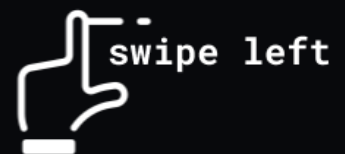


.POP

ARRAY METHODS EXPLAINED - PART 7



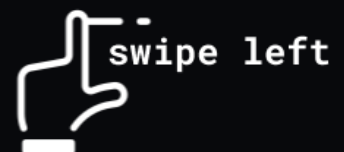
.POP

This is part 7 of my “**Array Methods Explained**” series. I will be explaining the **pop** Array method.

The **pop** method “pops out” and returns the last item of an array. Note that this method modifies the array it is used on.

Here is the Syntax:

```
array.pop()
```



.POP

Instead of using the **index** of the last item in array, you can use this quick method to easily retrieve the last item:

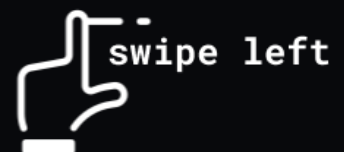
```
const array = [10, 20, 30, 40, 50]

const poppedItem = array.pop()

console.log(poppedItem) // 50

console.log(array) // [10, 20, 30, 40]
```

In the example above, you see how we applied the **pop** method on the **array**. Also, you can see that the **array** has been modified and doesn't have the last item again.



.POP

If you do not want the array to be modified, you can clone it before using the pop method:

```
const array = [10, 20, 30, 40, 50]

const poppedItem = [...array].pop()

console.log(poppedItem) // 50

console.log(array) // [10, 20, 30, 40, 50]
```

By cloning the array with the **spread** operator, the original array is no longer affected.

The opposite of pop is **shift**, which I explain in my next post.