wawiwa

Iterando en un arreglo con array forEach

El método forEach() ejecuta una función dada una vez por cada elemento del arreglo

```
const arr = ['a', 'b', 'c'];

arr.forEach(element => console.log(element));

// expected output: "a"

// expected output: "b"

// expected output: "c"
```

wawiwa

Iterando en un arreglo con array forEach

```
const items = ['item1', 'item2', 'item3'];
const copyItems = [];

for (let i = 0; i < items.length; i++) {
    copyItems.push(items[i]);
}
```

```
const items = ['item1', 'item2', 'item3'];

items.forEach(function(item) {
    copyItems.push(item);
});
```

wawiwa

Iterando en un arreglo con array forEach

```
// Por atención que el índice 2 fue saltado, ya que no har ningún
elemento
// en esa posición del arreglo...
[2, 5, , 9].forEach(function(element, index, array) {
    console.log('a[' + index + '] = ' + element);
});

// logs:
// a[0] = 2
// a[1] = 5
// a[3] = 9
```

callbackFn es invocada con tres argumentos:

1. El valor del elemento
2. El índice del elemento
3. El arreglo que se está recorriendo

wawiwa

Iterando en un arreglo con array forEach

```
const words = ['one', 'two', 'three', 'four'];

words.forEach(function(word) {
    console.log(word);

    if (word === 'two') {
        words.shift(); // 'one' va a ser borrado del arreglo
    }
}); // one // two // four

console.log(words); // ['two', 'three', 'four']
```

wawiwa

Iterando map con forEach

El método **Map.forEach** es utilizado para iterar sobre el mapa en la función dada y ejecuta la función con cada par key-value.

Sintáxis:

```
myMap.forEach(callback, value, key, thisArg)
```

Parámetros: Éste método acepta cuatro parámetros como se había mencionado arriba y descrito abajo:

- **callback:** Esta es la función que se ejecuta en cada llamada de la función.
- **value:** Este es el valor para cada iteración.
- **key:** Esta es la clave para llegar a la iteración.
- **thisArg:** Este es el valor para usar como "this" cuando se ejecuta el callback.

```
// Creando un mapa usando el objeto Map
const mp = new Map();

// Agregando valores al mapa
mp.set("a", 1);
mp.set("b", 2);
mp.set("c", 3);

// Imprimiendo el objeto en la consola
mp.forEach((value, key) => {
    console.log(value, key);
});
```

Resultado:

```
1a
2b
3c
```

wawiwa

Iterando map con forEach

El método forEach ejecuta el callback dado una vez por cada clave en el mapa que exista. No es invocado pro claves que han sido borradas. Sin embargo, es ejecutada por valores que están presentes pero su valor es undefined.

```
function logMapElements(value, key) {
    console.log(`m[${key}] = ${value}`);
}

new Map([
    ['foo', 3],
    ['bar', {}],
    ['baz', undefined]
]).forEach(logMapElements);

// expected output: "m[foo] = 3"
// expected output: "m[bar] = [object Object]"
// expected output: "m[baz] = undefined"
```

callback es invocado con los siguientes argumentos:

- El valor de entrada
- La clave de entrada
- El mapa que se está recorriendo

wawiwa

Iterando en un set con forEach

El método forEach() ejecuta una función dada una vez por cada valor en el objeto Set, en orden de inserción.
El método forEach() ejecuta el callback dado una vez por cada valor que existe en el objeto Set. No es invocada por valores que han sido borrados. Sin embargo, es ejecutado por valores que están presentes pero su valor es undefined.
callback es invocado con tres argumentos:

- El valor del elemento
- La clave del elemento
- El objeto Set que está siendo recorrido

No hay claves en los objetos Set, sin embargo, los primeros dos valores son valores contenidos en el Set. Esto es para hacerlo consistente con otros métodos forEach() para Map y Array.

wawiwa

Iterando en un set con forEach

```
function logSetElements(val1, val2) {
    console.log(`s[${val1}] = ${val2}`);
}

new Set(['foo', 'bar', undefined]).forEach(logSetElements);

// resultado esperado: "s[foo] = foo"
// resultado esperado: "s[bar] = bar"
// resultado esperado: "s[undefined] = undefined"
```

callback es invocado con tres argumentos:

- El valor del elemento
- La clave del elemento
- El objeto Set que está siendo recorrido

wawiwa

Práctica: Iterando en un arreglo con array forEach práctica

Escribe un código que imprima el resultado que ves abajo:

Instrucciones:

1. Crea un arreglo específico
2. Usa forEach()
3. Bonus: imprima el arreglo para ver si el arreglo ha cambiado

```
1 x 1 = 1
2 x 2 = 4
3 x 3 = 9
4 x 4 = 16
5 x 5 = 25
```

wawiwa

Solución Iterando en un arreglo con array forEach

JS:

```
// un arreglo de números
const numberArray = [1, 2, 3, 4, 5];

// muestra el cuadrado de cada número
const returnValue = numberArray.forEach(num =>
   console.log(`${num} x ${num} = ${num * num}`);
);

// el arreglo no cambió
console.log(numberArray);
```

Resultado:

```
1 x 1 = 1
2 x 2 = 4
3 x 3 = 9
4 x 4 = 16
5 x 5 = 25
[1,2,3,4,5]
```

wawiwa

¿Preguntas?