EP. 9

THE FRONTEND INTERVIEW

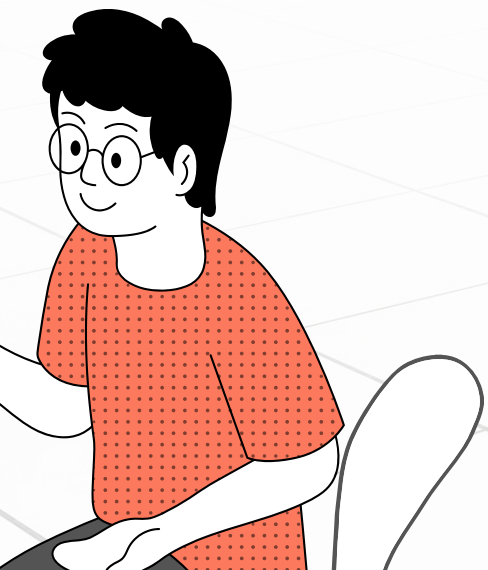
@frontendhabibi

Alright, let's take another code snippet. You have to tell me the result it will print.

Okay...

@frontendhabibi

```
const fruits = ['mango', 'orange', 'banana']
console.log(fruits)
fruits.push('apple')
console.log(fruits)
```



This code first initializes an array fruits with items as 'mango', 'orange', and 'banana' so the same array will be printed with it's elements, in the third line a new item 'apple' will be added to the existing fruits array and the updated array will be printed in the next line.

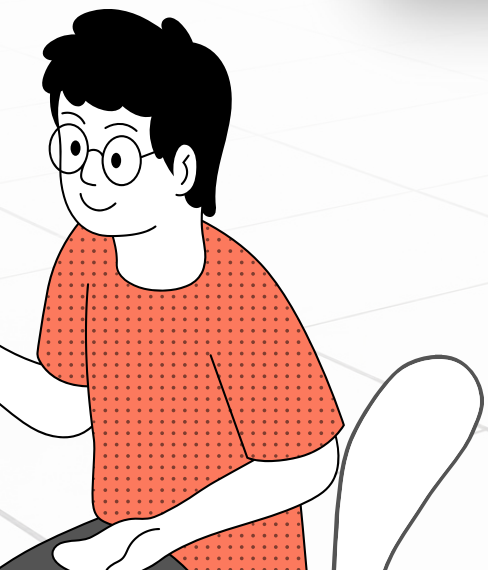
But the array here is declared with const keyword and the value of such variables can't be changed / updated right?

The variable declared with const keyword can't be reassigned. Here we are not assigning a new array to the fruits variable, but we are just updating the existing array whose reference is already stored in fruits, which is perfectly fine.

What will be the output of the next snippet and why?

@frontendhabibi

```
1  (() ⇒ {
2        if (true) {
3             var b = 20;
4             let c  = 25;
5        }
6        console.log(b)
7        console.log(c)
8  })()
```



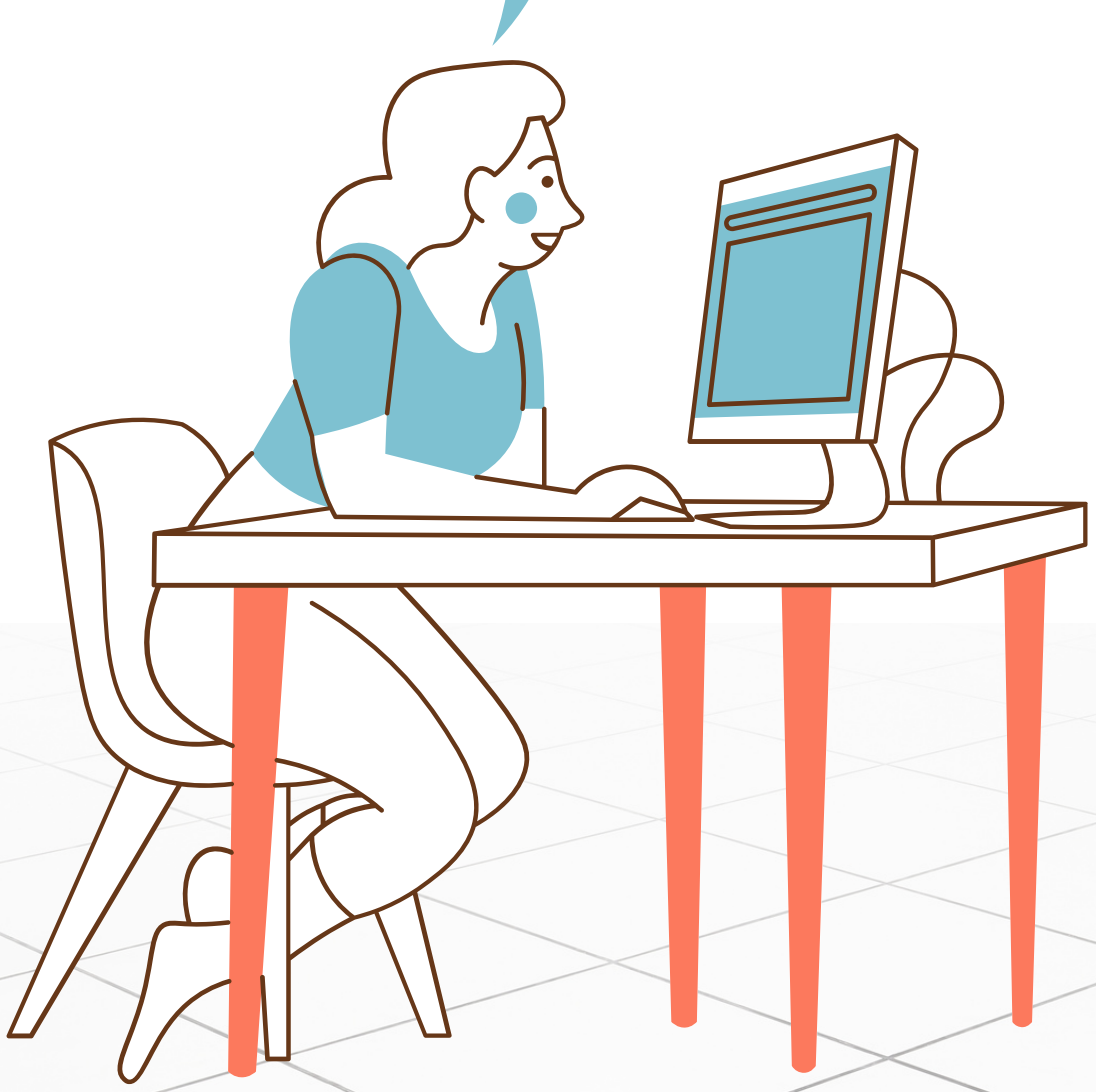
@frontendhabibi

This code snippet will log the value of b as 20 and while executing the line no. 7, it will trhow a ReferenceError: not defined, since we are trying to access the variable c out of its scope.

@frontendhabibi

But the variable b is also declared in the same scope as c, we are able to access b then why can't we access c ?

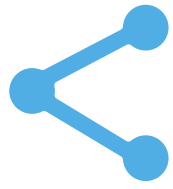
Variables declared with let or const have block scope so they can't be accessed outside of the block of its declaration whereas, variables declared with var have local scope so they can be accessed anywhere in the scope of the current execution context.



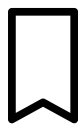
TO BE CONTINUED...

LIKE

COMMENT



SHARE



SAVE FOR FUTURE REFERENCE

@frontendhabibi