Khong Minh Quan @quankm1097 Follow
619 17 13

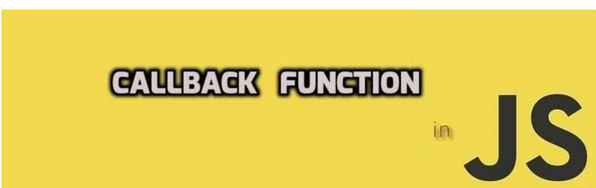
Published Nov 9th, 2018 9:37 PM - 5 min read
618 0 4

Thế nào là một Callback Function trong JavaScript

JavaScript

Đến với thế giới JavaScript, ta sẽ bắt gặp đủ loại function. Chắc hẳn bạn đã nghe tới Arrow Function, còn nếu chưa biết tới thì bạn có thể tham khao theo đường link này. Bên cạnh *Arrow Function*, JavaScript còn một loại function mà ta thường hay gặp, đó là **Callback Function**. Nghe tên có vẻ quen quen đúng không, hoặc có thể mình đã dùng rùi mà chưa để ý kĩ tới nó.

1. Tổng quát



Trước tiên, ta cần biết rằng một function trong JavaScript thực chất là một **first-class object**, nghĩa là function hay object hay array đều được lưu trữ như nhau. Function có thể được lưu trữ dưới dạng biến trong một Object hay một Array, và được truyền đi dưới dạng một tham số hoặc được gọi tới bằng một function khác. Đây cũng là bản chất của Callback Function trong JavaScript. Ta có thể nói răng CallBack Function là một kĩ thuật được ứng dụng rộng rãi nhất trong Functional Programming mà ta có thể bắt gặp ở bất kì đoạn code JavaScript nào. Nhưng được dùng rộng rãi thế vậy, đôi lúc ta vẫn hoài nghi về cách vận hành cũng như ứng dụng của Callback Function.

2. Thế nào là một Callback function hay Higher-order Function

Callback Function hay được gọi với tên khác là Higher-order Function, là một function được truyền vào một function (gọi là F1) khác dưới dạng tham số, và được gọi trong function F1 đó.

Đây một ví dụ sơ đăng nhất của callback function :

```
function notify(){
  console.log("Hello world");
}

function taskOne(callback){
  // Gọi notify function
  callback();
}

// Truyền function notify dưới dạng biến vào taskOne
taskOne(notify);
```

Theo ví dụ như trên, ta khai báo hai function tên là **notify()** và **taskOne()**. Khi ta gọi thực thi taskOne(), phía trong function lại gọi tới một function là **callback()**. Như ta để ý thì **callback** cũng là tên tham số của function **taskOne**, mà ở dòng thực thi ta truyền vào là notify.

Hay đây là một ví dụ thường gặp ở bất kì project nào sử dụng jQuery

```
$("#btn_1").click(function() {
  alert("Btn 1 Clicked");
});
```

Như đã thấy, ta truyền một function dưới dạng tham số vào phương thức click. Khi click dc thực thi sẽ gọi tới function mà ta khai báo.

3. Vậy Callback function hoạt đông như thế nào

Ta có thể truyền function như là một biến và return nó trong một function khác. Khi ta truyền callback function như là một tham số tới một function khác, ta chỉ truyền định nghĩa. Nó sẽ được thực thi khi ta truyền cả function dưới dang tham số

Và chúng ta đã có định nghĩa của function callback dưới dang tham số, ta có thể thực thi bất kì lúc nào trong function chứa nó.

Chú ý: Function Callback không được thi thức thi. Nó có tên là "Callback" mà nhỉ 😄 nên chỉ được thực thi khi function chứa nó gọi đến callback function.

3.1 Callback Function là Closure

Khi ta truyền callback function dưới dạng tham số tới một function khác, callback được thực thi trong body của function chứa nó với cái tên ta đặt ở nơi nhận tham số truyền vào. Như chúng ta đã biết, closure có thể truy cập đến scope của function, cho nên callback function có thể sử dụng các biến của function chứa nó hay thậm chí global scope

4. Nhưng cách gọi Callback function

1. Sử dụng anonymous functions

Quay lại đoạn code bên trên, jQuery sử dụng rất nhiều method mà trong đấy có tham số là callback function.

```
  alert("Btn 1 Clicked");
});
```

2. Sử dụng một callback function đã được đặt tên

Trong testFunction có một tham số là callback. Ta gọi đến function callback khi thực thi testFunction.

```
function namedCallback(){
   alert(“namedCallback()”);
}
function testFunction(callback){
   callback();
}
testFunction(namedCallback);
```

3. Truyền tham số tới callback function

Chúng ta có thể truyền tham số vào callback function như bất kì function khác.

```
var name = “Minh Quan”;

function whoWriteThis(param){
   alert(“whoWriteThis() called by “+param);
}

function testFunction(callback){
   callback();
}

testFunction(namedCallback(author));
```

4. Gọi nhiều callback function

Ta có thể truyền nhiều callback function dưới dạng tham số.

```
Var someUlr = …;

function successCallback(){
   //success code
}

function completeCallback(){
   //complete code
}

function errorCallback(){
   //error code
}

$.ajax({
   url: someUrl,
   success: successCallback,
   complete: completeCallback,
   error: errorCallback
})
```

5. Tóm gọn

ngôn ngữ event-driven như JavaScript. Và cuối cùng, ta phải nắm rõ callback function dùng để làm gì :

- Để thực hiện các tác vụ bất đồng bộ
- Cho những tác vụ event listeners/handlers

Bên cạnh đấy hiểu được Callback Function giúp chúng ta:

- Viết nhiều đoạn code dễ đọc hơn
- DRY—Do not repeat yourself
- Tăng khả năng bao trì
- Hiểu được luồng xử lý của nhiều thừ viện JavaScript

6. Tham khảo

https://nodefunction.com/callback-function/callback-function-in-javascript-node-js/
https://medium.com/@fotios.floros/explaining-javascript-callbacks-3d5a9ad52819
https://javascriptissexy.com/understand-javascript-callback-functions-and-use-them/

Tools Machine Learning

Login to comment

LINKS