UD4  
EJERCICIOS: Map, Filter y Reduce

short line

# Resumen Gráfico

# 

# 

# ***¡Todos los ejercicios deben de realizarse haciendo uso de las funciones map, filter y reduce!***

# Ejercicios

## Ejercicio 1

Crea una función multiplicarNumeros ( object ) que multiplique todas las variables “numéricas” de un objeto por 2. Por ejemplo para la instancia de objeto siguiente:

|  |
| --- |
| // antes de la llamada let menu = [{  width: 200,  height: 300,  title: "Mi menu" }];  multiplicarNumeros(menu);  // Después de la llamada // menu = { width: 400, height: 600, title: "Mi menu" }; |

* La función no debe de devolver ningún tipo de valor.
* Debes de mirar si los valores son numéricos o no antes de multiplicarle su valor por 2.

<script>

// antes de la llamada

let n ;

let ob;

let menu = {

width: 200,

height: 300,

title: "Mi menu"

};

function multiplicarNumeros(menu){

n = Object.values(menu);

ob = n.map( function (n){

if(typeof n === 'number'){

return n\*2;

}else{

return n;

}

});

menu.width = ob[0];

menu.height = ob[1];

}

multiplicarNumeros(menu);

console.log(menu);

//multiplicarNumeros(menu);

// Después de la llamada

// menu = { width: 400, height: 600, title: "Mi menu" };

## Ejercicio 2

Tenemos un objeto que guarda los salarios de un equipo de trabajo:

|  |
| --- |
| **let salarios = {  Juan: 10000,  Alicia: 16000,  Lucas: 13000 }** |

Escribe un script para sumar estos salarios y mostrar el resultado de la suma.

<script>

let salarios = {

Juan: 10000,

Alicia: 16000,

Lucas: 13000

}

let total = 0;

function sumaSalarios(salarios){

salarios = Object.values(salarios);

salarios.map(function (salarios){

return total = total + salarios;

})

}

sumaSalarios(salarios);

console.log("Total: " + total);

</script>

## 

## Ejercicio 3

Dado el siguiente array de objetos:

|  |
| --- |
| const iceCreams = [  { flavor: 'pineapple', color: 'white' },  { flavor: 'strawberry', color: 'red' },  { flavor: 'watermelon', color: 'red' },  { flavor: 'kiwi', color: 'green' },  { flavor: 'mango', color: 'yellow' },  { flavor: 'pear', color: 'green' } ]; |

3.1 . ¿Cómo podría obtener un array con solo los helados que son de color rojo ?

//Helados de color rojo

rojos = iceCreams.filter(funcioPrueba => funcioPrueba.color === 'red').map(hela => "Flavor: " + hela.flavor + ", Color: " + hela.color);

console.log("Array de helados de color rojo: ", rojos);

3.2. ¿ Cómo podríamos obtener un array que solo contenga los sabores de los helados tal como [ “pineaple” , “watermelon”, “kiwi”, ……..etc ] ?

## //Sabores

## var sabores = [];

## 

## iceCreams.map(function (saboreh){

## sabores.push(saboreh.flavor);

## return saboreh.flavor;

## })

## 

## console.log("Sabores: " + sabores);

## 

## Ejercicio 4

Dado la siguiente estructura de datos sobre pilotos y años de experiencia:

|  |  |  |
| --- | --- | --- |
| |  | | --- | | let pilotos = [  {  id: 10,  name: "Poe Dameron",  years: 14,  },  {  id: 2,  name: "Temmin 'Snap' Wexley",  years: 30,  }, | | {  id: 41,  name: "Tallissan Lintra",  years: 16,  },  {  id: 99,  name: "Ello Asty",  years: 22,  } ]; |

4.1. ¿Cómo podríamos obtener los años totales de experiencia de todos los pilotos juntos?

//Años totales de experiencia de todos los piotos juntos

let total = 0;

pilotos.map(function (anikos){

return total = total + anikos.years;

})

console.log("Años totales de experiencia: ", total);

4.2. Realice la misma operación del apartado anterior (4.1) pero haciendo uso del operador “arrow”:

//Años totales de experiencia de todos los piotos juntos con operador arrow =>

anios = pilotos.map(anikos=>anikos.years).reduce((acum, actual)=> acum + actual, 0);

console.log("Años de experiencia: ", anios);

4.3 Ahora vamos a tratar de obtener el piloto con más experiencia. (Esta hay que darle un par de vueltas).

//Piloto con mas experiencia

let max = 0;

let piloto;

pilotos.map(function (anikos){

if(anikos.years==0){

max = anikos.years;

}else if(anikos.years>max){

max = anikos.years;

piloto = anikos.name;

}

})

console.log("El piloto con más años de experiencia es ", piloto, " con ", max, " años de experiencia");

## Ejercicio 5

Dado la siguiente estructura de datos sobre pilotos con id, nombre y su división (faction):

|  |  |
| --- | --- |
| var pilots = [  {  id: 2,  name: "Wedge Antilles",  faction: "Rebels",  },  {  id: 8,  name: "Ciena Ree",  faction: "Empire",  }, | {  id: 40,  name: "Iden Versio",  faction: "Empire",  },  {  id: 66,  name: "Thane Kyrell",  faction: "Rebels",  } ]; |

5.1. ¿Cómo podría obtener los pilotos que fueran solo Rebeldes (Rebels)?

//Pilotos rebeldes

var pilotes = "Pilotos rebeldes: ";

pilots.filter(function(pilo){

if(pilo.faction === "Rebels"){

pilotes = pilotes + pilo.name + ", ";

};

})

console.log(pilotes);

5.2. Realice la misma operación del apartado 5.1 pero haciendo uso del operador “arrow”.

//Pilotos rebeldes con arrow =>

pilotes = "Pilotos Rebeldes: ";

gg = pilots.filter(pilo=>pilo.faction === "Rebels").map(piloh => piloh.name);

console.log(pilotes + gg);

## Ejercicio 6

Dado la siguiente estructura de datos sobre personajes

|  |  |
| --- | --- |
| var personnel = [  {  id: 5,  name: "Luke Skywalker",  pilotingScore: 98,  shootingScore: 56,  isForceUser: true,  },  {  id: 82,  name: "Sabine Wren",  pilotingScore: 73,  shootingScore: 99,  isForceUser: false,  },  {  id: 22,  name: "Zeb Orellios",  pilotingScore: 20,  shootingScore: 59,  isForceUser: false,  }, | {  id: 15,  name: "Ezra Bridger",  pilotingScore: 43,  shootingScore: 67,  isForceUser: true,  },  {  id: 11,  name: "Caleb Dume",  pilotingScore: 71,  shootingScore: 85,  isForceUser: true,  }, ]; |

6.1. Intente obtener el total de todas las puntuaciones (tanto pilotingScore como shootingScore juntas) de todos aquellos personajes que sean solo usuarios de la fuerza (isForceUser == true). (CONSEJO: Aquí va a necesitar encadenar diversos operadores).

//Puntuación total

var contador = 0;

personnel.filter(function (per){

if(per.isForceUser===true){

contador = contador + per.pilotingScore + per.shootingScore;

}

})

console.log("Puntuación total: " + contador);

6.2. Realice lo anterior pero haciendo uso esta vez del operador arrow (=>).

//Puntuación total con operador arrow =>

pp = personnel.filter(per=>per.isForceUser===true).map(pers=>pers.shootingScore + pers.pilotingScore).reduce((acum, actual)=>acum+actual, 0);

console.log("Puntuación total: " + pp);