.flat()

El método **flat()** crea una nueva matriz con todos los elementos de sub-array concatenados recursivamente hasta la profundidad especificada.

sintaxis

*var newArray = arr*.flat(*[depth]*);

ejemplo

var arr1 = [1, 2, [3, 4]];

arr1.flat();

// [1, 2, 3, 4]

var arr2 = [1, 2, [3, 4, [5, 6]]];

arr2.flat();

// [1, 2, 3, 4, [5, 6]]

var arr3 = [1, 2, [3, 4, [5, 6]]];

arr3.flat(2);

// [1, 2, 3, 4, 5, 6]

.flatmap()

El método **flatMap()** primero mapea cada elemento usando una función de mapeo, luego aplana el resultado en una nueva matriz. Es idéntico a un [map](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array/map) seguido de un [flatten (en-US)](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/flat" \o "Currently only available in English (US))de profundidad 1, pero flatMap es a menudo útil y la fusión de ambos en un método es ligeramente más eficiente.

Sintaxis

var new\_array = arr.flatMap(function callback(currentValue[, index[, array]]) {

// return element for new\_array

}[, thisArg])

Parametros

callback

Función que produce un elemento de la nueva matriz, tomando tres argumentos:

currentValue

El elemento actual que se procesa en la matriz.

indexOptional

El índice del elemento actual que se procesa en la matriz.

arrayOptional

La matriz map fue llamada.

thisArgOptional

Valor para usar como this al ejecutar callback.

Ejemplo

var arr1 = [1, 2, 3, 4];

arr1.map(x => [x \* 2]);

// [[2], [4], [6], [8]]

arr1.flatMap(x => [x \* 2]);

// [2, 4, 6, 8]

// solo un nivel es aplanado

arr1.flatMap(x => [[x \* 2]]);

// [[2], [4], [6], [8]]

.forEach()

El método **forEach()** ejecuta la función indicada una vez por cada elemento del array.

## [Sintaxis](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach#sintaxis)

arr.forEach(function callback(currentValue, index, array) {

// tu iterador

}[, thisArg]);

### [Parámetros](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach#parameters)

**callback**

Función a ejecutar por cada elemento, que recibe tres argumentos:

**currentValue**

El elemento actual siendo procesado en el array.

**index Optional**

El índice del elemento actual siendo procesado en el array.

**array Optional**

El vector en el que forEach() esta siendo aplicado.

**thisArg Optional**

Valor que se usará como this cuando se ejecute el callback.

Ejemplo

function logArrayElements(element, index, array) {

console.log("a[" + index + "] = " + element);

}

// Nótese que se evita el 2° índice ya que no hay ningún elemento en esa posición del array

[2, 5, , 9].forEach(logArrayElements);

// salida:

// a[0] = 2

// a[1] = 5

// a[2] = 9