

[Trang Chủ](#) / [Hướng Dẫn](#) / [ES6 Cơ Bản](#) / ES6 import và export

ES6 import và export

Nguyễn Đạt 23 March 2019

ES6 giới thiệu câu lệnh `import` và `export` hỗ trợ lập trình viên có thể quản lý code theo từng **mô-đun**.

JavaScript Mô-Đun Là Gì

Mô-đun là một phương pháp lập trình mà trong đó các đoạn code liên quan được tách ra các phần khác nhau (hay còn gọi là mô-đun).

Trước ES6 một số thư viện JavaScript hỗ trợ các cách viết code khác nhau để hỗ trợ quản lý các mô-đun (module).

Ví dụ Node.js hỗ trợ sử dụng `require()` và `module.exports()`.

Tạo mô-đun trong file `say_hello.js`

```
var sayHello = function (name) {  
  console.log("Xin chào! Tên tôi là " + name);  
}  
  
module.exports = {  
  sayHello: sayHello  
};
```

Sử dụng mô-đun tạo ra ở trên trong file `hello.js`:

```
var sayHello = require('./say_hello.js');  
sayHello("Peter");
```

Sử dụng chương trình Node.js để chạy file `hello.js`:

```
$ node hello.js  
// Xin chào! Tên tôi là Peter
```

Chúng ta có thể sử dụng câu lệnh `import` và `export` trong phiên bản ES6 để thay thế cho `require()` và `module.export()`.

Lưu ý: Để sử dụng tính năng `export` và `import` trong ES6 bạn cần sử dụng các trình duyệt hỗ trợ ES6 như Edge, Chrome, Firefore và nâng cấp trình duyệt lên phiên bản mới nhất.

ES6 Câu Lệnh `export`

Câu lệnh `export` dùng để *xuất* ra một mô-đun:

```
let sayHello = function (name) {  
  console.log("Xin chào! Tên tôi là " + name);  
}  
  
export { sayHello };
```

Bạn có thể export nhiều mô-đun trong cùng một file:

```
let sayHello = function (name) {  
  console.log("Xin chào! Tên tôi là " + name);  
}  
  
let sayGoodbye = function () {  
  console.log("Chào tạm biệt!");  
}  
export { sayHello, sayGoodbye };
```

ES6 Câu Lệnh `import`

Câu lệnh `import` trong ES6 dùng để *nhập* vào mô-đun từ một file cho trước (có thể là từ một thư viện hay mô-đun mà chúng ta tự định nghĩa).

Ví dụ để nhập vào mô-đun `sayHello` được tạo ra ở ví dụ trên:

```
import { sayHello } from 'say_hello.js';
```

Đoạn code trên sẽ *nhập* vào mô-đun `sayHello` từ tập tin `say_hello.js` và gán giá trị của mô-đun này cho biến `sayHello`.

Bạn có thể tùy ý đặt tên biến khi import mô-đun `sayHello` sử dụng từ khoá `as`:

```
import { sayHello as sayHelloFunction } from 'say_hello.js';
```

Lưu ý: Trong trường hợp có nhiều mô-đun được tạo ra trong cùng một file thì bạn có thể tùy ý lựa chọn số lượng mô-đun nhập vào chứ không cần thiết phải nhập tất cả.

JavaScript ES6 Frontend

[Chỉnh sửa](#)

[« ES6 Arrow Function](#)

[ES6 Destructuring Assignment »](#)

1 Phản Hồi

[Tuấn Nguyễn](#) 26 October 2018

Vậy thực sự 2 cái này khác nhau gì vậy ạ

[Thêm bình luận](#)

0

Vote

Thêm Phản Hồi

Đăng nhập để thêm nội dung...

Gửi

Danh Sách Bài Học

1. [ES6 Const và Let](#)
2. [ES6 Rest Parameter](#)
3. [ES6 Spread Operator](#)
4. [ES6 Default Parameter](#)
5. [ES6 Template Literal](#)
6. [ES6 Object Literal](#)
7. [ES6 Arrow Function](#)
8. [ES6 import và export](#)
9. [ES6 Destructuring Assignment](#)

[Giới Thiệu](#) [Blog](#) [Việc Làm](#) [Chủ Đề](#) [Người Dùng](#) [Liên Hệ](#)

Thiết kế và phát triển bởi [CodeHub.vn](#) © 2019