

Hiểu về map, filter và reduce trong JavaScript

27 tháng 06, 2017

Khi làm việc với các dự án JavaScript, chắc hẳn bạn đã trải qua các tình huống cần xử lý các dữ liệu. Những lúc ấy hẳn bạn sẽ sử dụng `for-loops` để giải quyết vấn đề, tuy nhiên nó sẽ ngày càng trở nên rối và làm cho code phình ra. Hầu hết trong các trường hợp, sử dụng `map`, `filter` hay `reduce` lại làm cho công việc trở nên dễ dàng hơn. Code của bạn sẽ "sạch" hơn và dễ đọc hơn. Ít nhất sẽ được như thế, khi mà bạn đã hiểu cách chúng làm việc và khi nào nên sử dụng chúng.

Một số nguyên tắc mà tôi sử dụng để xác định nên dùng hàm nào:

- Nếu tôi có 1 mảng và tôi muốn xử lý từng biến trong mảng theo cùng 1 cách, trả về các giá trị sau xử lý (số lượng đúng bằng số lượng phần tử ban đầu của mảng) thì tôi sẽ sử dụng `map`.
- Nếu tôi đã có 1 mảng nhưng tôi chỉ muốn lấy các phần tử theo 1 tiêu chuẩn nhất định, tôi sử dụng `filter`.
- Nếu tôi đã có 1 mảng nhưng tôi muốn sử dụng các giá trị trong mảng để tạo ra vài thứ khác hoàn toàn mới, tôi sử dụng `reduce`.

Các quy tắc này nghe có vẻ mơ hồ, do đó để hiểu cách ứng dụng của từng hàm, cách tối nhất luôn là tìm hiểu qua các ví dụ. Tôi sẽ sử dụng 1 mảng mẫu như dưới đây:

```
const animals = [
  {
    "name": "cat",
    "size": "small",
    "weight": 5
  },
  {
    "name": "dog",
    "size": "small",
    "weight": 10
  },
  {
    "name": "lion",
    "size": "medium",
    "weight": 150
  },
  {
    "name": "elephant",
    "size": "big",
    "weight": 5000
  }
]
```

Hàm map()

Giả sử ta cần 1 mảng chứa tên của các con vật. Nếu sử dụng `for-loop` :

```
let animal_names = [];

for (let i = 0; i < animals.length; i++) {
  animal_names.push(animals[i].name);
}
```

TechMaster Vietnam · Thường phản hồi trong...

Xin chào! Chúng tôi có thể giúp gì cho bạn?

Tiếp tục dưới tên Groove

Không phải bạn ư? Hãy đăng nhập vào Messenger

```
let animal_names = animals.map((animal, index, animals) => {
  return animal.name
})
```

Hàm `map()` nhận vào 3 tham số (theo thứ tự):

- Phần tử hiện tại của mảng.
- Chỉ số của phần tử hiện tại trong mảng.
- Mảng ban đầu.

1 số ưu điểm khi dùng `map()` thay vì `for-loop`:

- Với `map()`, ta không cần quản lý trạng thái của vòng lặp như với `for-loop`.
- Ta không cần sử dụng chỉ số để truy cập vào đúng phần tử trong mảng.
- Ta không cần tạo mảng mới và `push()` từng giá trị vào. `map()` trả về 1 mảng mới với các giá trị đã được chỉnh sửa, do đó có thể dễ dàng gán nó cho 1 biến khác.

Có 1 điều bạn cần phải ghi nhớ, đó là luôn sử dụng `return` để trả về từng giá trị sau chỉnh sửa, nếu không mảng trả về cuối cùng sẽ chỉ chứa các giá trị `undefined`.

Tham khảo các [khóa học lập trình](#) online, onlab, và [thực tập lập trình](#) tại TechMaster

Hàm filter()

Tiếp tục với 1 bài toán khác, lần này ta muốn 1 mảng chỉ chứa các con vật nhỏ. Nếu dùng `for-loop`:

```
let small_animals = [];

for (let i = 0; i < animals.length; i++) {
  if (animals[i].size === "small") {
    small_animals.push(animals[i])
  }
}
```

Sử dụng hàm `filter()`:

```
let small_animals = animals.filter((animal) => {
  return animal.size === "small"
})
```

Hàm `filter` cũng nhận vào 3 tham số như hàm `map()`, tuy nhiên trong trường hợp này ta chỉ sử dụng tham số đầu tiên. Nó cũng có các lợi ích như hàm `map()`, cũng sử dụng câu lệnh `return` trong thân hàm. Tuy nhiên, với `filter()`, ta cần đảm bảo câu lệnh `return` trả về `true` hoặc `false` vì nó là điều kiện để "lọc" ra các giá trị phù hợp trong mảng.

Hàm reduce()

Hãy đến với bài toán thứ 3: bạn cần tính tổng trọng lượng của các con vật. Hãy bắt đầu với `for-loop`:

```
let total_weight = 0;

for (let i = 0; i < animals.length; i++) {
  total_weight += animals[i].weight
}
```

Và hàm `reduce()`:

TechMaster Vietnam · Thường phản hồi trong...



Xin chào! Chúng tôi có thể giúp gì cho bạn?

Tiếp tục dưới tên Groove

[Không, phải bạn ư? Hãy đăng nhập vào Messenger](#)

```
let total_weight = animals.reduce((weight, animal, index, animals) => {
  return weight += animal.weight
}, 0)
```

Với hàm `reduce()` , các tham số truyền vào sẽ khác 1 chút so với 2 hàm kể trên:

- Tham số đầu tiên là giá trị khởi tạo. Ta cần set giá trị khởi tạo ở cuối hàm. Trong ví dụ trên là `0` . Nó có thể là bất cứ giá trị nào.
- Tham số thứ 2 là phần tử hiện tại trong mảng.
- Tham số thứ 3 và 4 giống với 2 hàm kể trên.

Nói lại 1 lần nữa, những lợi ích khi sử dụng `reduce()` cũng tương tự như `map()` , `filter()` , làm cho code ngắn hơn, dễ đọc hơn. Lần này, hàm `reduce()` trả về giá trị `weight` sau khi cộng, thứ sẽ lại trở thành tham số đầu tiên cho hàm `reduce()` kế tiếp. Khi chạy đến hàm `reduce()` với giá trị cuối cùng trong mảng, nó sẽ trả về tổng khối lượng các con vật (giá trị `weight` cuối cùng) và gán vào biến `total_weight` .

Kết luận

Với các ví dụ đơn giản kể trên, ta đã hiểu hơn về cách sử dụng các hàm `map` , `filter` và `reduce` . Các hàm này sẽ càng tối ưu hơn với các dữ liệu hay mã code nhiều, phức tạp. Tôi khuyên bạn nên tập làm quen với chúng nếu bạn còn bỡ ngỡ. Rồi bạn sẽ nhận ra chúng giúp cho code của bạn sạch sẽ, dễ đọc và dễ bảo trì hơn.

Bài viết được dịch từ: <https://hackernoon.com/understanding-map-filter-and-reduce-in-javascript-5df1c7eee464>



Kid
32 bài viết

Bài viết liên quan

[Năm 2019 người mới bắt đầu nên học ngôn ngữ lập trình nào ?](#)

[Top 10 Project Về JavaScript](#)

[Tạo Cloud Backend Cho Ứng Dụng Android trong Firebase](#)

[Tổng hợp những framework Java phổ biến nhất](#)

[Thực hành Lab giữa kỳ các lớp Web3, Web4 và Web tối - Đợt 2](#)

Khoá học hay

[Python phân tích xử lý dữ liệu](#)

[Xây dựng ứng dụng Web bán hàng bằng Golang](#)

[SQL cơ bản qua các ví dụ](#)

[Web Frontend cho người mới học](#)

[Web Frontend cơ bản cho người mới Buổi tối](#)

TechMaster Vietnam · Thường phản hồi trong...

X

Xin chào! Chúng tôi có thể giúp gì cho bạn?

Tiếp tục dưới tên Groove

[Không, phải bạn ư? Hãy đăng nhập vào Messenger](#)

7/29/2019		Hiểu về map, filter và reduce trong JavaScript	
<u>Công ty TNHH TechMaster Vietnam</u>	<u>Giảng viên</u>	<u>Tuyển sinh: 038 309 7229</u>	<u>CS1: Tầng 12A, tòa nhà Viwaseen</u>
<u>Ltd</u>	<u>Quy định</u>	<u>huyen@techmaster.vn</u>	<u>Tower, số 48, Tố Hữu, Lê Văn Lương</u>
Số ĐKDN: 0105392153	<u>Hướng dẫn mua khóa học</u>	<u>Tư vấn đào tạo doanh nghiệp: 0902</u>	<u>kéo dài, Hà Nội</u>
Ngày cấp: 4-7-2011	<u>Ưu đãi và hoàn trả học phí</u>	<u>209 011</u>	<u>CS2: 14 ngõ 4, Nguyễn Đình Chiểu,</u>
Nơi cấp: Sở kế hoạch - đầu tư Hà nội	<u>Bảo vệ thông tin khách hàng</u>	<u>cuong@techmaster.vn</u>	<u>Quận Hai Bà Trưng, Hà Nội</u>
Người đại diện pháp luật: Lê Minh			
Thu			
Phụ trách nội dung: Trịnh Minh			
Cường			



TechMaster Vietnam · Thường phản hồi trong...✕

 Xin chào! Chúng tôi có thể giúp gì cho bạn?

Tiếp tục dưới tên Groove

[Không, phải bạn ư? Hãy đăng nhập vào Messenger](#)