



Tháng 12 khai giảng

Lập Trình

RUBY

Full time: 540 Giờ

Part time: 240 Giờ

Create your future with code



Nguyen Luong Huy @Creat10n

Follow

★ 69

👤 2

✍ 7

Published Mar 14th, 11:31 AM - 4 min read

TABLE OF CONTENTS

👁 2.4K

💬 2

🔗 4

1. Giới Thiệu

2. Phân Biệt

2.1. Biến var

2.2. Biến let

2.3. Biến const

SUGGESTED ORGANIZATIONS

...



Sun\* Cyber Security Team

🔗 25

👤 4

👥 13



Mino

🔗 2

👤 2

👥 9



Emulate Infotech Pvt. Ltd.



+4



es6

JavaScript

es2015



## 1. Giới Thiệu

ES6 (viết tắt của ECMAScript 6) là một tập hợp các kỹ thuật nâng cao của Javascript.

ECMAScript do Hiệp hội các nhà sản xuất máy tính Châu Âu đề xuất là tiêu chuẩn của

va neu moi browser lai co cach chay javascript khac nhau thi cac trang web khong the hoạt động trên tất cả các trình duyệt đó được, vì vậy cần có một chuẩn chung để bắt buộc các browser phải phát triển dựa theo chuẩn đó.

Một trong những tính năng nổi bật của ES6 đó là sự bổ sung **let** và **const** phục vụ cho việc khai báo biến dữ liệu. Vậy tại sao lại cần tới **let** và **const** dù trước đó đã có **var** để khai báo biến, thì ở bài viết này chúng ta sẽ cùng tìm hiểu lý do của từng kiểu khai báo biến.



+4



## 2. Phân Biệt

### 2.1. Biến var

Như đã biết trong Vanilla Javascript, với từ khóa **var** chúng ta có thể khai báo đa dạng

Như đã biết trong vanilla javascript, với từ khóa `var` chúng ta có thể khai báo đa dạng các kiểu biến như number, string, boolean, etc. Trừ trường hợp được khai báo bên trong 1 function (khi đó biến `var` sẽ có scope là **function/locally scoped**), biến `var` sẽ có scope là **globally scoped**. Đặc biệt, biến `var` còn có thêm tính chất **hoisting**: nghĩa là dù khai báo ở đâu thì biến đều sẽ được đem lên đầu scope trước khi code được thực hiện.

VIBLO

Posts

Questions

Discussions

Search Viblo



Sign In/Sign up

▲  
+4  
▼



```
console.log (greeting);  
var greeting = "say hello";
```

sẽ được biên dịch là:

```
var greeting;  
console.log(greeting); // greeting is undefined  
greeting = "say hello";
```

Nhưng có một vấn đề với biến `var` . Giả sử ta có ví dụ như sau:

```
var greeting = "hey hi";  
var times = 4;  
  
if (times > 3) {  
  var greeting = "say Hello instead";  
}  
  
console.log(greeting); //"say Hello instead"
```

Vì thỏa điều kiện `if` nên `greeting` khi này sẽ có giá trị là `"say Hello instead"` và sẽ vẫn giữ nguyên giá trị này sau khi thoát ra khỏi block `if` . Điều này sẽ không có gây vấn

VIBLO

Posts

Questions

Discussions

Search Viblo



Sign In/Sign up

▲  
+4  
▼



## 2.2. Biến let

Một trong những nguyên nhân khiến `let` có thể thay thế `var` để xử lý vấn đề nêu trên là vì biến `let` được khai báo sẽ có scope là **block scoped** chứ không phải globally hay locally scoped.

```
let greeting = "say Hi";
let times = 4;

if (times > 3) {
  let hello = "say Hello instead";
  console.log(hello); // "say Hello instead"
}
console.log(hello); // hello is not defined
```

Chúng ta có thể thấy là đối với biến có scope là block scoped nếu ra khỏi scope được khai báo thì sẽ không thể sử dụng được nữa.

`let` cho phép chúng ta cập nhật giá trị của biến chứ không cho phép chúng ta tái khai

VIBLO

Posts Questions Discussions

Search Viblo



Sign In/Sign up

+4

```
let greeting = "say Hi";
console.log(greeting); // "say Hi"
```

```
greeting = "say Hello instead";
console.log(greeting); // "say Hello instead"
```

```
-----

let greeting = "say Hi";
let greeting = "say Hello instead"; // error: Identifier 'greeting' has already been declared
```

Tuy nhiên, đối với các block khác nhau thì việc tái khai báo biến sẽ không sinh ra lỗi vì đối với từng scope, mỗi biến sẽ được xem xét là 1 biến riêng khác biệt.

```
let greeting = "say Hi";
if (true) {
  let greeting = "say Hello instead";
  console.log(greeting); // "say Hello instead"
}
console.log(greeting); // "say Hi"
```

Giống với `var`, `let` cũng có tính **hoisting** tuy nhiên lại khác nhau ở chỗ thay vì `var` được khởi tạo với giá trị là `undefined` thì `let` sẽ không có bất kỳ giá trị khởi tạo nào.

VIBLO

Posts Questions Discussions

Search Viblo



Sign In/Sign up

## 2.3 Biến const

+4



Tương tự với `let` cũng có scope là **block scoped**, và **hoisting** thì chúng ta có thêm 1 kiểu khai báo biến nữa là `const`. Trong biến `const` nếu trường hợp kiểu của biến là **primitive** (bao gồm string, number, boolean, null, và undefined) thì chúng ta sẽ không thể tái khai báo hay cập nhật giá trị mới để thay thế cho giá trị trước đó của biến.

```
const greeting = "say Hi";
greeting = "say Hello instead"; // error : Assignment to constant variable.

-----

const greeting = "say Hi";
const greeting = "say Hello instead"; // error : Identifier 'greeting' has already
```

Đối với trường hợp kiểu biến là **reference** (bao gồm object, array, và function) thì tuy không thể tái khai báo hay cập nhật giá trị của biến nhưng chúng ta vẫn có thể cập nhật giá trị cho thuộc tính của biến đó.

```
const greeting = {
  message : "Hello",
```

VIBLO

Posts

Questions

Discussions

Search Viblo



→ Sign In/Sign up



```
greeting.message = "say Hello instead";
console.log(greeting); // {message:"say Hello instead",number:"five"}
```



## Related

### Hoisting trong javascript

Nguyễn Hữu Sử

6 min read

👁 686 🗨 6 💬 2 🔄 15

### ES6 - The Good Part (Phần 1)

Nguyễn Phúc Lương

8 min read

👁 1339 🗨 5 💬 3 🔄 9

### Giới thiệu Typescript - Sự khác nhau giữa Typescript ...

Uy Tran

7 min read

👁 9392 🗨 1 💬 0 🔄 5

### Hoisting trong javascript - gà có trước hay trứng (phần 1)

Pham Quang Hung

2 min read

👁 415 🗨 1 💬 4 🔄 3

## More from Nguyen Luong Huy

### Interactive effects with CSS clip path

Nguyen Luong Huy

### Setting up NeoVim on Ubuntu

Nguyen Luong Huy

### Giới thiệu Retrofit và một số tính năng nổi trội

Nguyen Luong Huy

### Các cách khai báo hàm trong JavaScript

Nguyen Luong Huy

2 min read

9 0 0 0

9 min read

81 0 0 1

4 min read

24 0 0 0

13 min read

179 1 0 3

VIBLO

Posts

Questions

Discussions

Search Viblo



Sign In/Sign up

Login to comment

VIBLO

Posts

Questions

Discussions

Search Viblo



Sign In/Sign up

0 | Reply Share ...



Nguyen Luong Huy @Creat10n

Mar 14th, 1:00 PM

Thanks bạn nhé, mình vừa update lại rồi.

0 | Reply Share ...

## RESOURCES

Posts

Questions

Videos

Discussions

Tools

System Status

Organizations

Tags

Authors

Recommend System

Machine Learning

## SERVICES



Viblo Code



Viblo CV

## MOBILE APP



## LINKS



© 2019 Viblo. All rights reserved.

[Feedback](#) [Help](#) [FAQs](#) [RSS](#) [Terms](#)

DMCA PROTECTED

