



What happens when you run JavaScript code?

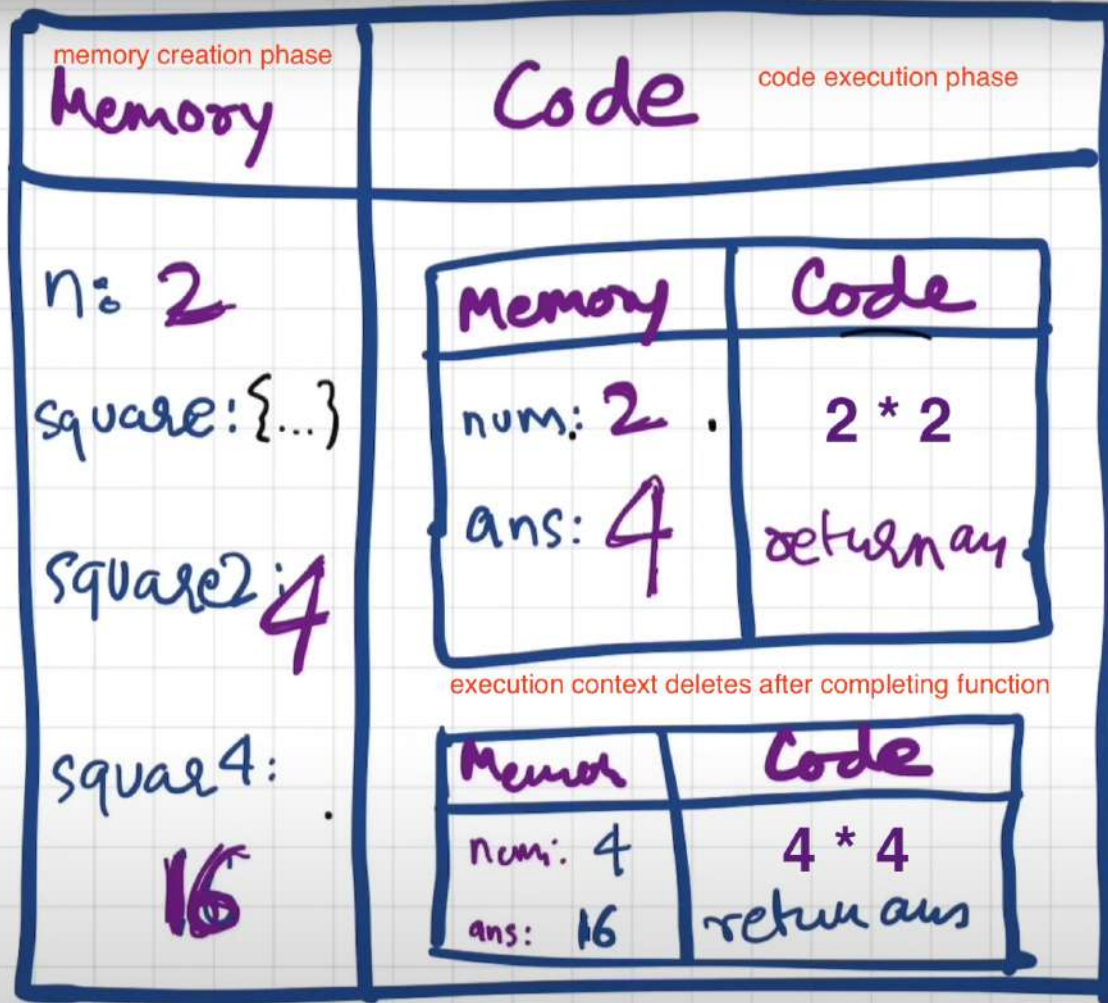


| Memory | Code |
|----------------------------------|------|
| n : undefined square: {...} | |
| square2: undefined | |
| square4: undefined | |



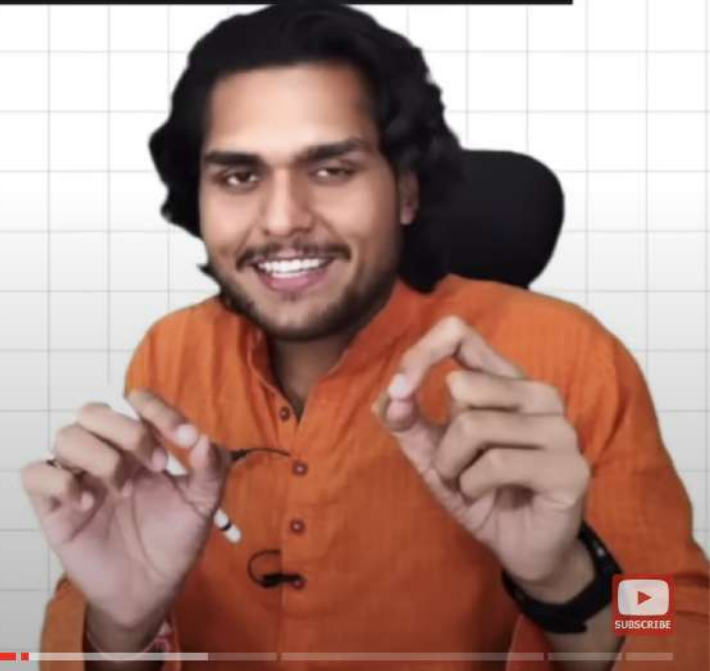
```
1 var n = 2;  
2 function square (num) {  
3     var ans = num * num;  
4     return ans;  
5 }  
6 var square2 = square(n);  
7 var square4 = square(4);
```





```

1 var n = 2;
2 function square (num) {
3     var ans = num * num;
4     return ans;
5 }
6 var square2 = square(n);
7 var square4 = square(4);
    
```



Call Stack



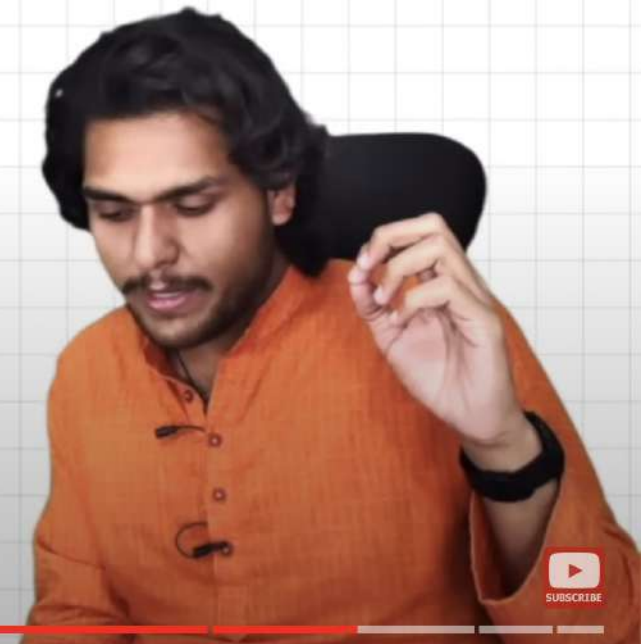
| Memory | Code |
|---------------|------|
| n: 2 | |
| square: {...} | |
| square2: 4 | |
| square4: 16 | |

| Memory | Code |
|--------|------------|
| num: 2 | |
| ans: 4 | return ans |

| Memory | Code |
|---------|------------|
| num: 4 | |
| ans: 16 | return ans |

```

1 var n = 2;
2 function square (num) {
3     var ans = num * num;
4     return ans;
5 }
6 var square2 = square(n);
7 var square4 = square(4);
    
```



“Call stack maintains the
order of execution of
execution contexts”



0. Call Stack

1. Execution Context Stack

2. Program Stack

3. Control Stack

4. Runtime Stack

5. Machine Stack

