



Published May 28th, 2018 11:18 AM - 2 min read



9 trick nhỏ giúp code JS gọn hơn

<u>JavaScript</u> 000

Bài viết đc dịch từ: https://medium.freecodecamp.org/9-neat-javascript-tricks-e2742f2735c3

1. Clear hoặc truncate một array

Chỉ bằng cách thay đổi length, bạn có thể clear hoặc truncate một array mà ko cần reassign nó.

```
const arr = [11, 22, 33, 44, 55, 66];
// truncanting
arr.length = 3;
console.log(arr); //=> [11, 22, 33]
// clearing
arr.length = 0;
console.log(arr); //=> []
console.log(arr[2]); //=> undefined
```

2. Set giá trị mặc định của parameter với object destructuring

Nhiều khi, bạn cần phải truyển một object với các giá trị mặc định cho một hàm kiểu như:

```
doSomething({ foo: 'Hello', bar: 'Hey!', baz: 42 });
function doSomething(config) {
  const foo = config.foo !== undefined ? config.foo : 'Hi';
  const bar = config.bar !== undefined ? config.bar : 'Yo!';
  const baz = config.baz !== undefined ? config.baz : 13;
  // ...
}
```

Đây là một cách cũ nhưng hiệu quả. Cách trên vẫn chạy bình thường, tuy nhiên lại dài dòng một cách ko cần thiết. Với object destructuring của es6, bạn có thể viết ngắn gọn lại như sau:

```
function doSomething({ foo = 'Hi', bar = 'Yo!', baz = 13 }) {
}
```

Và kể cả bạn muốn cho object config là optional, điều đó có thể thực hiện dễ dàng bằng cách:

```
function doSomething({ foo = 'Hi', bar = 'Yo!', baz = 13 } = {}) {
```

3. Object destructuring với phần tử của mảng

Bạn có thể assign các phần tử của mảng tới từng biến riêng biệt bằng cách dùng object destructuring:

```
const csvFileLine = '1997,John Doe,US,john@doe.com,New York';
const { 2: country, 4: state } = csvFileLine.split(',');
```

4. Switch với các khoảng giá trị

Một trick đơn giản để so sánh các khoảng giá trị trong câu lệnh switch:

```
function getWaterState(tempInCelsius) {
  let state;

switch (true) {
    case (tempInCelsius <= 0):
        state = 'Solid';
        break;
    case (tempInCelsius > 0 && tempInCelsius < 100):
        state = 'Liquid';
        break;
    default:
        state = 'Gas';
  }
  return state;
}</pre>
```

5. Await nhiều hàm async với async/await

Bạn có thể đợi nhiều hàm async xử lý xong bằng cách sử dụng Promise.all:

```
await Promise.all([anAsyncCall(), thisIsAlsoAsync(), oneMore()])
```

6. Tạo pure object

Bạn có thể tạo một 100% pure object, ko kế thừa bất kì property hay method nào từ Object (ví dụ constructor, toString() ...)

```
const pureObject = Object.create(null);
console.log(pureObject); //=> {}
console.log(pureObject.constructor); //=> undefined
console.log(pureObject.toString); //=> undefined
console.log(pureObject.hasOwnProperty); //=> undefined
```

7.Format JSON code

JSON.stringify ko chỉ đơn giản là biết một object thành string. Bạn có thể làm đẹp output JSON với nó:

```
const obj = {
  foo: { bar: [11, 22, 33, 44], baz: { bing: true, boom: 'Hello' } }
};
// Tham số thứ 3 là số space để indent output JSON
JSON.stringify(obj, null, 4);
// =>"{
         "foo": {
// =>
             "bar": [
// =>
// =>
                 11,
// =>
                 22,
// =>
                 33,
// =>
                 44
// =>
             ],
             "baz": {
// =>
                 "bing": true,
// =>
                 "boom": "Hello"
// =>
// =>
             }
// =>
// =>}"
```

8. Remove các phần tử bị trùng khỏi mảng

Bằng cách sử dụng Set của ES2015 và spread operator, bạn có thể dễ dàng remove các phần tử trùng nhau trong mảng:

```
const removeDuplicateItems = arr => [...new Set(arr)];
removeDuplicateItems([42, 'foo', 42, 'foo', true, true]);
//=> [42, "foo", true]
```

9. Flatten mảng đa chiều

Bạn có thể dễ dàng flatten một array bằng spread operator:

```
const arr = [11, [22, 33], [44, 55], 66];
const flatArr = [].concat(...arr); //=> [11, 22, 33, 44, 55, 66]
```

Tuy nhiên cách trên chỉ hoạt đông với mảng 2 chiều. Nhưng hằng cách cử dụng đê quy hạn có thể khiến nó hoạt



Related

↑ +7 **↓**

```
Một số thủ thuật JavaScript với ES6
    ECMAScript 7: những tính năng đáng chú ý
    Đào Tùng
                                                                  Dinh Tai
    5 min read
                                                                  3 min read

◆ 1565  
◆ 8  
◆ 2  
◆ 9

    ② 3205 ③ 10 № 13 ♠ 18
    Trở thành Functional Programmer - Phần 6
                                                                  Một số tips tối ưu tốc độ khi làm việc với array trong
VIBLO
                                                                                                             i → Sign In/Sign up
                                         Search Viblo
    18 min read
                                                                  5 min read
    ● 1339 ● 8 ● 3 ♦ 5
```

More trom Rni i uai Ha

Cùng tìm hiểu về slice, splice và cú spread syntax trong

lavacorint

Bui Thai Ha

3 min read

Tự build một thư viện validate form bất đồng bộ chỉ

-.100 dàng codo với Poact Hooks

Bui Thai Ha

6 min read

● 169 <> 2
● 1
● 2

<u>Tạo một drag and drop list trong React mà ko dùng bất kì thư viện nào</u>

Bui Thai Ha

3 min read

React Hooks, tại sao chúng ta nên sử dụng chúng

Bui Thai Ha

5 min read

② 2539 **③** 5 **№** 1 **♦** 4

Comments

RESOURCES SERVICES

Posts <u>Organizations</u> <u>Viblo Code</u>

<u>Questions</u> <u>Tags</u> Viblo CV

<u>Videos</u> <u>Authors</u>

<u>Discussions</u> <u>Recommend System</u>

Tools Machine Learning

System Status

MOBILE APP





LINKS







© 2019 Viblo. All rights reserved.

<u>Feedback</u>

Help

FAQs

RSS

Terms

DMCA (1) PROTECTED

