BACKEND

TALLER BASE DE BACKEND FUNCIONES

OSCAR DAVID RUIZ SAEZ

BRAYAN ARCOS BURBANO

INSTITUTO TECNOLOGICO DEL PUTUMAYO

DESARROLLO DE BASE DE DATOS

MOCOA – PUTUMAYO

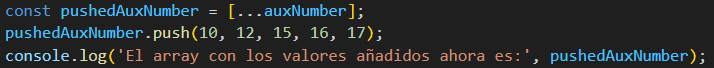
2024

1)código de clases que nos dio el profesor



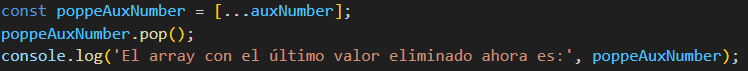
* 1. agregar al array por medion de push 10,12,15,16 y 17

Se crea una nueva constante llamada pushedAuxNumber que es una copia del array auxNumber luego Se añaden los números 10, 12, 15, 16 y 17 al final del array pushedAuxNumber



* 1. eliminar el ultimo elmento con pop

Se crea una constante llamada poppeAuxNumber que es una copia del array auxNumber luego Se elimina el último elemento del array poppeAuxNumber

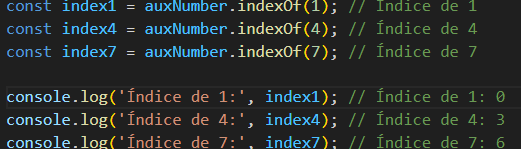


* 1. encontrar el indice de 1,4 y 7 con indexOf

Se obtiene el índice del valor 1 en el array auxNumber y se almacena en la constante index1

Se obtiene el índice del valor 4 en el array auxNumber y se almacena en la constante index4.

Se obtiene el índice del valor 7 en el array auxNumber y se almacena en la constante index7.



* 1. extraer la parte del array con slice de 2 a 9

Se crea una constante llamada slicedAuxNumber que contiene una porción del array auxNumber desde el índice 2 hasta el índice 9



* 1. verificar si todos los elementos son mayores a 1 con every

Se verifica si todos los elementos del array auxNumber son mayores que 1 utilizando el método every. El resultado de esta verificación se almacena en la constante everyOneAuxNumber



* 1. verificar si algun elemento es mayor o igual que 7 con some

Se verifica si al menos uno de los elementos del array auxNumber es mayor o igual a 7 utilizando el método some.

El resultado de esta verificación se almacena en la constante someOneAuxNumber.



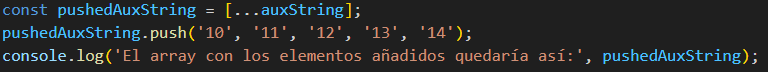
2) código que entrego el profesor



2.1)

Se crea una nueva constante llamada pushedAuxString que es una copia del array auxString.

Se añaden los strings ‘10’, ‘11’, ‘12’, ‘13’ y ‘14’ al final del array pushedAuxString.

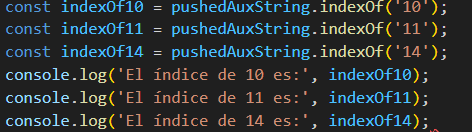


2.2)

Se obtiene el índice del valor '10' en el array pushedAuxString y se almacena en la constante indexOf10

Se obtiene el índice del valor '11' en el array pushedAuxString y se almacena en la constante indexOf11.

Se obtiene el índice del valor '14' en el array pushedAuxString y se almacena en la constante indexOf14.



3)código que entrego el profesor:



3.1)

Se crea una constante llamada wordsArray que contiene un array de strings, resultante de dividir el string message en palabras, utilizando el espacio como separador.



3.2)

Se crea una constante llamada joinedString que contiene un string resultante de unir todos los elementos del array wordsArray, separados por una coma y un espacio.



4)código que envio el profesor:  


4.1)

Se crea una constante llamada namesArray que contiene un array de strings, resultante de dividir el string nameList en nombres individuales, utilizando la coma y el espacio como separador.



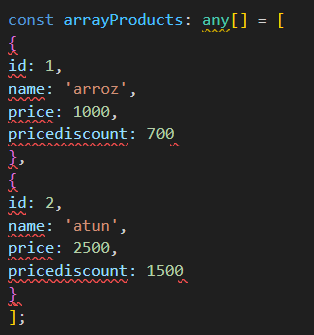
4.2)

Se crea una constante llamada joinedNames que contiene un string resultante de unir todos los elementos del array namesArray, separados por un espacio

Se imprime en la consola el mensaje: “El string en base al array anterior quedaría así:” seguido del valor de joinedNames.



5)código que entrego el profe:



5.1)

Se filtran los elementos del array arrayProducts utilizando el método filter, que evalúa cada elemento para verificar si su propiedad price es mayor a 900.

El resultado de esta filtración se almacena en la constante filterArrayProducts.

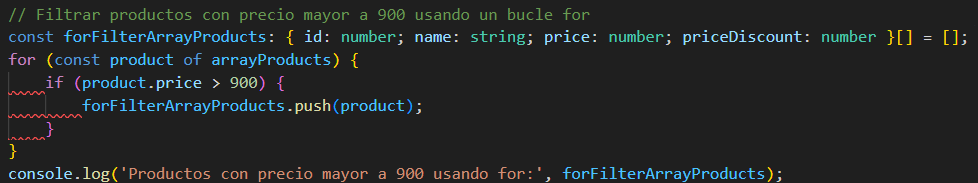


5.1)

Se declara una constante llamada forFilterArrayProducts que es un array vacío de objetos con las propiedades id, name, price y priceDiscount.

Se utiliza un bucle for...of para iterar sobre cada elemento del array arrayProducts.

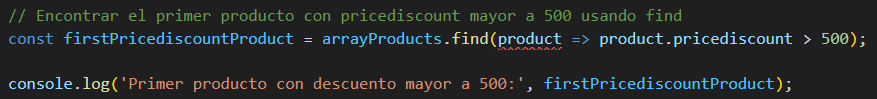
Dentro del bucle, se verifica si la propiedad price del producto es mayor a 900.



5.2)

Se utiliza el método find para buscar el primer elemento en el array arrayProducts que cumpla con la condición de tener la propiedad pricediscount mayor a 500.

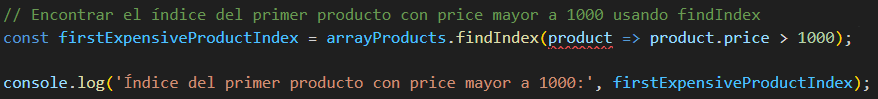
El resultado de esta búsqueda se almacena en la constante firstPricediscountProduct.



5.3)

Se utiliza el método findIndex para buscar el índice del primer elemento en el array arrayProducts que cumpla con la condición de tener la propiedad price mayor a 1000.

El resultado de esta búsqueda se almacena en la constante firstExpensiveProductIndex.



5.4)

Se crea una nueva constante llamada otherProductArrayProducts que es una copia del array arrayProducts.

Se añaden dos nuevos objetos al final del array otherProductArrayProducts. Estos objetos representan productos con las siguientes propiedades:

Primer objeto:

id: 3

name: ‘leche’

price: 3000

pricediscount: 2000

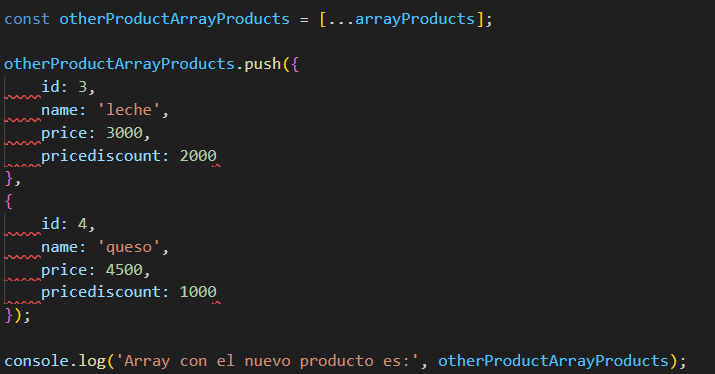
Segundo objeto:

id: 4

name: ‘queso’

price: 4500

pricediscount: 1000



6) código entrego el profe en clases:

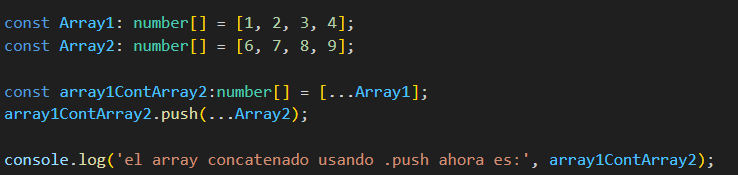
const Array1: number[] = [1, 2, 3, 4];

const Array2: number[] = [6, 7, 8, 9];

6.1)

Se crea una nueva constante llamada array1ContArray2 que es una copia del array Array1.

Se añaden todos los elementos del array Array2 al final del array array1ContArray2 utilizando el método push con el operador de propagación (...).



6.2)

Se utiliza el método sort para ordenar los elementos del array array1ContArray2 de menor a mayor.

La función de comparación (a, b) => a - b se pasa como argumento al método sort para asegurar que los elementos se ordenen numéricamente en orden ascendente.

El resultado de esta ordenación se almacena en la constante sortArray.



6.3)

Se utiliza el método sort para ordenar los elementos del array array1ContArray2 de mayor a menor.

La función de comparación (a, b) => b - a se pasa como argumento al método sort para asegurar que los elementos se ordenen numéricamente en orden descendente.

El resultado de esta ordenación se almacena en la constante sortedArrayDescending.

