

[Công nghệ ▾](#)[Tài liệu tham khảo & hướng dẫn ▾](#)[Phản hồi ▾](#)[Đăng nhập !\[\]\(d66ff64371a51729ac8c1cdaa685ba6f\_img.jpg\)](#)

# Array.prototype.filter()

Phương thức **filter()** dùng để tạo một mảng mới với tất cả các phần tử thỏa điều kiện của một hàm test.

Khảo sát MDN



Hãy giúp chúng tôi hiểu 10 nhu cầu hàng đầu của các nhà phát triển Web và nhà thiết kế.

Tham gia khảo sát



### JavaScript Demo: Array.filter()

```
1 var words = ['spray', 'limit', 'elite', 'exuberant', 'destruction', 'present'];
2
3 const result = words.filter(word => word.length > 6);
4
5 console.log(result);
6 // expected output: Array ["exuberant", "destruction", "present"]
7
```

Run ›

Reset

> Array ["exuberant", "destruction", "present"]

Khảo sát MDN



Hãy giúp chúng tôi hiểu 10 nhu cầu hàng đầu của các nhà phát triển Web và nhà thiết kế.

Tham gia khảo sát

Tham số 



## callback

Đây là hàm thử, dùng để kiểm tra từng phần tử của mảng. Trả về `true` để giữ lại phần tử, hoặc `false` để loại nó ra. Nó được gọi với ba tham số:

### element

Phần tử đang được xử lý trong mảng.

### index

Optional

Chỉ mục (index) của phần tử đang được xử lý.

### array

Optional

Mảng nguồn mà hàm `filter` đang xử lý.

### thisArg

Optional

Không bắt buộc. Giá trị của `this` bên trong hàm `callback`.

Giá trị trả về 


Một mảng mới với các phần tử đã thỏa điều kiện của hàm test. Nếu không có phần tử nào thỏa điều kiện, một mảng rỗng sẽ được trả về.

Khảo sát MDN



Hãy giúp chúng tôi hiểu 10 nhu cầu hàng đầu của các nhà phát triển Web và nhà thiết kế.

Tham gia khảo sát

`filter()` sẽ thực thi hàm `callback` trên từng phần tử của mảng  và dựng một mảng mới với các phần tử mà giá trị trả về của `callback` nếu ép kiểu sẽ mang giá trị `true`.

`callback` chỉ được thực thi tại những chỉ mục (index) của mảng mà chúng được gán giá trị; nó không được thực thi tại chỉ mục đã bị xóa hoặc chưa từng được gán giá trị. Những phần tử không thỏa điều kiện tại hàm thử `callback` sẽ bị bỏ qua, không được cho vào mảng mới.

`callback` được gọi với ba tham số:

1. giá trị của phần tử
2. chỉ mục (index) của phần tử
3. mảng ban đầu mà hàm thử đang được gọi lên

Nếu tham số `thisArg` được truyền cho hàm `filter`, nó sẽ được thay vào giá trị của từ khóa `this` trong hàm `callback`. Nếu không, giá trị `undefined` sẽ được dùng cho `this`. Tóm lại, giá trị của từ khóa `this` trong hàm `callback` được xác định tuân theo các quy tắc thông thường để xác định `this` trong một hàm.

`filter()` không làm thay đổi mảng mà nó được gọi.

Các phần tử được `filter()` chạy qua được xác định từ đầu trước khi `callback` được gọi lần đầu tiên. Những phần tử mới được thêm vào sau khi `filter()` bắt đầu chạy sẽ không được truyền vào `callback`. Trong lúc `filter()` đang chạy, nếu những phần tử hiện tại của

Khảo sát MDN



Hãy giúp chúng tôi hiểu 10 nhu cầu hàng đầu của các nhà phát triển Web và nhà thiết kế.

Tham gia khảo sát

## Lọc bỏ các giá trị nhỏ



Ví dụ sau sẽ dùng `filter()` để tạo một mảng lọc không có các phần tử nào nhỏ hơn 10.

```
1 function isBigEnough(value) {  
2   return value >= 10;  
3 }  
4  
5 var filtered = [12, 5, 8, 130, 44].filter(isBigEnough);  
6 // filtered is [12, 130, 44]
```

## Lọc các giá trị không hợp lệ khỏi JSON

Ví dụ sau sẽ dùng hàm `filter()` để lọc lại các phần tử của JSON chỉ chứa `id` có giá trị số và khác 0.

```
1 var arr = [  
2   { id: 15 },  
3   { id: -1 },  
4   { id: 0 },  
5   { id: 3 },
```

Khảo sát MDN



Hãy giúp chúng tôi hiểu 10 nhu cầu hàng đầu của các nhà phát triển Web và nhà thiết kế.

Tham gia khảo sát



```
13
14 function isNumber(obj) {
15     return obj !== undefined && typeof(obj) === 'number' && !isNaN(obj);
16 }
17
18 function filterByID(item) {
19     if (isNumber(item.id) && item.id !== 0) {
20         return true;
21     }
22     invalidEntries++;
23     return false;
24 }
25
26 var arrByID = arr.filter(filterByID);
27
28 console.log('Filtered Array\n', arrByID);
29 // Filtered Array
30 // [{ id: 15 }, { id: -1 }, { id: 3 }, { id: 12.2 }]
31
32 console.log('Number of Invalid Entries = ', invalidEntries);
33 // Number of Invalid Entries = 5
34
```

## Khảo sát MDN



Hãy giúp chúng tôi hiểu 10 nhu cầu hàng đầu của các nhà phát triển Web và nhà thiết kế.

**Tham gia khảo sát**

```
5  */
6  function filterItems(query) {
7    return fruits.filter(function(el) {
8      return el.toLowerCase().indexOf(query.toLowerCase()) > -1;
9    })
10 }
11
12 console.log(filterItems('ap')); // ['apple', 'grapes']
13 console.log(filterItems('an')); // ['banana', 'mango', 'orange']
```

Ví dụ ở trên với ES2015 [🔗](#)

```
1  const fruits = ['apple', 'banana', 'grapes', 'mango', 'orange'];
2
3  /**
4   * Array filters items based on search criteria (query)
5   */
6  const filterItems = (query) => {
7    return fruits.filter((el) =>
8      el.toLowerCase().indexOf(query.toLowerCase()) > -1
9    );
10 }
```

Khảo sát MDN



Hãy giúp chúng tôi hiểu 10 nhu cầu hàng đầu của các nhà phát triển Web và nhà thiết kế.

Tham gia khảo sát

# Polyfill



`filter()` chỉ được thêm vào đặc tả ECMA-262 phiên bản thứ 5; cho nên nó có thể không tồn tại trong một số hiện thực (implementation) của đặc tả. Bạn có thể xoay sở bằng cách thêm vào đoạn code bên dưới vào đầu script của bạn, cho phép sử dụng `filter()` tại những nơi mà nó không được hỗ trợ sẵn. Giải thuật trong hàm polyfill này chính xác với đặc tả trong ECMA-262, 5th edition, với yêu cầu `fn.call` trả về giá trị ban đầu của `Function.prototype.bind()`, và `Array.prototype.push()` không bị thay đổi.

```
1  if (!Array.prototype.filter){
2    Array.prototype.filter = function(func, thisArg) {
3      'use strict';
4      if ( ! ((typeof func === 'Function' || typeof func === 'function') && this) )
5        throw new TypeError();
6
7      var len = this.length >>> 0,
8          res = new Array(len), // preallocate array
9          t = this, c = 0, i = -1;
10     if (thisArg === undefined){
11       while (++i !== len){
12         // checks to see if the key was set
13         if (i in this){
```

Khảo sát MDN



Hãy giúp chúng tôi hiểu 10 nhu cầu hàng đầu của các nhà phát triển Web và nhà thiết kế.

Tham gia khảo sát



```
22     // checks to see if the key was set
23     if (i in this){
24         if (func.call(thisArg, t[i], i, t)){
25             res[c++] = t[i];
26         }
27     }
28 }
29 }
30
31 res.length = c; // shrink down array to proper size
32 return res;
33 };
34 }
```



Đặc tả 

Khảo sát MDN



Hãy giúp chúng tôi hiểu 10 nhu cầu hàng đầu của các nhà phát triển Web và nhà thiết kế.


Tham gia khảo sát

ECMAScript Latest Draft (ECMA-262)

The definition of 'Array.prototype.filter' in that specification.

D

Draft



## Tương thích trình duyệt

[Update compatibility data on GitHub](#)

filter	
Chrome	Yes
Edge	12
Firefox	1.5
IE	9
Opera	Yes
Safari	Yes
WebView Android	Yes

Khảo sát MDN



Hãy giúp chúng tôi hiểu 10 nhu cầu hàng đầu của các nhà phát triển Web và nhà thiết kế.

Tham gia khảo sát

Samsung Internet Android	Yes	
nodejs	Yes	



Full support

## Tương tự

- `Array.prototype.forEach()`
- `Array.prototype.every()`
- `Array.prototype.some()`
- `Array.prototype.reduce()`

Khảo sát MDN



Hãy giúp chúng tôi hiểu 10 nhu cầu hàng đầu của các nhà phát triển Web và nhà thiết kế.

Tham gia khảo sát