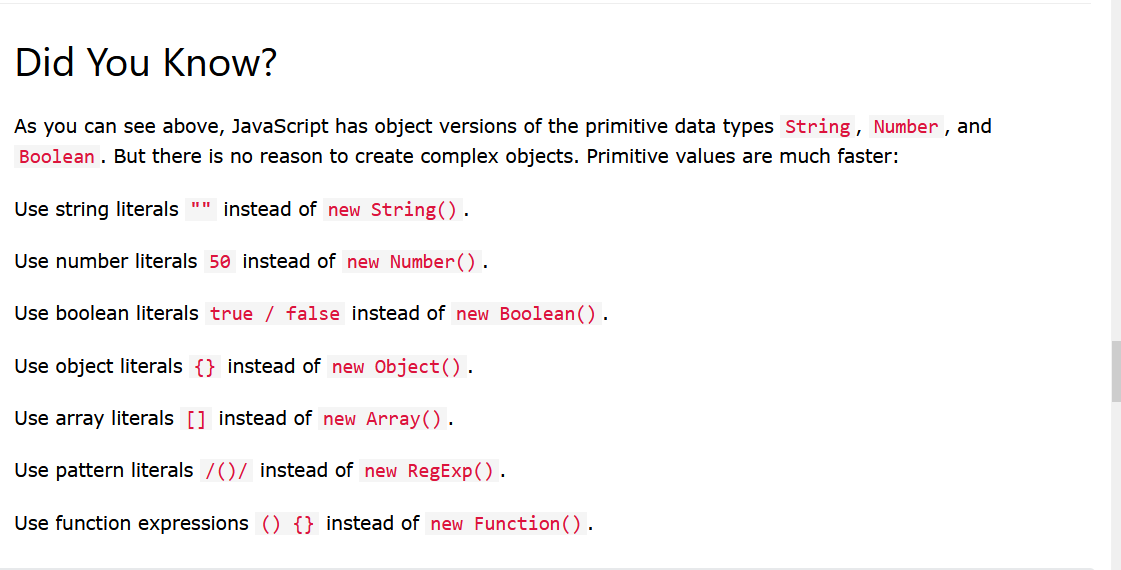
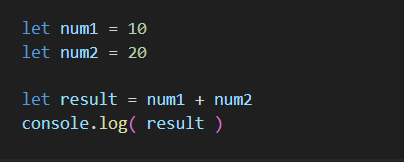
**Literals values are always faster and more efficient**

**So, try using them, instead of Objects**

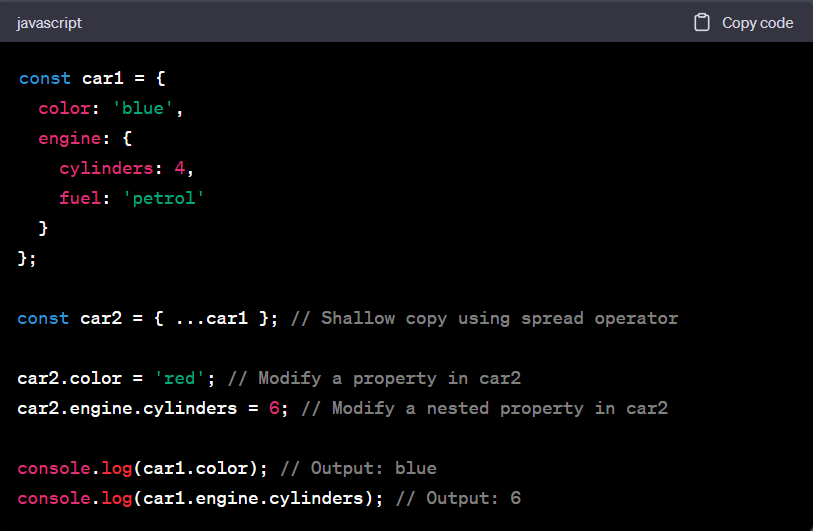
****

Use the spaces between the operators like this



**Shallow copy and Deep Copy**

**Shallow copy**



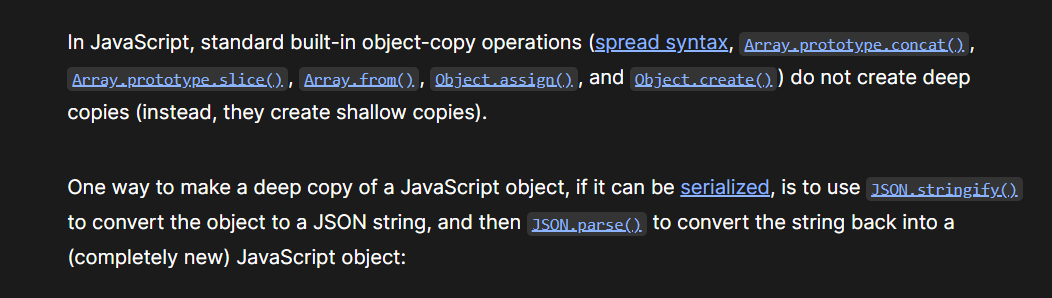
Using the spread operator will cause the shallow copy

Here if there are objects inside the object **their reference will get copied**

So, any changes in the newer objects will result in the changes to the original object

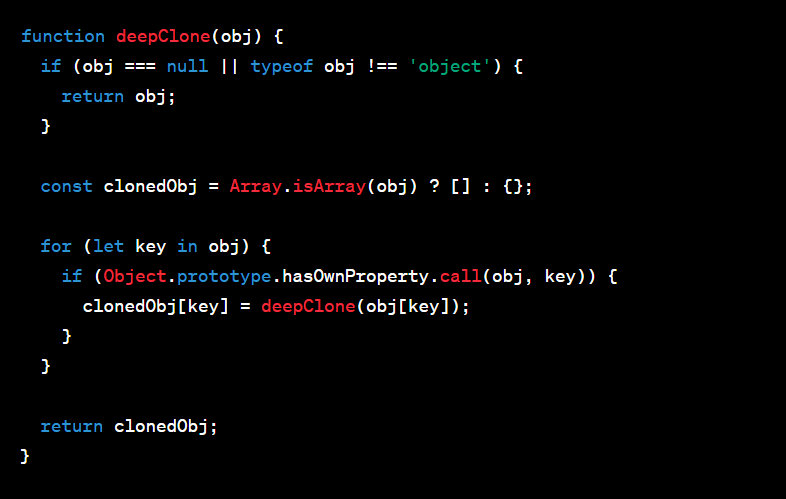
In **Deep copy** the **reference will not be copied** whole new object will be copied

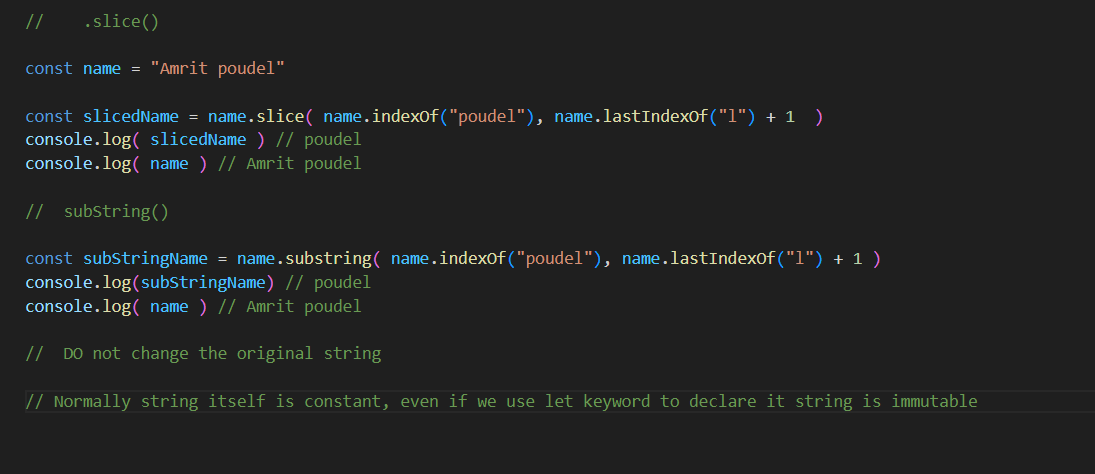
And changes in one do not affect the original object

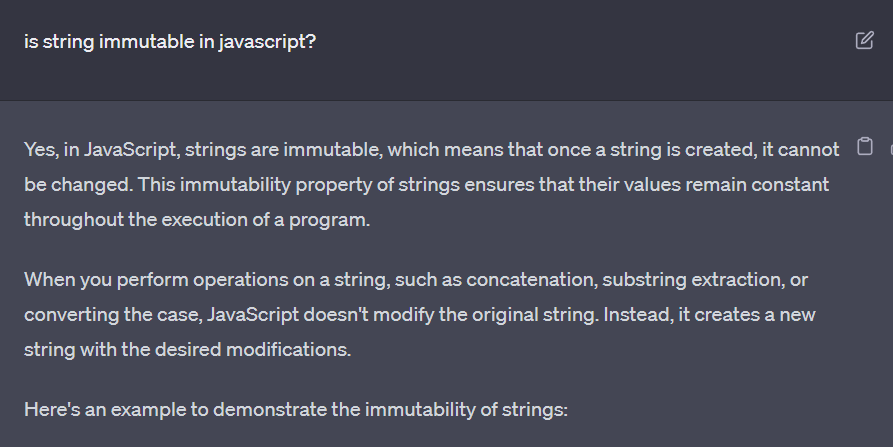


The JSON method of creating the deep copy is slower

So, build custom function of yourself instead

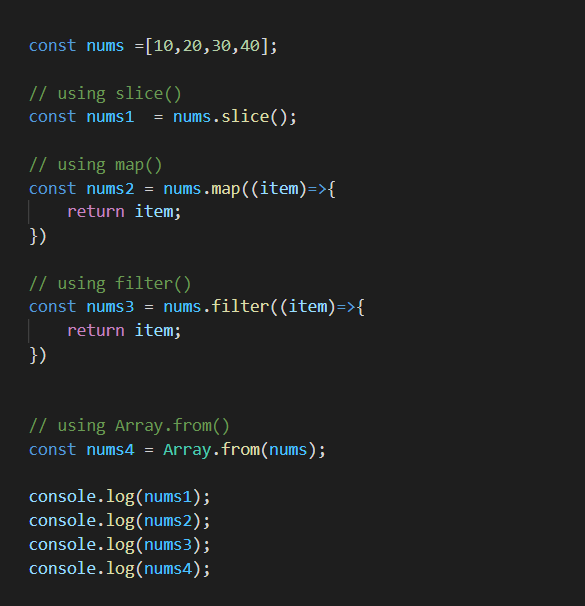






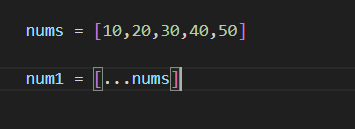
Copying the Array is such a big deal in javascript

There are several ways



OR

Using the … spread operator



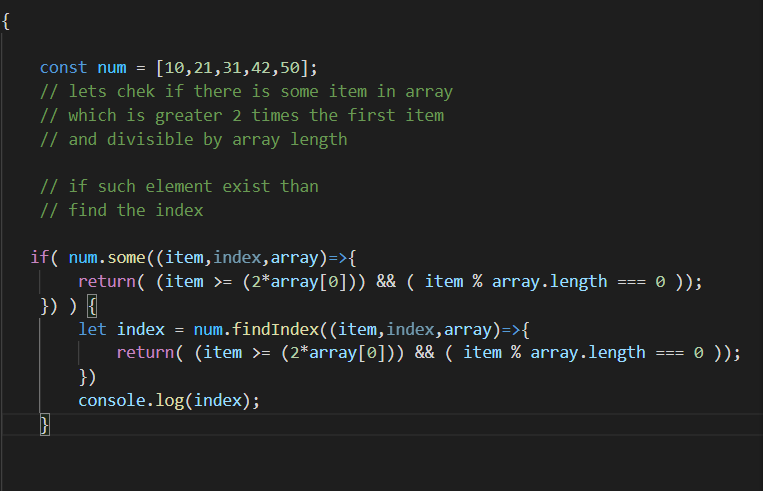
**.findIndex( ) it finds the index of the element that passes the provided test first**

it also takes item,index,array as arguments

Must be wondering why every methods takes these 3 arguments

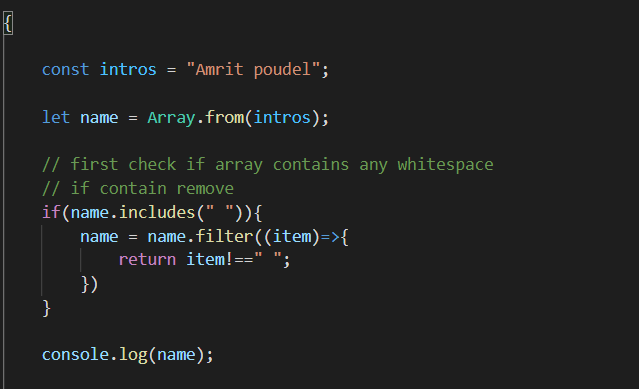
This is very very very handy and super useful at various complex cases

See

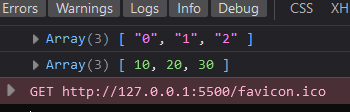


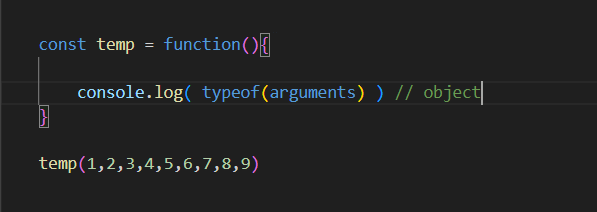
**Array.from() it is used to create an array from the string**

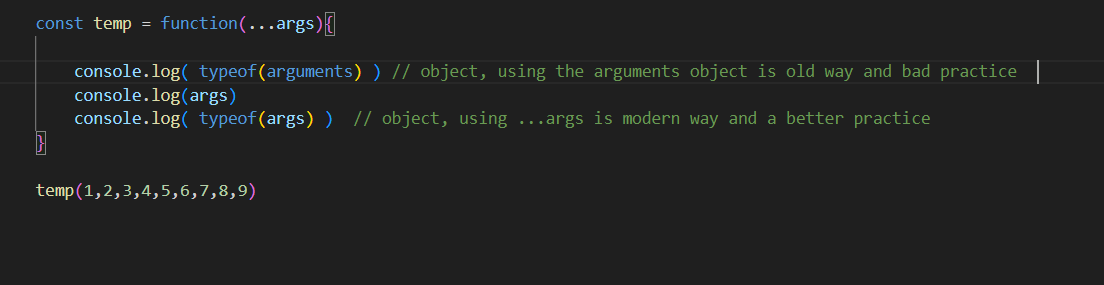
**.includes() used to check if the array contains the provided item or not**

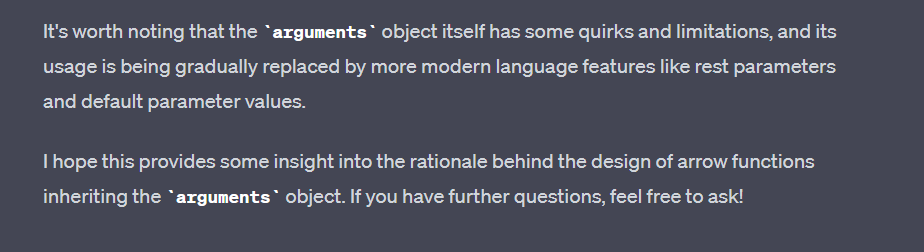


arguments object of function

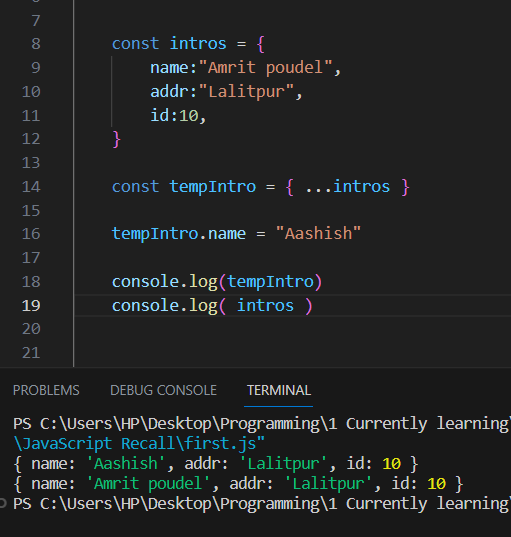








In both Arrays and Objects we can use the spread operator … to copy both the objects and arrays





It is the correct way of making an object “iterable”

