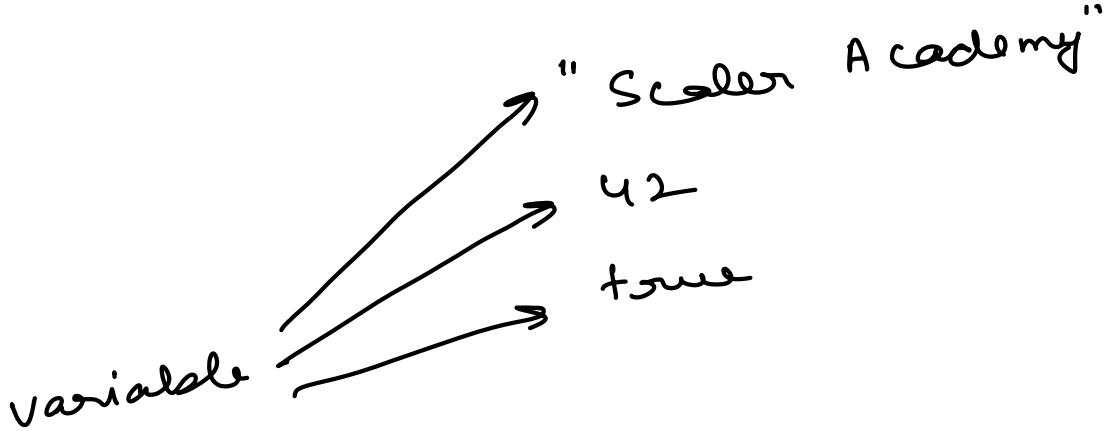
Agenda

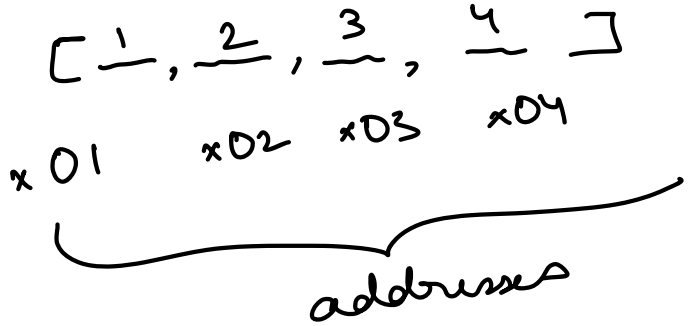
- → Functions
- → Execution Context
- → Hoisting and TDZ

Functions

The function is an abstract body and inside that body a particular logic is written and function accepts parameters to operate.

serve Beverage



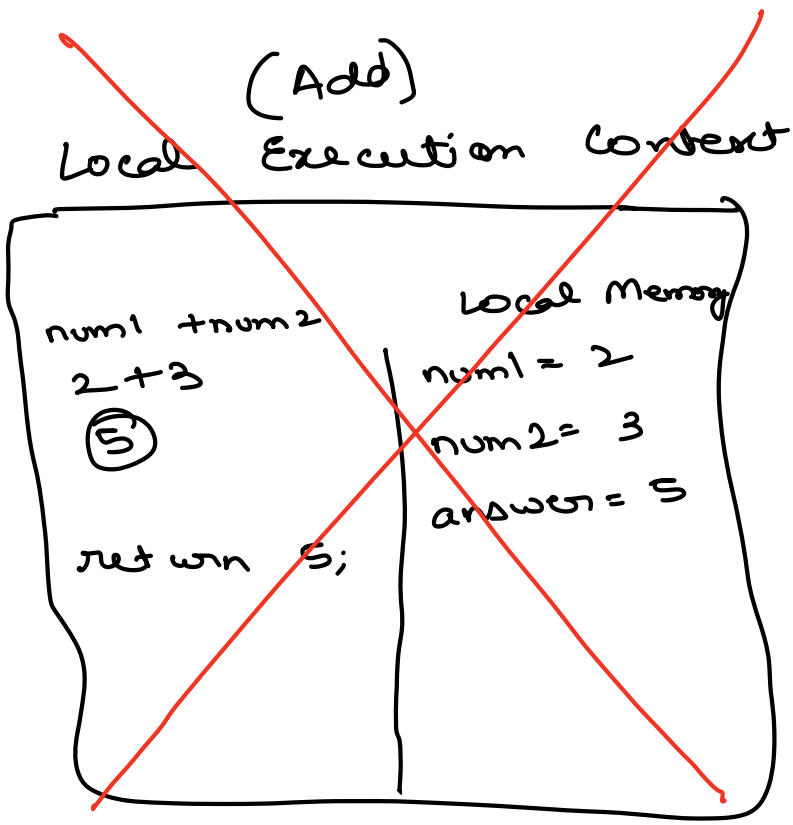


Execution context

When the JS Engine scans the JS file, it makes an environment called Execution context that handles the entire transformation and execution of code.

```
1   // Example 1
2   var a = 2;
3   var b = 3;
4
5   function add(num1, num2) {
6     var answer = num1 + num2;
7     return answer;
8   }
9
10  var sum = add(a, b);
11  console.log(sum); // 5
12
```

destroyed after use



Area B
Call Stack

Area A
Global memory
a = 2
b = 3
add = Fn
sum = 5

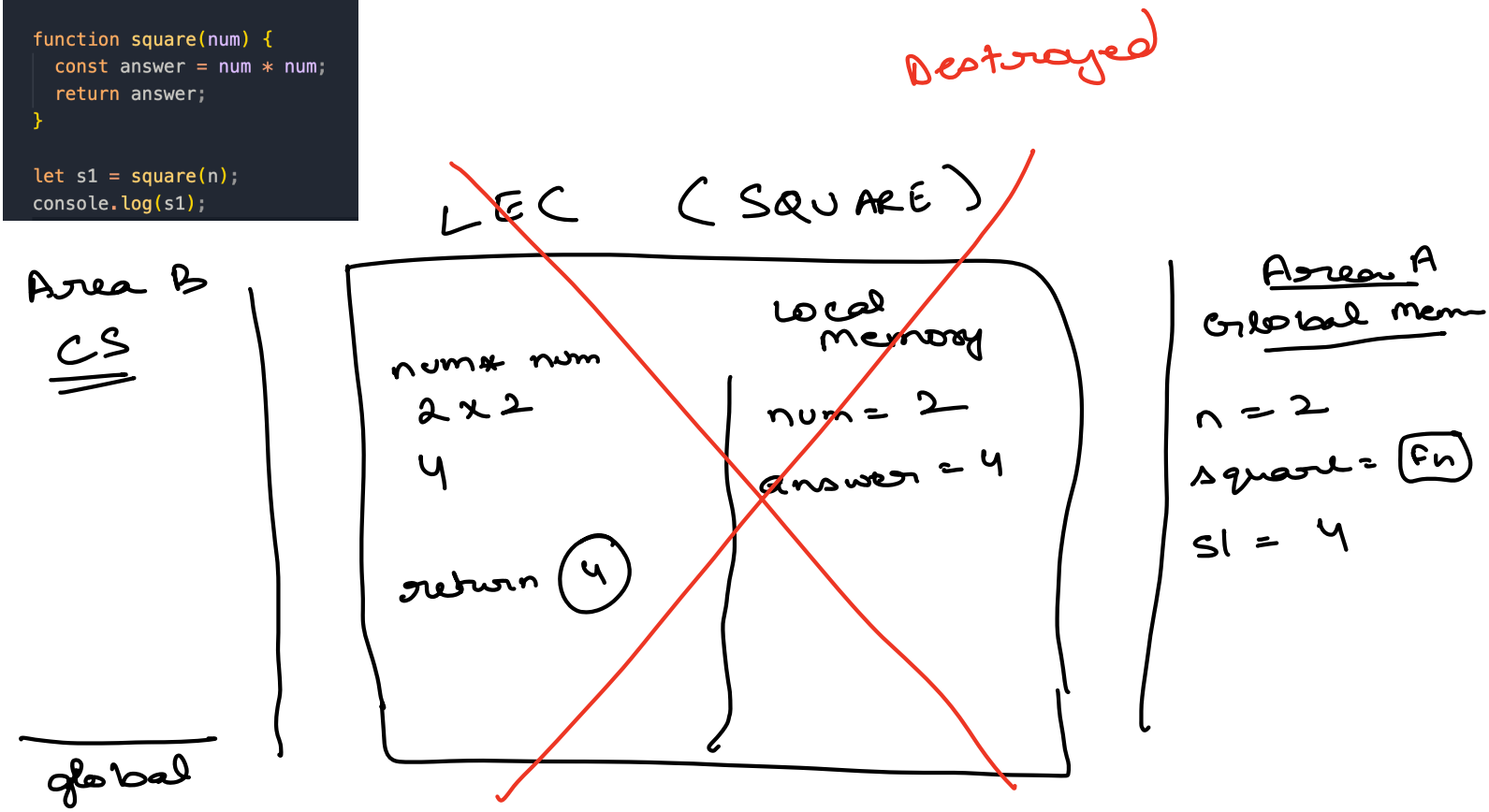
global

Call Stack:- A call stack is a mechanism that JS uses to track what function is currently running.

```js
// Exmaple 2
const n = 2;

function square(num) {
  const answer = num * num;
  return answer;
}

let s1 = square(n);
console.log(s1);
```



Hoisting

Hoisting is a behavior in JS language through which functions, variables, etc are moved to the top of the file at start of the program before the actual parsing begins.