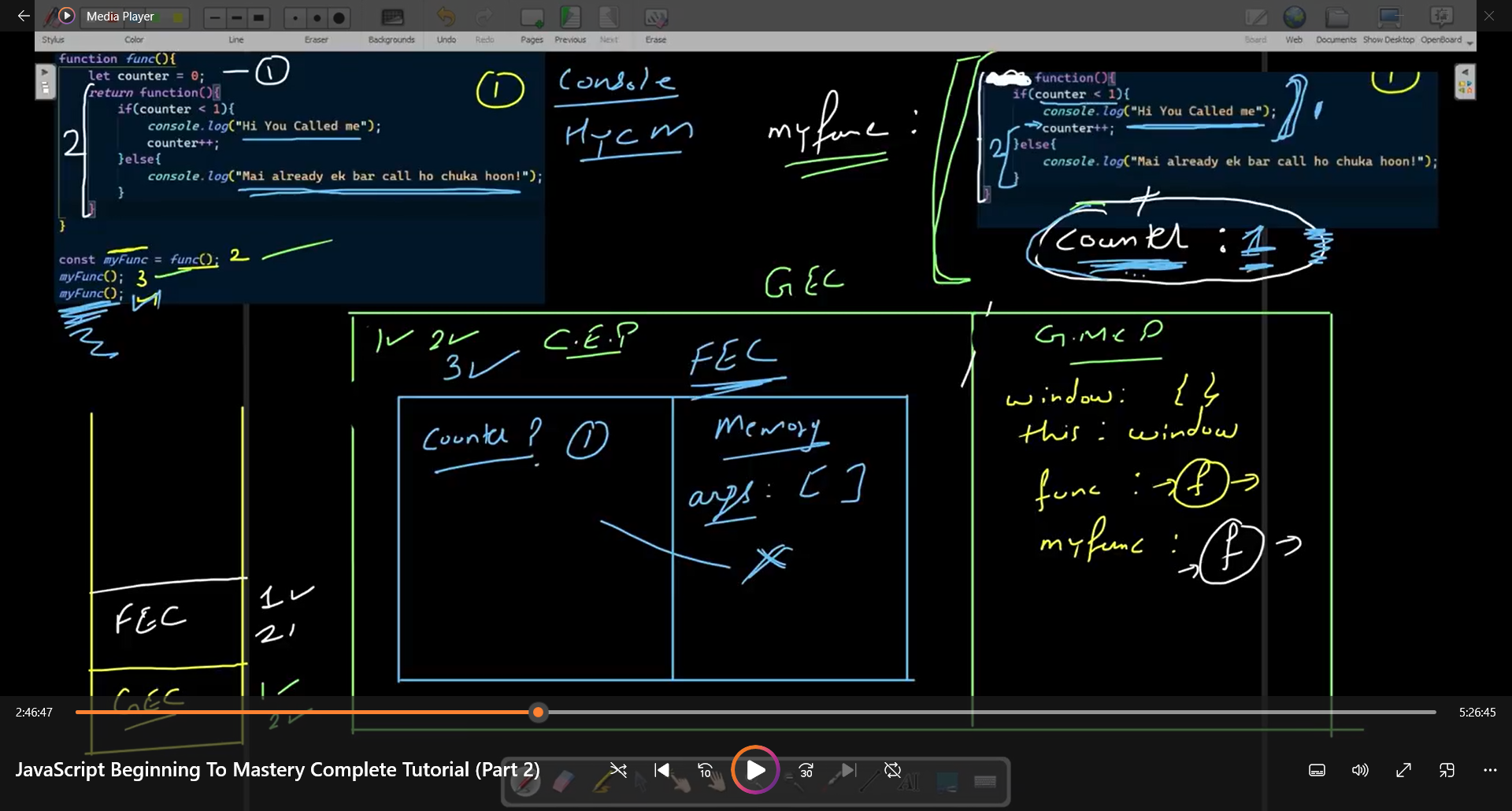


Before code execution the value of ‘myFunc’ and ‘counter’ will be uninitialized.



After code execution the value of myfunc : f() and counter : 0

Now we will call myFunc()



When ‘myFunc’ will be called it has ‘anonymous function’ as well as ‘counter : 0’

1st if code will be checked, whether counter < 1 or not but in local memory of myFunc () there is no counter variable then this is what the closure where it will search in the closure part there counter : 0 is present

--------------> after first time calling myFunc() -> “Hii you called me!!” will be printed

2nd time when we call the myFunc() then the counter value will be incremented to 1 and again it will check whether counter < 1 or not but here counter : 1 so, else part will be executed.

Counter value will only increase when we call the myFunc() again otherwise the counter value will be not increment