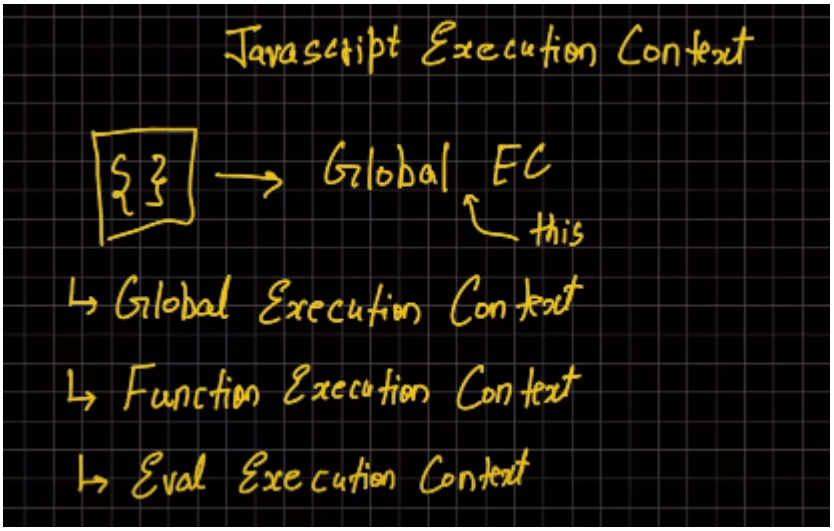
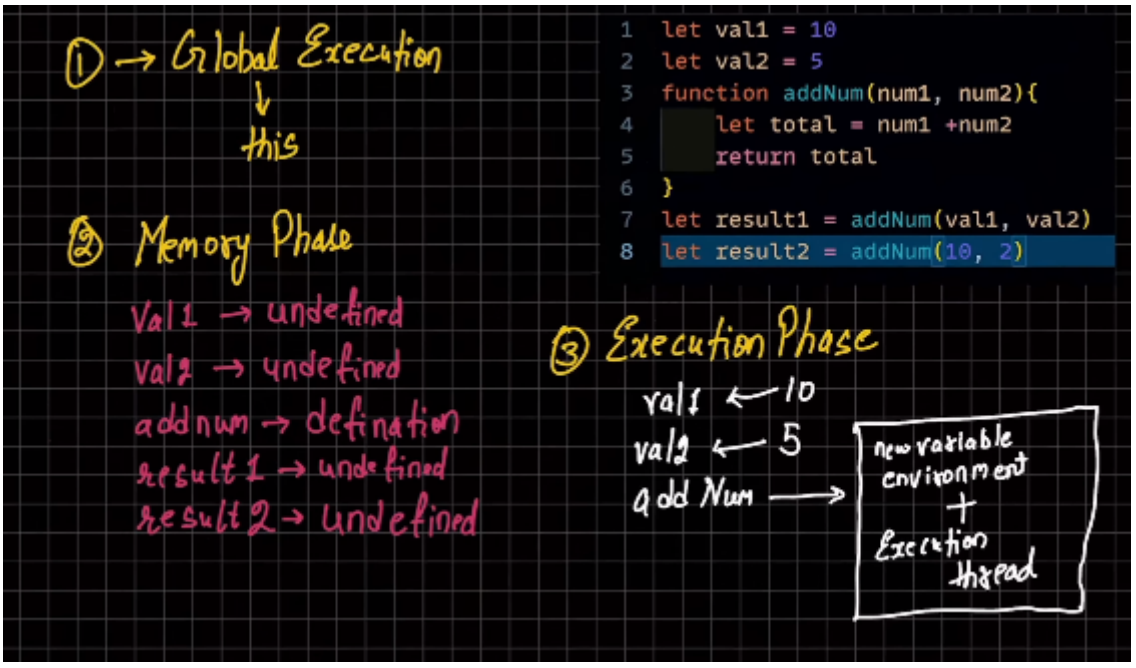
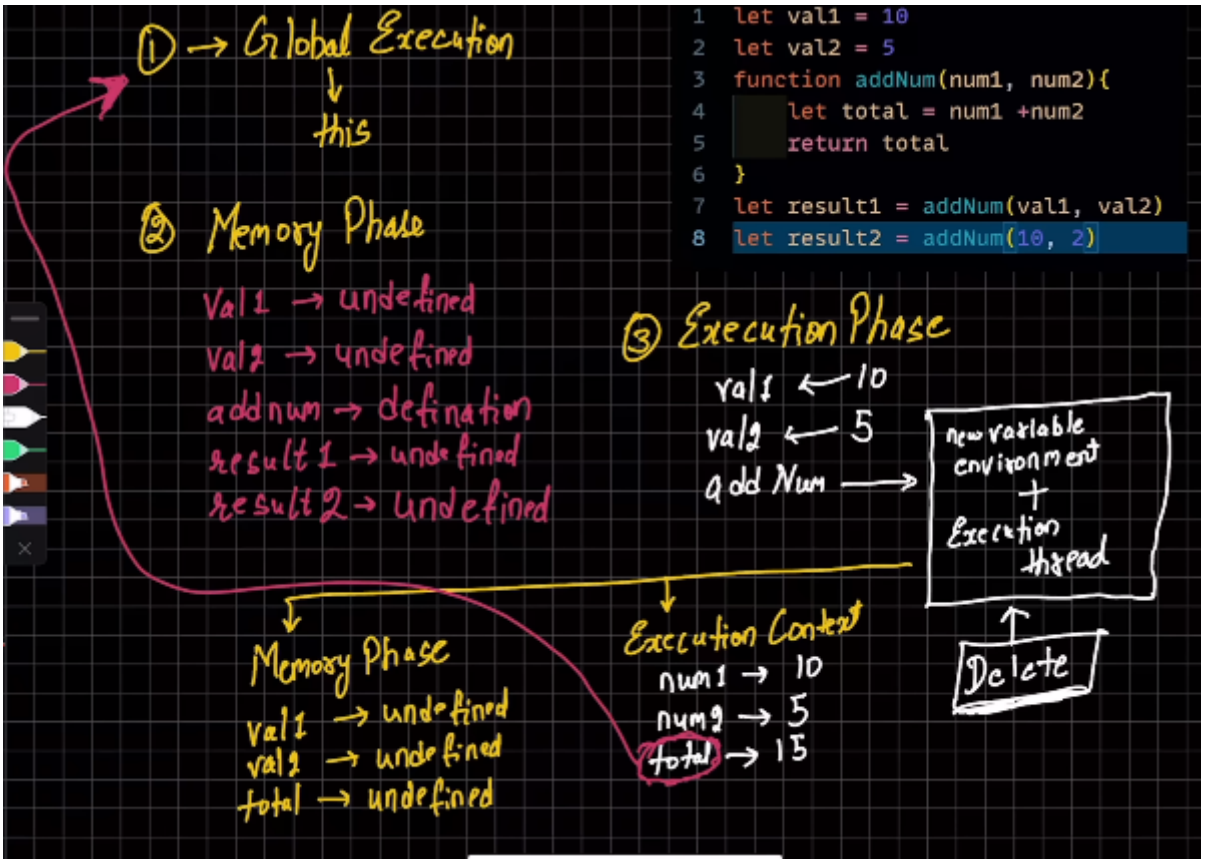
How JS works behind the scenes



Execution Context:

JavaScript uses execution contexts to manage the code's execution. An execution context is an environment in which the code is evaluated and executed. There are three types of execution contexts: Global, Function, and Eval.


① → Global Execution
↓
this
② Memory Phase
val1 → undefined
val2 → undefined
addnum → defination
result 1 → undefined
result 2 → undefined
1 let val1 = 10
2 let val2 = 5
3 function addNum(num1, num2){
4 let total = num1 +num2
5 return total
6 }
7 let result1 = addNum(val1, val2)
8 let result2 = addNum(10, 2)
③ Execution Phase
val1 ← 10
val2 ← 5
add Num →
new variable environment + Execution thread


① → Global Execution
↓
this
② Memory Phase
val1 → undefined
val2 → undefined
addnum → defination
result 1 → undefined
result 2 → undefined
1 let val1 = 10
2 let val2 = 5
3 function addNum(num1, num2){
4 let total = num1 +num2
5 return total
6 }
7 let result1 = addNum(val1, val2)
8 let result2 = addNum(10, 2)
③ Execution Phase
val1 ← 10
val2 ← 5
add Num →
new variable environment + Execution thread
Memory Phase
val1 → undefined
val2 → undefined
total → undefined
Execution Context
num1 → 10
num2 → 5
total → 15
Delete