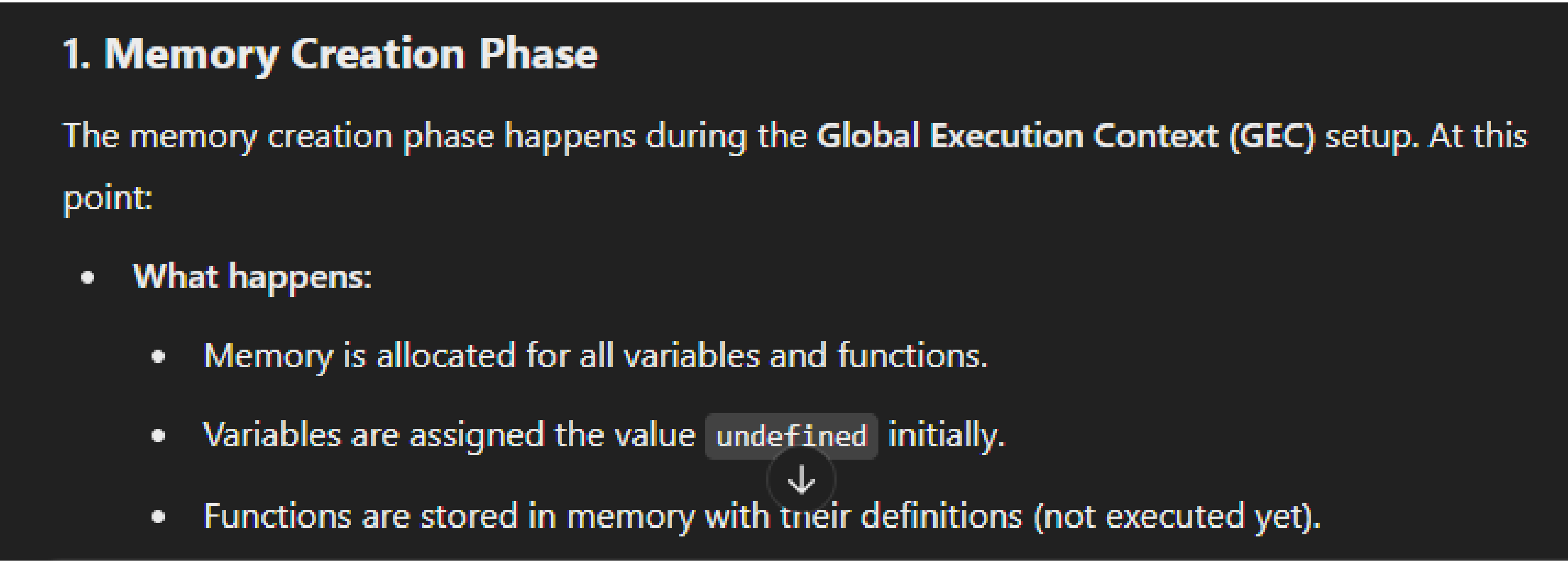
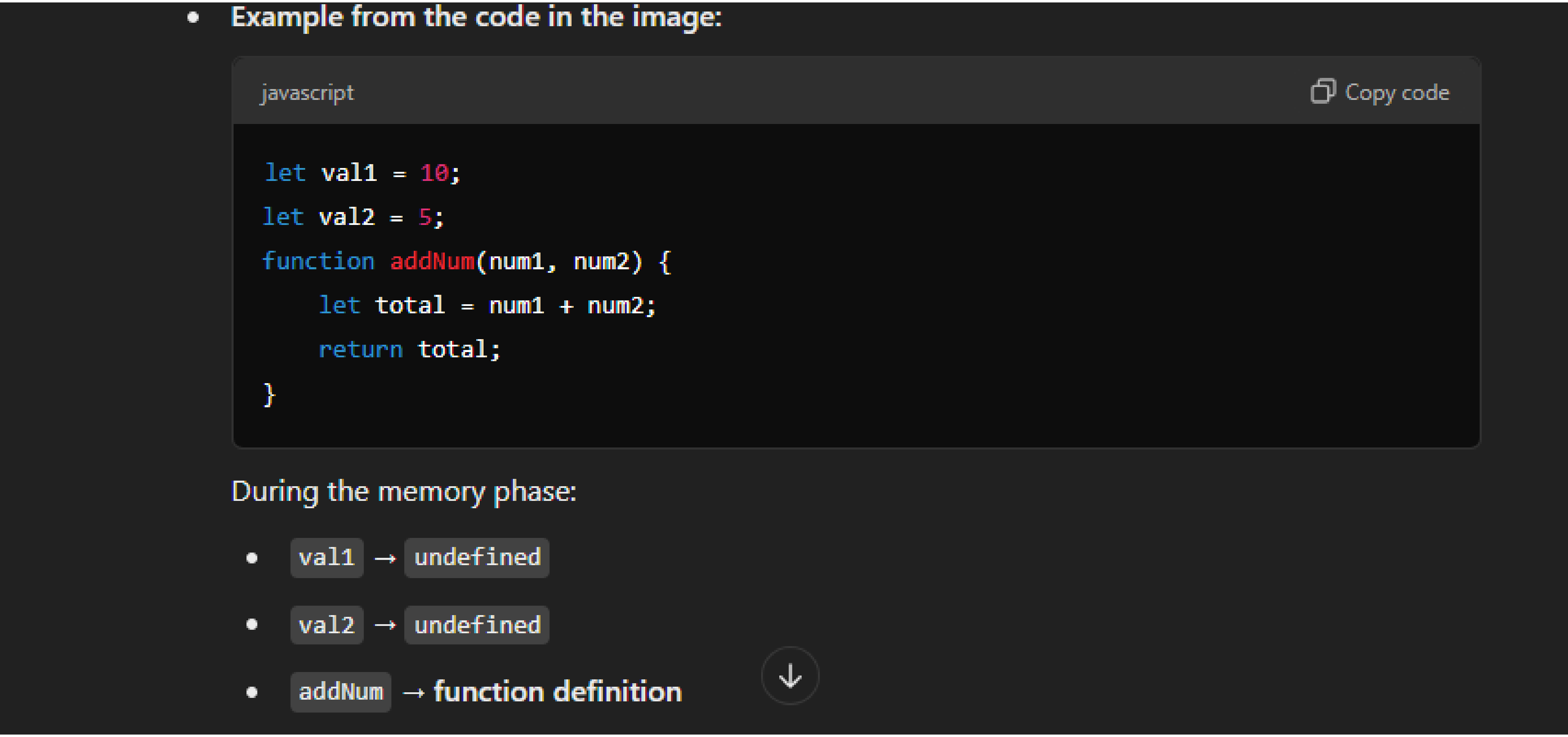


1. Memory Creation Phase

The memory creation phase happens during the **Global Execution Context (GEC)** setup. At this point:

- **What happens:**
    - Memory is allocated for all variables and functions.
    - Variables are assigned the value `undefined` initially.
    - Functions are stored in memory with their definitions (not executed yet).



- **Example from the code in the image:**

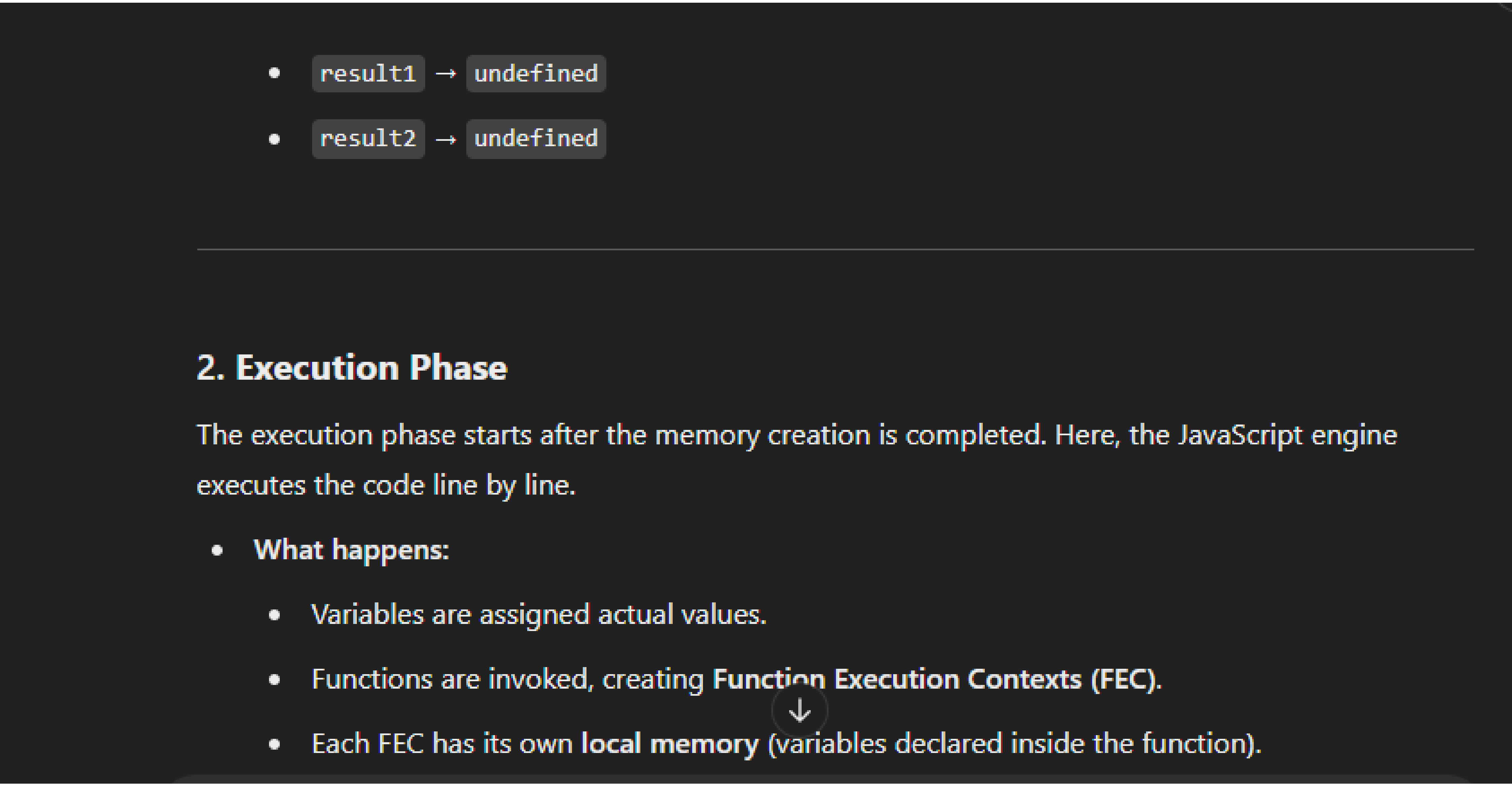
javascript

Copy code

```javascript
let val1 = 10;
let val2 = 5;
function addNum(num1, num2) {
    let total = num1 + num2;
    return total;
}
```

During the memory phase:

- `val1` → `undefined`
- `val2` → `undefined`
- `addNum` → **function definition**



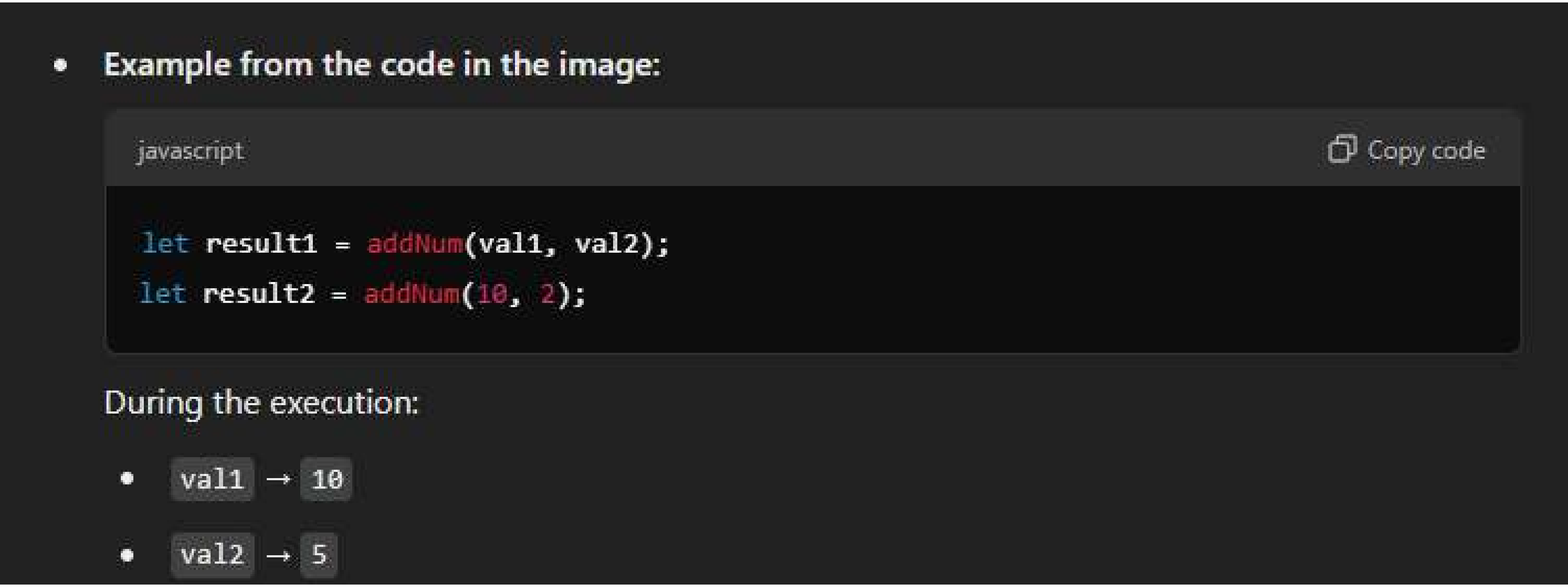
    - `result1` → `undefined`
    - `result2` → `undefined`

---

2. Execution Phase

The execution phase starts after the memory creation is completed. Here, the JavaScript engine executes the code line by line.

- **What happens:**
    - Variables are assigned actual values.
    - Functions are invoked, creating **Function Execution Contexts (FEC)**.
    - Each FEC has its own **local memory** (variables declared inside the function).

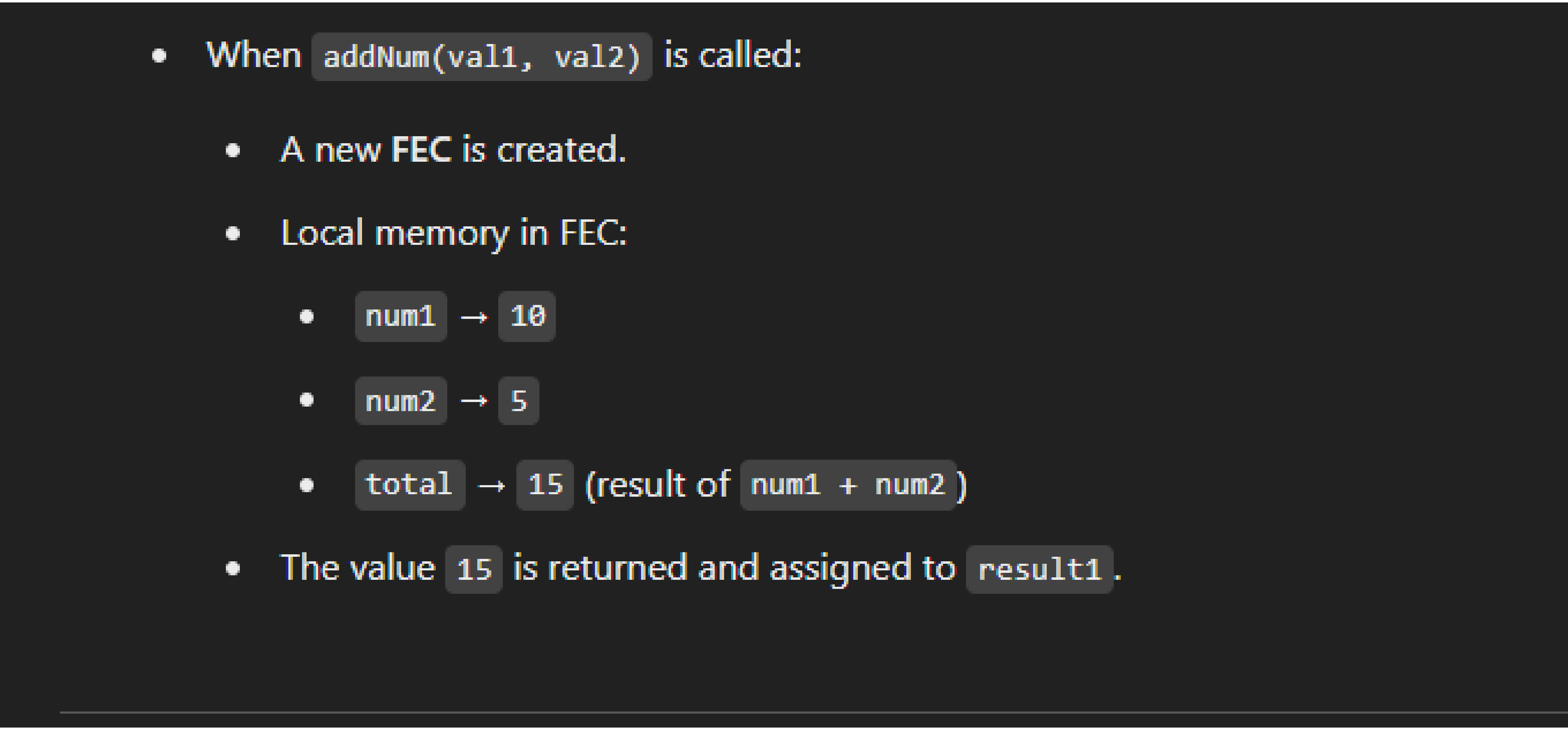


- **Example from the code in the image:**

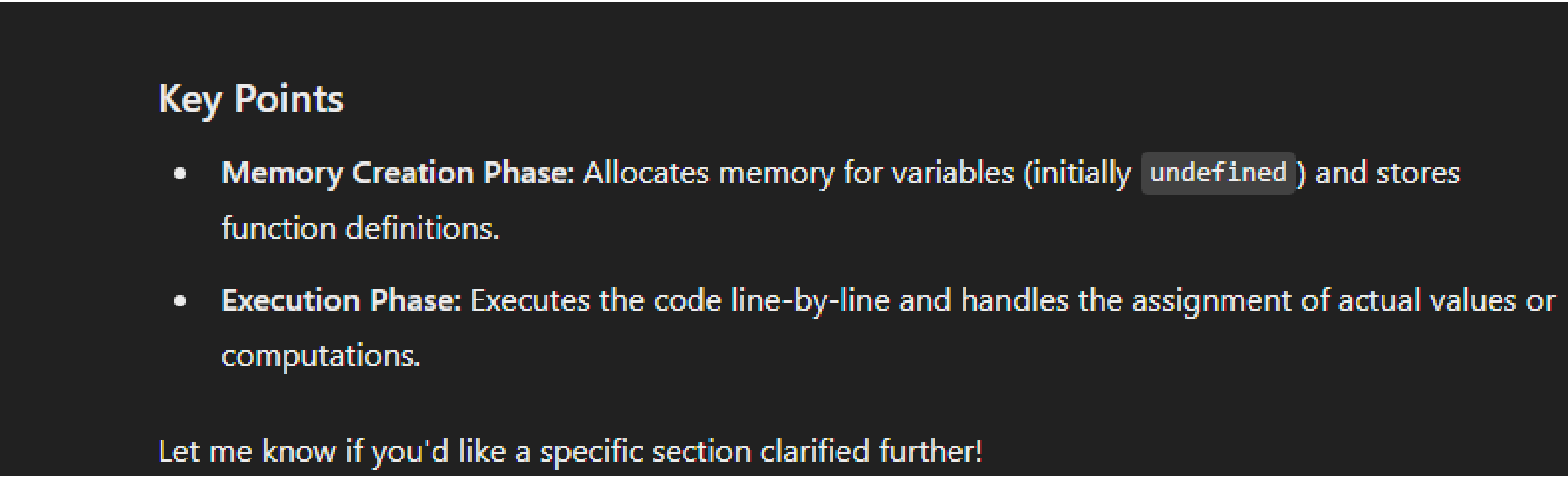
```javascript
let result1 = addNum(val1, val2);
let result2 = addNum(10, 2);
```

During the execution:

- `val1` → `10`
- `val2` → `5`



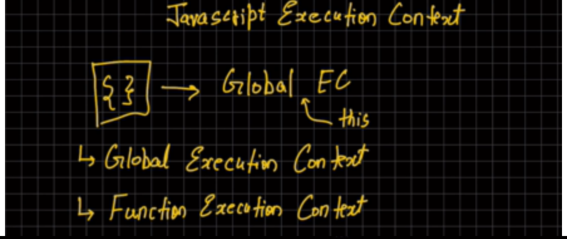
- When `addNum(val1, val2)` is called:
  - A new **FEC** is created.
  - Local memory in FEC:
    - `num1` → `10`
    - `num2` → `5`
    - `total` → `15` (result of `num1 + num2`)
  - The value `15` is returned and assigned to `result1`.



Key Points

- **Memory Creation Phase:** Allocates memory for variables (initially `undefined`) and stores function definitions.
- **Execution Phase:** Executes the code line-by-line and handles the assignment of actual values or computations.

Let me know if you'd like a specific section clarified further!

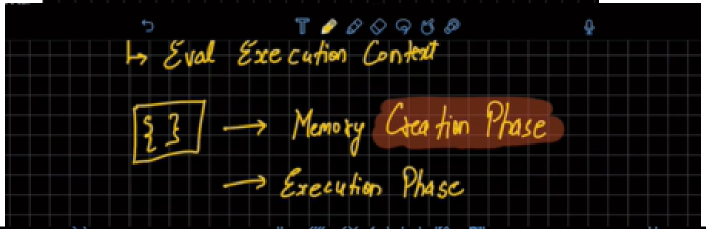


Javascript Execution Context

{ } → Global EC ↳ this

↳ Global Execution Context

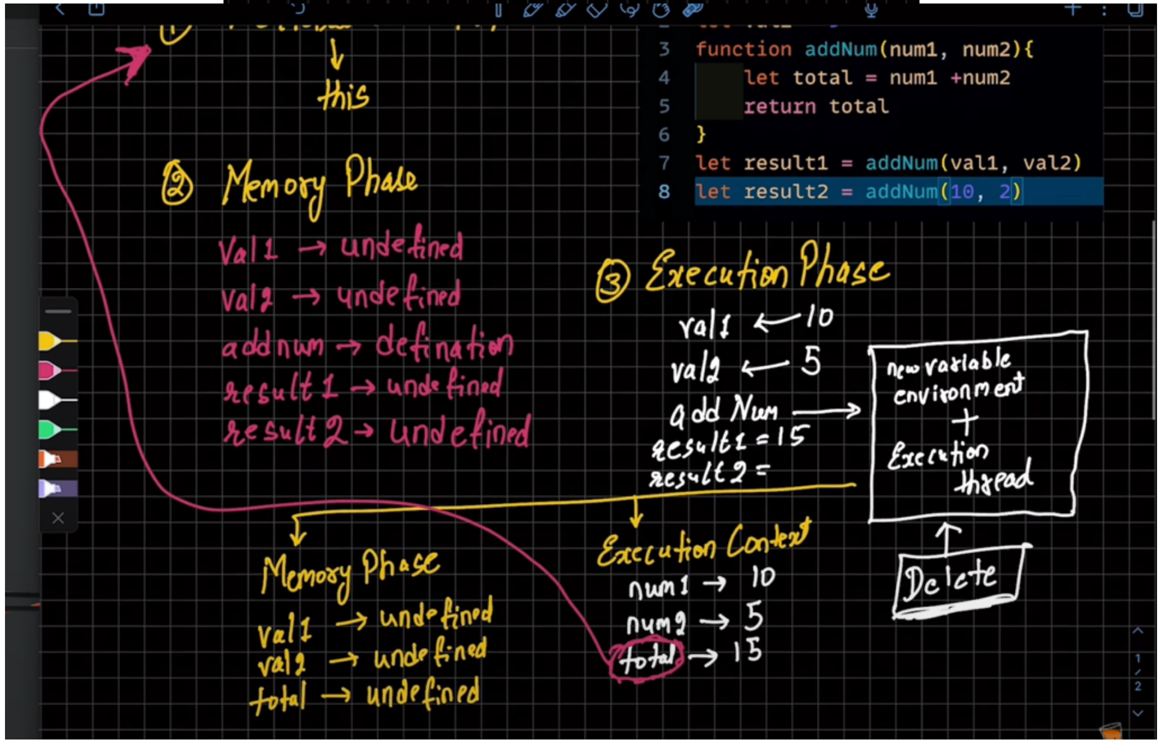
↳ Function Execution Context



↳ Eval Execution Context

{ } → Memory Creation Phase

→ Execution Phase



```
function addNum(num1, num2){
    let total = num1 +num2
    return total
}
let result1 = addNum(val1, val2)
let result2 = addNum(10, 2)
```

↓ this

② Memory Phase

Val1 → undefined

val2 → undefined

addnum → defination

result1 → undefined

result2 → undefined

③ Execution Phase

val1 ← 10

val2 ← 5

add Num → new variable environment + Execution thread

result1 = 15

result2 =

Memory Phase

val1 → undefined

val2 → undefined

total → undefined

Execution Context

num1 → 10

num2 → 5

total → 15

Delete