



ARREGLOS MULTIDIMENSIONALES 2

Recorrido de arreglos multidimensionales

Usando ciclo for

Para recorrer un arreglo multidimensional, por ejemplo uno bidimensional con ciclo for, se usan dos ciclos anidados y dos índices, i y j, de tal manera que el ciclo for externo recorra los elementos más externos, los cuales son arreglos, y el ciclo for interno recorra los elementos más internos, los cuales no son arreglos.

Ejemplo: Dado el siguiente arreglo `[[20, 80, 30], [10, 20, 50],[70, 60, 40]]` recorrerlo usando ciclo for. Como el arreglo dado es un arreglo bidimensional, entonces usamos dos ciclos for y dos índices.

```
let arregloBidimensional = [[20, 80, 30], [10, 20, 50],[70, 60, 40]];
//El siguiente ciclo recorre los elementos que conforman el arreglo
//arregloBidimensional es decir, [20, 80, 30], [10, 20, 50],[70, 60, 40]
for (let i = 0; i < arregloBidimensional.length; i++) {
  //El siguiente ciclo recorre los elementos de cada elemento del arreglo
  //arregloBidimensional, es decir 20, 80, 30, 10, 20, 50 , 70, 60, 40
  for (let j = 0; j < arregloBidimensional[i].length; j++) {
    //Acá se imprime cada elemento más interno del arreglo
    console.log(arregloBidimensional[i][j]);
  }
}
```

La salida del anterior programa en la consola del navegador es:

20
80
30
10
20
50
70
60
40
>



El código anterior es una fórmula que siempre va a funcionar para recorrer un arreglo bidimensional con ciclo for. Se puede usar simplemente reemplazando en él el nombre del arreglo bidimensional a recorrer.

Ejemplo: Recorrer el arreglo [["w", true, 4], [3, 99], [4, 6, 7, 8]]

Reemplazamos el nombre del arreglo a recorrer en el anterior código, es decir, cambiamos arregloBidimensional por arregloMixto, así:

```
let arregloMixto = [ ["w", true, 4], [3, 99], [4, 6, 7, 8] ];  
  
for (let i = 0; i < arregloMixto.length; i++) {  
  for (let j = 0; j < arregloMixto[i].length; j++) {  
    console.log(arregloMixto[i][j]);  
  }  
}
```

La salida del anterior programa en la consola del navegador es:

w
true
4
3
99
4
6
7
8
>

Usando ciclo forEach

Para recorrer un arreglo multidimensional, por ejemplo uno bidimensional con ciclo forEach, se usan dos ciclos anidados, de tal manera que el ciclo forEach externo recorra los elementos más externos, los cuales son arreglos, y el ciclo forEach interno recorra los elementos más internos, los cuales no son arreglos.



Ejemplo: Dado el siguiente arreglo `[[20, 80, 30], [10, 20, 50],[70, 60, 40]]` recorrerlo usando ciclo `forEach`.

Como el arreglo dado es un arreglo bidimensional, entonces usamos dos ciclos `forEach`

```
let arregloBidimensional = [[20, 80, 30], [10, 20, 50],[70, 60, 40]];
//El siguiente ciclo recorre los elementos que conforman el arreglo
//arregloBidimensional es decir, [20, 80, 30], [10, 20, 50],[70, 60, 40]
arregloBidimensional.forEach(i => {
  //El siguiente ciclo recorre los elementos de cada elemento del arreglo
  //arregloBidimensional, es decir 20, 80, 30, 10, 20, 50 , 70, 60, 40
  i.forEach(j => {
    //Acá se imprime cada elemento más interno del arreglo
    console.log(j);
  })
})
```

La salida del anterior programa en la consola del navegador es:

20
80
30
10
20
50
70
60
40
>

El código anterior es una fórmula que siempre va a funcionar para recorrer un arreglo bidimensional con ciclo `forEach`. Se puede usar simplemente reemplazando en él el nombre del arreglo bidimensional a recorrer.

En general, para recorrer un arreglo multidimensional, primero nos fijamos en la dimensión del arreglo, si el arreglo es de dimensión 2 y lo queremos recorrer con un ciclo `for`, entonces se usan DOS ciclos `for` anidados y DOS índices, si lo queremos recorrer con `forEach`, entonces se usan DOS ciclos `forEach` anidados. Si el arreglo que queremos recorrer es de dimensión 3 y lo queremos recorrer con un ciclo `for`, entonces se usan TRES ciclos `for` anidados y TRES índices, si lo queremos recorrer con `forEach`, entonces se usan TRES ciclos `forEach` anidados...



Taller

Usando lo anteriormente expuesto, manejando el número de índices correctos en cada caso, resuelva:

1. Dado el arreglo [["a", "z", "t"], [789, 887, 451], [true, false, 0]]:

- a) Recorrer el arreglo imprimiendo todos sus elementos más internos usando ciclo for
- b) Recorrer el arreglo imprimiendo todos sus elementos más internos usando ciclo forEach

2. Dado el arreglo [["%", 7, true, "#"], ["&", 55, false, "="], [[false, 0, 99, "?"], [true, 1000, "@", "i"]], [[44, 55, 66, 77], [1, 2, 3, 4]]]

- a) Recorrer el arreglo imprimiendo todos sus elementos más internos usando ciclo for
- b) Recorrer el arreglo imprimiendo todos sus elementos más internos usando ciclo forEach

3. Dado el arreglo [[2, 8, 9], [30, 71, 77], [11, 990, 41]]:

- a) Recorrer todos sus elementos más internos y mostrar los números que sean impares. Use cualquier ciclo, for o forEach.
- b) Recorrer todos sus elementos más internos y mostrar la suma de estos. Como sugerencia use una variable para acumular la suma, declarela antes del ciclo. Use cualquier ciclo, for o forEach.