

Módulo 10 - Conceptos Avanzados ES6 - Laboratorio

Ejercicios Entregables

hasId

Implementa una función llamada `hasId` que admita como parámetro un objeto y compruebe si dicho objeto tiene una propiedad llamada `id` (debe devolver booleano `true` / `false`).

TIP: No accedas a 'id' con punto (.) o con corchetes ([]).

head

Implementa una función llamada `head` tal que, dado un array como entrada, devuelva el primer elemento.

TIP: No utilices el corchete ([]) para acceder a la posición 0.

tail

Implementa una función llamada `tail` tal que, dado un array como entrada, devuelva un nuevo array con todos los elementos menos el primero.

TIP: No se debe modificar el array de entrada. Piensa en destructuring y rest.

swapFirstToLast

Implementa una función llamada `swapFirstToLast` tal que, dado un array como entrada, devuelva un nuevo array donde el primer elemento ha sido colocado en la última posición.

TIP: No se debe modificar el array de entrada. Piensa en destructuring y rest.

excludeId

Implementa una función llamada `excludeId` tal que, dado un objeto como entrada, devuelva dicho objeto clonado excepto la propiedad `id` si la hubiera.

TIP: No modifiques el objeto de entrada. Piensa en destructuring y rest.

wordsStartingWithA

Implementa una función llamada `wordsStartingWithA` tal que, dado un array de palabras como entrada, devuelva otro array filtrado con aquellas palabras que empiecen por a.

TIP: No utilices bucles.

concat

Implementa una función llamada `concat` tal que admita múltiples argumentos de tipo string y devuelva otro string con la concatenación de todos, separados por `|`.

TIP: No utilices bucles.

multArray

Implementa una función llamada `multArray` que admita un array de números (`arr`) y otro parámetro que sea un número (`x`) y devuelva un nuevo array donde cada elemento ha sido multiplicado por x.

TIP: No utilices bucles.

calcMult

Implementa una función llamada `calcMult` que admita múltiples números como argumento y devuelva como resultado el producto de todos ellos.

TIP: No utilices bucles.

Ejercicios Extra

swapFirstSecond

Implementa una función `swapFirstSecond` tal que dado un array, devuelva un nuevo array donde el primer elemento ha sido intercambiado por el segundo.

TIP: No modifiques el array de entrada original.

longest

Implementa una función `longest` tal que admita multiples strings como argumento de entrada así como un carácter cualquiera, y devuelva un booleano indicando si todos los strings comienzan por dicho carácter o no.

TIP: No utilices bucles. No accedas al primer carácter con corchetes (`[]`).

searchInStringV1

Implementa una función llamada `searchInStringV1` tal que dado un string como parámetro y también un carácter, devuelva cuantas veces aparece dicho carácter en el string.

TIP: No utilices bucles.

TIP: Convierte el string a un array mediante `Array.from()`.

searchInStringV2

Implementa el mismo ejercicio de antes, llamando a la función `searchInStringV2`, encontrando otra alternativa sin usar `reduce`.

TIP: No utilices bucles.

TIP: Convierte el string a un array mediante `Array.from()`.

sortCharacters

Implementa una función llamada `sortCharacters` tal que dado un string, lo devuelva con sus letras ordenadas alfabéticamente.

TIP: No utilices bucles. No modifiques el string original de entrada.

TIP: Convierte el string a un array mediante `Array.from()`.

shout

Implementa una función llamada `shout` tal que, dadas múltiples palabras como entrada, devuelva todas las palabras concatenadas en un texto donde aparezcan en mayúsculas y con exclamaciones.

TIP: No utilices bucles.

Lista de la compra

Dada la siguiente lista de la compra:

```
const shoppingCart = [
  { category: "Frutas y Verduras", product: "Lechuga", price: 0.8, units: 1 },
  { category: "Carne y Pescado", product: "Pechuga pollo", price: 3.75, units: 2 },
  { category: "Droguería", product: "Gel ducha", price: 1.15, units: 1 },
  { category: "Droguería", product: "Papel cocina", price: 0.9, units: 3 },
  { category: "Frutas y Verduras", product: "Sandía", price: 4.65, units: 1 },
  { category: "Frutas y Verduras", product: "Puerro", price: 4.65, units: 2 },
  { category: "Carne y Pescado", product: "Secreto ibérico", price: 5.75, units: 2 },
];
```

A. Obtén una nueva lista donde aparezca el IVA (21%) de cada producto.

B. Ordena la lista de más a menos unidades.

C. Obtén el subtotal gastado en droguería.

D. Obtén la lista por consola en formato "Producto -> Precio Total €" y ordenada por categoría.

TIP: No utilices bucles.