

Técnicas de Programación

Carrera Programador full-stack

Metodos

.foreach

El método `forEach` de JavaScript es una de las varias formas de recorrer un arreglo.

Vamos a comenzar con el siguiente arreglo

```
const numeros = [1, 2, 3, 4, 5];
```

.foreach

Si utilizáramos el tradicional "bucle for" para recorrer el arreglo sería así:

```
for (i = 0; i < numeros.length; i++) {  
    console.log(numeros[i]);  
}
```

En cambio el método `forEach` también se utiliza para recorrer un arreglo, pero utiliza una función diferente a la del clásico "bucle for".

.foreach

El método `forEach` pasa una función callback para cada elemento del arreglo junto con los siguientes parámetros:

1. Valor actual (requerido) - El valor del elemento actual del arreglo
2. Index (opcional) - El número de índice del elemento actual
3. Arreglo (opcional) - El objeto del arreglo al que pertenece el elemento actual

.foreach

En primer lugar, para recorrer un arreglo utilizando el método `forEach`, se necesita una función callback (o función anónima), que se ejecutará para cada elemento del arreglo. Debe tomar al menos un parámetro que represente los elementos del arreglo:

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

.foreach

Como alternativa, puede utilizar la representación de la función de flecha de ES6 para simplificar el código:

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

.foreach

Parámetros opcionales

Index: Representa el número de índice de cada elemento.

```
numeros.forEach((numero, index) => {  
  console.log('Indice: ' + index + ' Valor: ' + numero);  
});
```

Arreglo: El parámetro del arreglo es el propio arreglo.

```
numeros.forEach((numero, index, arreglo) => {  
  console.log(arreglo);  
});
```

```
▶ (5) [1, 2, 3, 4, 5]  
▶ (5) [1, 2, 3, 4, 5]  
▶ (5) [1, 2, 3, 4, 5]  
▶ (5) [1, 2, 3, 4, 5]  
▶ (5) [1, 2, 3, 4, 5]
```