


Saysua Blog

[HOME](#)[APPS](#)[ABOUT](#) 2013-08-28

Tìm hiểu về Callback trong JavaScript

Tôi bắt đầu chú ý đến callback của javascript trong thời gian làm đồ án tốt nghiệp ra trường. Khi đó đọc thông tin thấy js cũng viết server side được bằng cách sử dụng node.js, thấy cũng hay hay thế là dùng nó luôn. Và vậy là tôi bắt đầu bị callback làm cho đau đầu vì nó quá khác với cách viết code của những ngôn ngữ đã học như C hay Java. Từ khi biết về callback tôi mới nhận ra rằng thì ra mình đã sử dụng nó rất nhiều khi viết code với jQuery. Chắc hẳn ở đây, nhiều bạn đã sử dụng callback nhưng chưa để ý về nó. Trong khả năng của mình, tôi sẽ trình bày cho các bạn những kiến thức cơ bản về callback trong javascript.

Callback nó là cái gì?

Theo định nghĩa trên wikipedia: Callback là một phần mã thực thi được truyền vào như một tham số của một đoạn mã khác, nó sẽ được gọi lại (thực thi) tại một thời điểm thích hợp.

Chúng ta hãy cùng nhìn qua đoạn code jquery dưới đây:

```
1 $.get('ajax/test.html', function(data) {  
2     $('result').html(data);  
3     alert('Load was performed.');
```

```
4 });
```

Đây là một lệnh jQuery, nó sẽ gọi ajax với phương thức GET, lệnh này nhận 2 tham số: Đường dẫn để gọi ajax lấy nội dung và một function sẽ được thực thi khi lời gọi ajax thành công.

Tôi tạm giải thích cơ chế hoạt động của đoạn lệnh trên như sau:

- Trước tiên một lời gọi ajax được thực hiện để lấy nội dung của file ajax/test.html

- Sau khi lời gọi ajax thành công nó sẽ thực thi function được truyền vào với tham số data chính là nội dung của file ajax/test.html đã lấy về.

Do đó, tùy thuộc vào tốc độ lấy nội dung file mà callback function sẽ được thực hiện sớm hay muộn.

Function là object

Đến đây, tôi có một thắc mắc: Bằng cách nào trong javascript, function lại có thể được truyền vào như một tham số?

Bởi vì, trong javascript, function thực chất là một đối tượng (object), cụ thể hơn là một Function object . Nó được tạo ra bởi constructorFunction . Function object chứa một chuỗi là nội dung của function, như vậy thân của một function thực ra chỉ là một chuỗi kí tự.

```
1  cong = Function('a', 'b', 'return a+b');
2  cong(3,4) // => 7
```

Chính vì function cơ bản chỉ là một object nên ta có thể truyền nó như một tham số vào function khác.

Viết một callback

Sau đây là một ví dụ function sử dụng callback, bạn có thể theo nó để viết riêng một function sử dụng callback cho mình.

```
1  function goHome(vehicle, callback) {
2      alert("I'm going home by " + vehicle);
3
4      callback();
5  }
6
7  function playGame() {
8      alert("I'm playing game");
9  }
10
11  goHome("bicycle", playGame);
```

Ở đây, ta có function goHome nhận 2 tham số với tham số thứ 2 là một callback function. Khi ta chạy function goHome , trước tiên nó sẽ hiển thị một thông báo, sau đó

nó gọi đến callback function được truyền vào. Ta để ý ở đây, tham số truyền vào là callback không có dấu ngoặc, nó được thực thi trong function `goHome` bằng cách thêm dấu ngoặc vào sau.

Câu lệnh cuối cùng chính là câu lệnh thực thi function `goHome`, với callback truyền vào là tên function `playGame` được định nghĩa ở trên, function `playGame` khi thực thi nó hiện ra thông báo "I'm playing game".

Bạn có thể chạy thử đoạn code trên tại đây.

Nếu không truyền callback thì sao?

Bây giờ ta chạy function `goHome` ở trên mà không truyền vào tham số thứ 2.

```
1 function goHome(vehicle, callback) {
2     alert("I'm going home by " + vehicle);
3
4     callback();
5 }
6
7 goHome("bicycle");
```

Nếu bạn dùng trình duyệt Chrom và mở Developer Tool (Ctrl +Shift + J), khi chạy đoạn lệnh trên sẽ thấy một thông báo lỗi:

```
1 Uncaught TypeError: undefined is not a function
```

Chạy thử ví dụ ở đây. Nhớ bật Developer Tool để xem thông báo lỗi.

Nếu bạn muốn chương trình chỉ thực thi callback khi nó được thực sự truyền vào bạn có thể kiểm tra nó có tồn tại hay không trước khi gọi:

```
1 function goHome(vehicle, callback) {
2     alert("I'm going home by " + vehicle);
3
4     if (callback)
5         callback();
6 }
```

Để cho chắc chắn nó thực sự là một function được truyền vào, ta có thể kiểm tra kiểu của nó bằng lệnh `typeof`.

```
1 function goHome(vehicle, callback) {
2     alert("I'm going home by " + vehicle);
3
4     if (callback && typeof(callback) === "function")
5         callback();
6 }
```

Truyền tham số cho callback

Trong ví dụ ở trên, ta đã gọi function callback `playGame` mà không truyền cho nó một tham số nào cả. Tuy nhiên ta muốn gọi `playGame` với một trò chơi cụ thể thì làm thế nào? Đơn giản là truyền tham số cho nó như những function bình thường.

```
1 function goHome(vehicle, callback) {
2     alert("I'm going home by " + vehicle);
3
4     callback("Naruto");
5 }
6
7 function playGame(gameName) {
8     alert("I'm playing game " + gameName);
9 }
10
11 goHome("bicycle", playGame);
```

Bạn có thể chạy thử ví dụ trên ở đây.

Tổng kết

Trên đây là những trình bày về callback trong javascript, hy vọng nó sẽ giúp cho các bạn hiểu thêm về một khái niệm mới khi lập trình với javascript. Những thông tin này vẫn còn ở mức cơ bản, hẳn nhiều bạn có kinh nghiệm và hiểu biết sâu về nó sẽ thấy thiếu sót. Vì vậy mình rất mong được sự đóng góp hoặc trao đổi thêm của các bạn về vấn đề này.

Bài viết có tham khảo từ

- <http://recurial.com/programming/understanding-callback-functions-in-javascript/>
- <http://www.impressivewebs.com/callback-functions-javascript/>

NEWER

Viết chức năng login bằng tài khoản facebook không dùng Facebook SDK

OLDER


Xử lý mảng với Underscore.js

6 Comments

say sua

 Nguyễn Văn Được ▾

 Recommend

 Share

Sort by Best ▾

say sua requires you to verify your email address before posting. Send verification email to nguyenvanduoc071994@gmail.com ✕



Join the discussion...



Mạnh Hùng Vũ • a year ago

Bài viết rất hay, xin cảm ơn!

^ | ▾ • Reply • Share >



Tung Pham • 2 years ago

Cảm ơn bạn, xài mấy năm rồi mà bây giờ mới hiểu callback là gì =))

^ | ▾ • Reply • Share >



duy • 3 years ago

mình callback chính nó

```
function goHome(vehicle, callback) {  
  alert("I'm going home by " + vehicle);  
  callback();  
}
```

```
function playGame() {  
  alert("I'm playing game");  
}
```

```
goHome("bicycle", goHome('a'));
```

thì nó chỉ ra

I'm going home by a chứ không ra I'm going home by bicycle

^ | ▾ • Reply • Share >

**Thach Nguyen Xuan** Mod ➔ duy • 3 years ago

Chỗ này có thể giải thích như sau:

Khi bạn gọi:

```
>> goHome("bicycle", goHome('a'));
```

Nó sẽ thực hiện goHome('a') trước. khi đó nó alert lên câu "I'm going home by a". Tiếp tục nó gọi đến lện callback(), nhưng do bạn không truyền tham số callback vào nên nó bị lỗi:

```
>> TypeError: undefined is not a function
```

vì vậy chương trình dừng tại đó luôn. Không tiếp tục thực hiện được lệnh

```
>> goHome("bicycle", goHome('a'));
```

Nếu bạn sửa lại hàm goHome như sau thì chương trình sẽ alert 2 lần cho bạn:

```
function goHome(vehicle, callback) {
  alert("I'm going home by " + vehicle);
  if (callback) callback();
}
```

^ | v • Reply • Share ›

**Lam Lê** • 4 years ago

Cho mình hỏi

```
$.get('ajax/test.html', function(data) {
  $('<div>.result</div>').html(data);
  alert('Load was performed.');
```

});
- Ai truyền giá trị cho biến "data" ?

```
function goHome(vehicle, callback(data)) {
  alert("I'm going home by " + vehicle);
  data = 'something';
  callback(data);
}
```

- Muốn thêm đối số cho callback như trên thì viết ntn ?

^ | v • Reply • Share ›

**Thach Nguyen Xuan** Mod ➔ Lam Lê • 4 years ago

1. Về câu hỏi thứ nhất: Chính trong bản thân \$.get() nó có lệnh truyền giá trị cho biến "data". Lam chú ý chỗ này, khi mình gọi \$.get(...) là mình đang thực thi nó, chứ không phải mình đang định nghĩa function \$.get.

Mình có thể đoán rằng, định nghĩa \$.get nó như sau:

```
$.get = function(url, callback) {
```

```
// Làm gì đó ....
```

Như vậy khi bạn gọi:

see more

2 ^ | v • Reply • Share ›

Kiểm tra một phần tử tồn tại bằng JQuery

4 comments • 4 years ago •

Giới thiệu về saysua.com và mình

1 comment • 4 years ago •

AvatarJy Bhtrh — hay lắm bạn ạ.

Quản lý các gói thư viện phía client site bằng Bower

4 comments • 4 years ago •

A. một Thành Tựu Hoàn làm sao để lấy nh?

Khởi tạo dự án AngularJS với generator-boom

1 comment • 3 years ago •

Accepted Manuscript

Enter your email to subscribe my blog.

Subscribe

Ajax

Angular2

Angularis

Bower

Elastic search

Facebook

Godaddy

Gulp

Javascript

Jquery

Life

Logstash

Machine Learning

Mysql

Nginx

NodeJs

Nodejs

Nvm

OAuth

PHP

Python

Scrape

TeamCity

Visual Studio

<http://saysua.com/2013/08/28/js-tim-hieu-ve-callback-trong-javascript/>

RECENTS

Download file bằng Restangular và FileSaver
Kỷ luật là tự do
Những chức năng mới của ES6
Lưu log của nginx vào elastic search bằng logstash
Chạy Visual Studio Solution từ command line

ARCHIVES

March 2017 (1)
February 2017 (1)
December 2016 (3)
November 2016 (4)
October 2016 (1)
September 2016 (1)
April 2016 (1)
March 2016 (1)
December 2015 (4)
September 2015 (2)
August 2015 (2)
October 2014 (1)
January 2014 (1)
December 2013 (1)
November 2013 (1)
October 2013 (4)
September 2013 (3)
August 2013 (2)

