

 Cập nhật tháng 8 năm 2024

[Bài Đọc] Filter

Phương thức **filter** là một trong những phương thức phổ biến và mạnh mẽ trong JavaScript, được sử dụng để tạo một mảng mới từ các phần tử của mảng gốc mà thỏa mãn một điều kiện được xác định bởi hàm callback.

1. Cú pháp

```
array.filter(callback(element, index, array), thisArg)
```

2. Tham số

- **callback**: Hàm được gọi cho từng phần tử trong mảng. Hàm này nhận 3 tham số:
 - **element**: Phần tử hiện tại của mảng.
 - **index** (*tùy chọn*): Chỉ số của phần tử hiện tại.
 - **array** (*tùy chọn*): Mảng mà phương thức **filter** được gọi.
- **thisArg** (*tùy chọn*): Giá trị được sử dụng làm **this** trong hàm **callback**

3. Giá trị trả về

- Một mảng mới chứa tất cả các phần tử từ mảng ban đầu thỏa mãn điều kiện được trả về từ hàm **callback**.

4. Đặc điểm

- Không thay đổi mảng gốc:
 - **filter** không thay đổi mảng gốc mà trả về một mảng mới
- Duyệt qua tất cả các phần tử:
 - **filter** kiểm tra từng phần tử trong mảng ban đầu và chỉ giữ lại những phần tử thỏa mãn điều kiện
- Không hoạt động trên mảng rỗng:
 - Nếu mảng gốc là rỗng, **filter** sẽ trả về một mảng rỗng

5. Ví dụ cơ bản

- Lọc số chẵn

```
const numbers = [1, 2, 3, 4, 5];
const evenNumbers = numbers.filter(predicate: (num) => num % 2 === 0);
console.log(...data: evenNumbers); // Output: [2, 4]
```

- Lọc tên dài hơn 4 ký tự

```
const names = ["Alice", "Bob", "Charlie", "David"];
const longNames = names.filter(predicate: (name) => name.length > 4);
console.log(...data: longNames); // Output: ["Alice", "Charlie", "David"]
```

- Sử dụng chỉ số (index) trong callback

```
const numbers = [10, 20, 30];
const result = numbers.filter(predicate: (num, index) => index % 2 === 0);
console.log(...data: result); // Output: [10, 30]
```

6. Tài liệu tham khảo

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter

Danh sách các bài học

