[OSAKamp] ASOCIACIÓN DE EMPRESAS DE TECNOLOGÍAS LIBRES Y CONOCIMIENTO ABIERTO DE EUSKADI

02 – Funciones y Arrow Functions

Teoria

CODE4JOBS

Bizkaia
toru aldundia
diputación foral

[esle]

Funciones de Arrays

Los arrays tienen una serie de funciones que podemos usar dependiendo de lo que necesitemos hacer con esos arrays. Veremos su forma “normal” y su forma de “arrow function”. Hasta ahora, para declarar una función usábamos la palabra reservada function, le dábamos un nombre y generábamos el código dentro de las llaves. Las arrow function son una forma acortada de usar determinadas funciones. Tienen otras particularidades, pero las omitiremos por el momento.

forEach

El método forEach recibe una función anónima (sin nombre) que tiene como parámetro un elemento. El forEach pasa a ese parámetro cada elemento del array uno tras otro. Es como un for, va recorriendo el array. La diferencia es que para acceder a cada posición del array utilizábamos la i, y aquí lo que sucede es que, en cada vuelta, el parámetro va a contener los datos de la posición del array en la que estemos.

```
let numeros = [1,2,3,4,5,6,7,8,9,10]

numeros.forEach(function(numero){
    console.log(numero)
})
```

Map

Al igual que el forEach, map pasa a ese parámetro cada elemento del array uno tras otro. La diferencia con forEach es que la función map genera un nuevo array con los nuevos elementos que devuelve la función pasada a map con cada elemento del array original.

```
let numeros = [1,2,3,4,5,6,7,8,9,10]

console.log(numeros);

let numerosPorDos = numeros.map(function(numero){
    return numero*2;
})

console.log(numerosPorDos);
```

filter

Esta función va a filtrar los elementos del array, y a crear un nuevo array devolviendo aquellos que cumplan con la condición que le indicamos.

```
let numeros = [1,2,3,4,5,6,7,8,9,10]

console.log(numeros);

let numerosPares = numeros.filter(function(numero){
    return numero % 2 === 0;
})
console.log(numerosPares);
```

reduce

Este método recibe dos parámetros en la función, un total y cada uno de los elementos. Si no se especifica un total, el primer elemento del array es el total y se comienza por el segundo elemento. Por cada elemento, total tiene el valor del return anterior y actual tiene el valor del elemento actual. Devuelve un único valor resultado de ejecutar la función en cada uno de los elementos

```
let numeros = [1,2,3,4,5,6,7,8,9,10]

console.log(numeros);

let sumaArray = numeros.reduce(function(total, actual) {
    return total += actual;
    });

console.log(sumaArray);
```

Ahora veremos estas mismas funciones en su forma arrow function. Ponemos los parámetros que pasamos a la función entre paréntesis. En caso de que solo sea un parámetro, podemos no poner las paréntesis. Después los símbolos igual mayor (sin espacio entre ellos) y finalmente las llaves envuelven lo que haremos dentro de la función. Si lo que hacemos dentro de la función es una sola instrucción con return, podemos también quitar las llaves, el return y el punto y coma.

```
let numeros = [1,2,3,4,5,6,7,8,9,10]

numeros.forEach(numero => console.log(numero))

let numerosPorDos = numeros.map(numero => numero*2)
let numeroPorDosConReturn = numeros.map(numero => {return numero*2})

let filter = numeros.filter(numero => numero % 2 === 0);
let filterConReturn = numeros.filter(numero => {return numero % 2 === 0})

let sumaArray = numeros.reduce((total, actual) => total += actual)
let sumaArrayConReturn = numeros.reduce((total, actual) => {return total
+= actual})
```

Aquí están todas las funciones de array que hemos visto antes en su forma arrow function.