

Modern Web (https://ithelp.ithome.com.tw/2018ironman/web)

重新認識 JavaScript (https://ithelp.ithome.com.tw/users/20065504/ironman/1259)系列 第 18 篇

DAY 18

重新認識 JavaScript: Day 18 Callback Function 與 IIFE

2018鐵人賽 (https://ithelp.ithome.com.tw/tags/articles/2018%E9%90%B5%E4%BA%BA%E8%B3%BD)

javascript (https://ithelp.ithome.com.tw/tags/articles/javascript)



Kuro Hsu (https://ithelp.ithome.com.tw/users/20065504/ironman)

2017-12-21 11:51:48

21844 瀏覽

(https://ithelp.ithome.com.tw/users/20065504/ironman)

本系列文章已重新編修，並在加入部分 ES6 新篇章後集結成書，有興趣的朋友可至天瓏書局選購，感謝大家支持。

購書連結 <https://www.tenlong.com.tw/products/9789864344130>
(https://www.tenlong.com.tw/products/9789864344130)

讓我們再次重新認識 JavaScript！

在上一篇文章當中，我們介紹了函式的參數與 arguments 物件，那麼今天的分享中，我們繼續來看看 Function 在 JavaScript 的各種不同面貌。

Callback Function

你可能常聽到人家在講 Callback function，但你真的知道 Callback function 是什麼嗎？其實 Callback function 跟一般的函式沒什麼不同，差別在於被呼叫執行的時機。

先前介紹事件的時候有說過，「JavaScript 是一個事件驅動 (Event-driven) 的程式語言」，而事件的概念就如同：

辦公室電話響了 (事件被觸發 Event fired) -> 接電話 (處理事件 Event Handler)

而寫成程式碼就類似：

```
// 註：這裡只是比喻，並沒有電話響這個事件 XD
Office.addListener( '電話響', function(){ /* 接電話 */ }, false);
```



鍵盤救國 AMA

iT邦幫忙 Ask Me Anything

(https://ithelp.ithome.com.tw/ama/?utm_source=ithelp&utm_medium=banner300x250)

鐵人賽得獎題目

- 入吾 Go 中：走訪 Go 語言內部實作 (https://ithelp.ithome.com.tw/users/20103524/ironman/2589?sc=iThelpR)
- 在 React 生態圈內打滾的一年 feat. TypeScript (https://ithelp.ithome.com.tw/users/20106935/ironman/2188?sc=iThelpR)
- 和艦長一起 30 天玩轉 GitLab (https://ithelp.ithome.com.tw/users/20120986/ironman/2733?sc=iThelpR)
- 區塊練起來-智能合約與DApp開發 (https://ithelp.ithome.com.tw/users/20119338/ironman/2150?sc=iThelpR)
- Android 十全大補 (https://ithelp.ithome.com.tw/users/20120419/ironman/2641?sc=iThelpR)
- Flutter---Google推出的跨平台框架，Android、iOS一起搞定 (https://ithelp.ithome.com.tw/users/20119550/ironman/2221?sc=iThelpR)
- 金魚都能懂的 CSS 選取器 - 金魚都能懂了你還怕學不會嗎 (https://ithelp.ithome.com.tw/users/20112550/ironman/2799?sc=iThelpR)
- 寫給PHP開發者的30堂網路爬蟲開發 (https://ithelp.ithome.com.tw/users/20103975/ironman/2903?sc=iThelpR)
- Machine Learning with TensorFlow on Google Cloud Platform (https://ithelp.ithome.com.tw/users/20120208/ironman/2380?sc=iThelpR)
- CTF 的三十道陰影 (https://ithelp.ithome.com.tw/users/20121059/ironman/2810?sc=iThelpR)

前往鐵人賽 (https://ithelp.ithome.com.tw/2020ironman?sc=iThelpR)

鐵人賽熱門訂閱

- 菜鳥工程師必修的 30 堂溝通課 (https://ithelp.ithome.com.tw/users/20040221/ironman/2236?sc=iThelpR)
- 前端工程師用 JavaScript 學演算法 (https://ithelp.ithome.com.tw/users/20106426/ironman/2136?sc=iThelpR)
- Python 程式交易 30 天新手入門 (https://ithelp.ithome.com.tw/users/2012

可以看到，Office 透過 `addEventListener` 方法註冊了一個事件，當這個事件被觸發時，它會去執行我們所指定的第二個參數，也就是某個「函式」(接電話)。

換句話說，這個函式只會在滿足了某個條件才會被動地去執行，我們就可以說這是一個 **Callback function**。

把函式當作另一個函式的參數？

經歷過各種「事件」的你，相信也發現了一件事，所謂的「**Callback function**」其實就是「**把函式當作另一個函式的參數，透過另一個函式來呼叫它**」。

什麼意思呢？除了剛剛介紹的「事件」以外，還有另一個經典的案例：

```
window.setTimeout( function(){ ... }, 1000);
```

如果我們希望隔某段時間之後，執行某件事，就可以透過 `window.setTimeout` 來幫助我們達成。

像是上面的範例中，`window.setTimeout` 帶有兩個參數，第一個是要做的事情，用一個函式來代表，第二個則是時間 (1/1000 秒, *milliseconds*)。而第一個參數的函式也是 **Callback function** 的一種：「在經過了某段時間之後，才執行的函式」。

當然，上面兩個範例中的 **Callback function** 我們也可以把它單獨抽出來定義：

```
var handler = function() {  
    /* 接電話 */  
};  
  
Office.addEventListener( '電話響', handler, false);
```

這樣會使你的程式看起來更清楚。

除了事件以外，還有另一個會需要用到 **Callback function** 的場景，就是「**控制多個函式間執行的順序**」。

什麼意思呢？來看個簡單的例子。

這裡定義了兩個 **function**：

```
var funcA = function(){  
    console.log('function A');  
};  
  
var funcB = function(){  
    console.log('function B');  
};  
  
funcA();  
funcB();
```

因為 `funcA` 與 `funcB` 都會立即執行，所以執行結果必定為：

```
"function A"  
"function B"
```

0536/ironman/2571?sc=iThelpR)

金魚都能懂的 CSS 選取器 - 金魚都能懂了你還怕學不會嗎

登入/註冊 (https://ithelp.ithome.com.tw/users/2011

2550/ironman/2799?sc=iThelpR)

突破困境：資安開源工具應用

(https://ithelp.ithome.com.tw/users/20118848/ironman/2682?sc=iThelpR)

金魚都能懂的這個網頁畫面怎麼切 - 金魚都能懂了你還怕學不會嗎

(https://ithelp.ithome.com.tw/users/20112550/ironman/2623?sc=iThelpR)

從 Hooks 開始，讓你的網頁 React 起來

(https://ithelp.ithome.com.tw/users/20103315/ironman/2668?sc=iThelpR)

讓 TypeScript 成為你全端開發的 ACE！

(https://ithelp.ithome.com.tw/users/20120614/ironman/2685?sc=iThelpR)

深入現代前端開發

(https://ithelp.ithome.com.tw/users/20103565/ironman/2527?sc=iThelpR)

分散式系統 - 在分散的世界中保持一致

(https://ithelp.ithome.com.tw/users/20121042/ironman/2792?sc=iThelpR)

📍 前往鐵人賽

(https://ithelp.ithome.com.tw/2020ironman?sc=iThelpR)

熱門問題

如何判斷該電腦是桌機還是筆電？

(https://ithelp.ithome.com.tw/questions/10198244)

Python Selenium 移除list中【不可見】的連結

(https://ithelp.ithome.com.tw/questions/10198274)

如何搬動win10 下方工作列的縮圖？

(https://ithelp.ithome.com.tw/questions/10198238)

總公司&分公司-AD架構觀念問題

(https://ithelp.ithome.com.tw/questions/10198221)

關於共用方法的整理

(https://ithelp.ithome.com.tw/questions/10198213)

請問全台有哪些大學有在接壓力測試的業務

(https://ithelp.ithome.com.tw/questions/10198228)

用robocopy指令刪除特定天數的資料夾

(https://ithelp.ithome.com.tw/questions/10198277)

網路IP設定完馬上又顯示成自動取得

(https://ithelp.ithome.com.tw/questions/10198262)

多媒體電子看板IP連線設置

(https://ithelp.ithome.com.tw/questions/10198290)

問c++相關問題

(https://ithelp.ithome.com.tw/questions/10198270)

php
(<https://ithelp.ithome.com.tw/tags/questions/php>)

5
5
5
5
5

31
讚

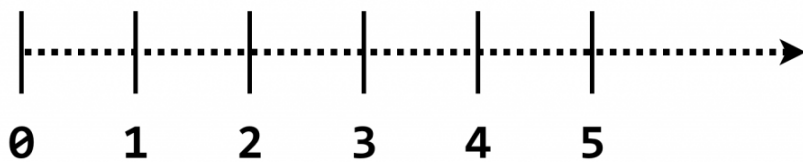


為什麼會這樣呢，先前曾經說過：

「切分變數有效範圍的最小單位是 "function" 」
「切分變數有效範圍的最小單位是 "function" 」
「切分變數有效範圍的最小單位是 "function" 」

很重要，所以要再講三次。

假設我們的時間軸是這樣的：



但我們知道，JavaScript 是一個「非同步」的語言，所以當我們執行這段程式時，for 迴圈並不會等待 window.setTimeout 結束後才繼續，而是在執行階段就一口氣跑完。

```
for( var i = 0; i < 5; i++ ) {  
  window.setTimeout(function() {  
    console.log(i);  
  }, 1000);  
}
```



(https://itadapi.ithome.com.tw/media/click?q=B3~ithome_forum~2004B311)

熱門文章

5 位大大齊聚 AMA 分享口罩查詢服務開發經驗，把握被增幅的好機會一起來聊聊吧！

(<https://ithelp.ithome.com.tw/articles/10231167>)

2020年您不容錯過的9種免費數據提取工具

(<https://ithelp.ithome.com.tw/articles/10231145>)

[自學Python紀錄] HackerRank 新手30天挑戰-Day03

(<https://ithelp.ithome.com.tw/articles/10231137>)

遠端工作躲疫情，但網路連線夠安全嗎？不可不知的VPN連線

(<https://ithelp.ithome.com.tw/articles/10231159>)

動態規劃經典題: 最大正方形面積

(<https://ithelp.ithome.com.tw/articles/10231142>)

[自學Python紀錄] HackerRank 新手30天挑戰-Day06

(<https://ithelp.ithome.com.tw/articles/10231195>)

IQ益智題- 外星人綁架事件之猜自己帽子

(<https://ithelp.ithome.com.tw/articles/10231188>)

Phoenix American Financial Services - Fund Accounting Services

(<https://ithelp.ithome.com.tw/articles/10231163>)

IQ益智題- 破解賭場老闆的帽子猜奇偶賭局

(<https://ithelp.ithome.com.tw/articles/10231220>)

[自學Python紀錄] HackerRank 新手30天挑戰-Day04

(<https://ithelp.ithome.com.tw/articles/10231157>)

一週點數排行

更多點數排行 ▶

(https://ithelp.ithome.com.tw/point_rank/total)

- 1 海綿寶寶 (antijava)
(<https://ithelp.ithome.com.tw/users/2000>)
- 2 阿展展展 (eric211924)
(<https://ithelp.ithome.com.tw/users/2011>)
- 3 浩瀚星空 (yoching)
(<https://ithelp.ithome.com.tw/users/2010>)
- 4 listennn08 (listennn08)
(<https://ithelp.ithome.com.tw/users/2011>)



實際上 for 迴圈在開始執行的一瞬間就把「五次的」
`window.setTimeout()` 跑完了



於是實際執行時，`console.log()`
會在「一秒鐘之後」同時印出「五次 5」

很明顯地，這段程式有兩個問題需要被解決：

1. 提出問題的人(不是)

1. `console.log` 印出來的數字
2. `console.log` 執行的時間

首先先看「印出來的數字」。

由於切分變數有效範圍的最小單位是 `function`，這代表著當我每一次執行 `setTimeout` 的時候，

```
window.setTimeout(function() {  
  console.log(i);  
}, 1000);
```

裡面 `console.log(i);` 的 `i` 變數是去函式「外層」拿的對吧？

(不知道為什麼的朋友趕快回去看 重新認識 JavaScript: Day 10 函式 Functions 的基本概念 (<https://ithelp.ithome.com.tw/articles/10191549>))

也就是說，因為 `for` 迴圈並不會等待 `window.setTimeout` 結束後才繼續，所以當 `window.setTimeout` 內的 `Callback Function` 執行時，拿到的 `i` 已經是跑完 `for()` 迴圈的 5。

實際上 for 迴圈在這時候就已經執行完畢
換句話說，i 在這時候就已經從 0 跑到 5 了

登入/註冊 (<https://ithelp.ithome.com.tw/users/login>)



window.setTimeout() 等了一秒後才執行，
這時候 i 的值已經是跑完 for 迴圈的「5」

那麼要怎麼解決這個問題呢？

利用「切分變數有效範圍的最小單位是 "function"」這個特性，我們可以把 window.setTimeout 透過一個「立即被呼叫的特殊函式」來包裝。

像這樣的寫法：

```
(function(){  
  // 做某事...  
  
})();
```

很多從其他語言轉來寫 JavaScript 的朋友可能不習慣。

沒關係，我們來拆解一下結構，你就會發現這樣的寫法其實沒有那麼難懂。

首先從一般單純的函式寫起：

```
function doSomething ( i ){  
  // 做某事...  
  
}
```

有個叫 doSomething 的函式，裡面有個參數 i，當我們要呼叫的時候，會用小括號 () 來呼叫它: doSomething(xxx)，大家應該都很熟悉了。

那麼，假設我們希望在定義函式的當下就呼叫它：

1. 先加個小括號 () 把這個函式包起來：

```
(function doSomething ( i ){  
  // 做某事...  
  
})
```

2. 因為要呼叫它，所以在後面再加個小括號 ()

```
(function doSomething ( i ){  
    // 做某事...  
  
})(123)
```

看出來了嗎？

一般的呼叫方式：

```
doSomething(123);
```

函式宣告當下即呼叫：

```
(function doSomething ( i ){  
    // 做某事...  
  
})(123);
```

假設這個函式我們只需要在宣告的當下執行，之後不會再呼叫，那麼甚至連名字都不用了：

```
(function ( i ){  
    // 做某事...  
  
})(123);
```

喔喔喔！這不就是我們一開始看到的那個「立即被呼叫的特殊函式」(**Immediately Invoked Function Expression, IIFE**) 嗎！

```
(function(){  
    // 做某事...  
  
})();
```

回到 `for` 迴圈與 `window.setTimeout` 的問題。

```
for( var i = 0; i < 5; i++ ) {  
    window.setTimeout(function() {  
        console.log(i);  
    }, 1000);  
}
```

我們知道像上面這樣的寫法，在執行 `window.setTimeout` 的時候，`i` 早已變成了 `5`，那麼為了保留每一次執行迴圈時那個「當下的」`i`，我們可以用一個立即被呼叫的特殊函式將它包覆起來，然後將 `i` 作為參數傳入：


```
for( var i = 0; i < 5; i++ ) {  
    // 為了凸顯差異，我們將傳入後的參數改名為 x  
    // 當然由於 scope 的不同，要繼續在內部沿用 i 也是可以的。  
    (function(x){  
        window.setTimeout(function() {  
            console.log(x);  
        }, 1000);  
    })(i);  
}
```

31
讚

這時候你會發現，執行的結果就會是我們預期的 0 1 2 3 4 了。

但還是在一秒鐘後同時出現啊？

嘿嘿，相信聰明的你已經發現，由於 for 迴圈在一瞬間就跑完，等於那一瞬間它向 window 依序註冊了五次 timer，每個 timer 都只等待一秒鐘，當然同時出現囉。
所以我們稍微修改一下：

```
for( var i = 0; i < 5; i++ ) {  
  
    (function(x){  
        // 將原本的 1000 改成 1000 * x  
        window.setTimeout(function() {  
            console.log(x);  
        }, 1000 * x);  
    })(i);  
  
}
```

像這樣，就可以依序印出我們要的結果囉！

上面我們用了一個經典的案例來跟各位介紹 IIFE 的用法，除了在迴圈內呼叫 function 我們會需要用 IIFE 來把參數的值保留起來之外[註1]，IIFE 還有另外幾個好處，就是可以減少「全域變數」的產生，同時也避免了變數名稱衝突的機會。

如果你有去看過 jQuery 的原始碼，就會發現 jQuery 也用了相同的手法將 window 與 undefined [註2] 保留起來：

```
(function( window, undefined ) {  
  
    // 略...  
  
})( window );
```

- [註1] ES6 以後新增了 let 與 const，且改以 { } 作為它的 Scope。
換句話說，將範例中的 for 改為 let 就可以做到保留 i 在執行迴圈當下的「值」的效果：

```
for( let i = 0; i < 5; i++ ) {  
    window.setTimeout(function() {  
        console.log(i);  
    }, 1000);  
}
```

• [註2] undefined 是可以被修改的 (詳見 [重新認識 JavaScript: Day 03 變數與資料型別](https://ithelp.ithome.com.tw/questions))
技術文章 (https://ithelp.ithome.com.tw/articles?tab=tech) 類別 (https://ithelp.ithome.com.tw/articles/10190873))，所以 jQuery 雖然在 IIFE 定義了兩個參數，但只傳了一個 winodw，就是為了保持 undefined 原本登入/註冊 (https://ithelp.ithome.com.tw/users/login) 樣子。

31 讚

留言 2 追蹤 (https://ithelp.ithome.com.tw/users/login)

檢舉 (https://ithelp.ithome.com.tw/users/login)

← 上一篇 重新認識 JavaScript: Day 17 函式裡的「參數」 (https://ithelp.ithome.com.tw/articles/10192368) 下一篇 → 重新認識 JavaScript: Day 19 閉包 Closure (https://ithelp.ithome.com.tw/articles/10193009)

系列文

重新認識 JavaScript (https://ithelp.ithome.com.tw/users/20065504/ironman/1259) 共 37 篇

目錄

 RSS系列文 (https://ithelp.ithome.com.tw/rss/series/1259)

 訂閱系列文 (https://ithelp.ithome.com.tw/users/login) 510 人訂閱

33 重新認識 JavaScript 番外篇 (3) - 鐵人賽戰況分析 (https://ithelp.ithome.com.tw/articles/10196538)

34 重新認識 JavaScript 番外篇 (4) - 台灣街景封面產生器 with Google map 街景 (https://ithelp.ithome.com.tw/articles/10196779)

35 重新認識 JavaScript 番外篇 (5) - 鐵人賽戰況分析 II (https://ithelp.ithome.com.tw/articles/10196900)

36 重新認識 JavaScript 番外篇 (6) - 網頁的生命週期 (https://ithelp.ithome.com.tw/articles/10197335)

37 重新認識 JavaScript 番外篇 (7) - 判斷式 (a == 1 && a == 2 && a == 3) 結果為 true ? (https://ithelp.ithome.com.tw/articles/10197606)

 完整目錄 (https://ithelp.ithome.com.tw/users/20065504/ironman/1259)

2 則留言

 0

**RocMark** (https://ithelp.ithome.com.tw/users/20104135/profile)
iT邦新手 5 級 · 2018-03-09 13:50:59
(https://ithelp.ithome.com.tw/users/20104135/profile)
感謝大佬，終於比較全面性的了解CallBack了

我對於IIFE的理解是可以防止變量及外部訪問

想問一下 let 解決了 var 汙染全域問題，是否也同時有 IIFE寫法帶來的優點

IIFE 還有需要被使用嗎?(在可以用ES6的環境下)
有需要能用在哪裡呢?

回應 2 檢舉 (https://ithelp.ithome.com.tw/users/login)

**Kuro Hsu** (https://ithelp.ithome.com.tw/users/20065504/profile)
iT邦新手 4 級 · 2018-03-09 15:22:28
(https://ithelp.ithome.com.tw/users/20065504/profile)
過去使用 IIFE 最主要的原因就是為了避免變數污染造成的問題。

若是 ES6+ 環境以 `let` 取代 `var`，再加上 `{ }` 將變數鎖定在 block-scope 之後，就可以完全取代 IIFE。

技術問答 (<https://ithelp.ithome.com.tw/questions>)

技術文章 (<https://ithelp.ithome.com.tw/articles?tab=tech>)

登入/註冊 (<https://ithelp.ithome.com.tw/users/login>)



RocMark (<https://ithelp.ithome.com.tw/users/20104135/profile>)

iT邦新手 5 級 · 2018-03-09 19:28:35 (<https://ithelp.ithome.com.tw/users/20104135/profile>)

了解了 感謝!

登入發表回應 (<https://ithelp.ithome.com.tw/users/login>)

31
讚



jack1234552000 (<https://ithelp.ithome.com.tw/users/20108315/profile>)

iT邦新手 5 級 · 2019-06-02 22:31:18

(<https://ithelp.ithome.com.tw/users/20108315/profile>)

請問為什麼 印出aaa

是迴圈跑完的i

```
var aaa = new Array();
var bbb = new Object();
for(let i =0;i<5;i++)
{
    bbb.x = i;
    aaa.push(bbb);
}
console.log(aaa);
```

1 回應

檢舉 (<https://ithelp.ithome.com.tw/users/login>)



Kuro Hsu (<https://ithelp.ithome.com.tw/users/20065504/profile>)

iT邦新手 4 級 · 2019-06-02 22:41:25 (<https://ithelp.ithome.com.tw/users/20065504/profile>)

because object is "call by reference".

ref: 重新認識 JavaScript: Day 05 JavaScript 是「傳值」或「傳址」?

(<https://ithelp.ithome.com.tw/articles/10191057>)

登入發表回應 (<https://ithelp.ithome.com.tw/users/login>)

我要留言

立即登入留言 (<https://ithelp.ithome.com.tw/users/login>)

在家上班不再只是選擇題，而是企業現在必備的能力。
從IT的技術、應用、維運，全面探討企業在未來的發展與趨勢

(https://itadapi.ithome.com.tw/media/click?q=B1~ithome_forum~2004B110)

電週文化事業版權所有、轉載必究 | Copyright © iThome (<https://www.ithome.com.tw/>) 刊登廣告 (<https://www.ithome.com.tw/aboutus/>)

授權服務 (<https://www.ithome.com.tw/aboutus/>) 服務信箱 (<mailto:ithelp@mail.ithome.com.tw>)

隱私權聲明與會員使用條款 (<https://www.ithome.com.tw/terms>) iT邦幫忙使用說明 (<https://ithelp.ithome.com.tw/guide>)