

# 程式人《十分鐘系列》



用十分鐘瞭解

《JavaScript 的程式世界》

陳鍾誠

2016 年 9 月 10 日

# 以年齡來看

- 1995 年被創造出來的 JavaScript  
實在有點老！

# 很巧的是

- JavaScript 和 Java 幾乎是同年齡！

但與其說這是個巧合

- 不如說是個《商業陰謀》！

在那個瀏覽器剛被世界認識的年代

# 全世界的軟體公司

- 都想進入瀏覽器裏分一杯羹

# 瀏覽器的先鋒 Netscape 公司

- 為了讓瀏覽器互動性更好
- 找了布蘭登 (Brendan Eich) 這位員工來發明一個為瀏覽器打造的程式語言

布蘭登·艾克



出生	1961年 (54–55歲)
	 美國匹茲堡賓夕法尼亞州
母校	伊利諾伊大學厄巴納-香檳分校
知名於	JavaScript
網站	<a href="http://brendaneich.com">brendaneich.com</a> 

# 他們將這個語言

- 命名為 LiveScript



# 但是在當時

- 為了對抗那個到處撒 IE 瀏覽器，甚至硬綁在 windows 內部出貨搶占地盤的微軟。
- Netscape 決定和昇陽 (Sun) 公司結盟

# 當時的昇陽公司

- 正在力推一個叫做 Java 的語言，也是主打《瀏覽器內部應用程式》。



# 只是 Java 採用的方法

- 是在瀏覽器裡外掛一個 JVM 虛擬機插件，然後讓 Java 在 JVM 內執行。
- 但是 LiveScript 則是內建在瀏覽器裏採直譯的方式執行的，所以相對速度會比較慢一點。

# Netscape 既然決定

- 要與昇陽聯盟對抗微軟
- 於是就展示了《誠意》，把  
LiveScript 語言改叫 JavaScript，感  
覺好像是《迷你 Script 版的 Java》一  
樣！

# 但是這樣一改

- 兩個原本沒有血緣關係的程式語言，突然變成了親家！

# 結果是讓大家誤會

- JavaScript 就是 Java 的陽春版
- Java 就是 JavaScript 的完整功能版

# 這個後遺症

- 一直延續到了現在
- 還有很多老一輩的程式人，認為 JavaScript 是 Java 的簡易版。
- JavaScript 和 Java 是同一系列的！

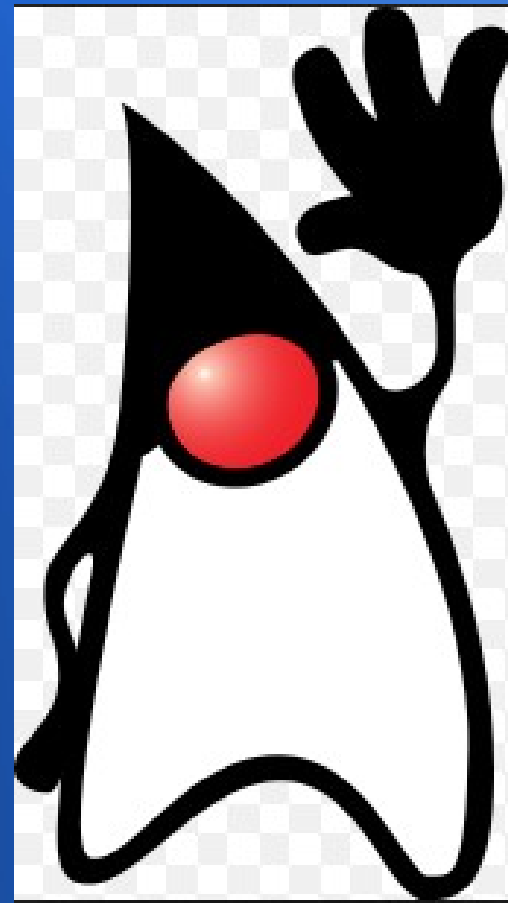
# 但是

- 這個一點都不美麗的誤會
- 還沒有畫下句點，另一個誤會又開始醞釀了！



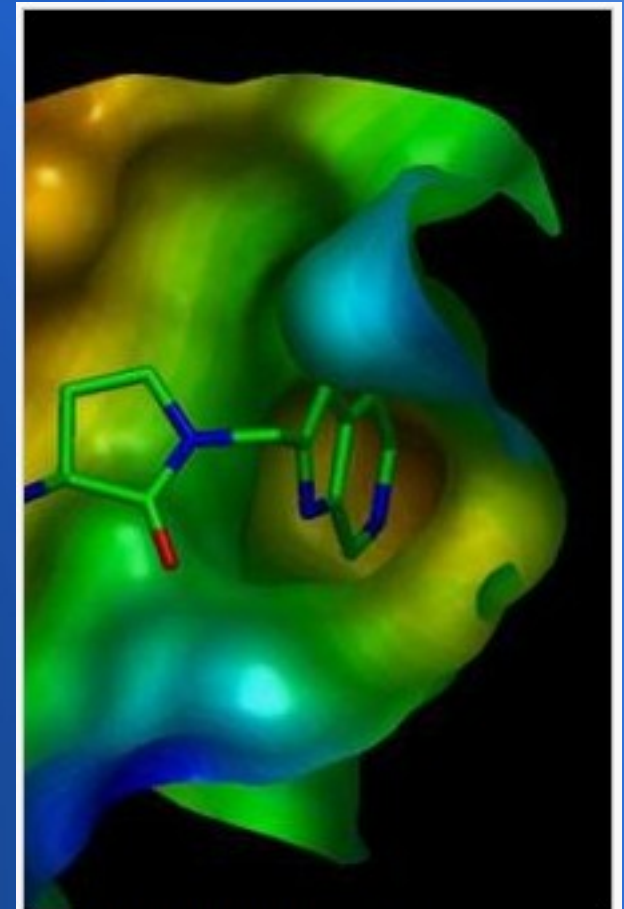
# 當初昇陽推 Java 時

- 主要的目標是成為《瀏覽器內應用》的霸主！
- 所以推出了一個稱為 Java Applet 的技術。



# 所謂的 Java Applet

- 就是可以嵌在瀏覽器內執行的 Java 程式，通常和動畫繪圖有關



A Java applet that uses 3D hardware acceleration to visualize 3D files in .pdb format downloaded from a server<sup>[1]</sup>

# 昇陽推銷 Java 的口號

- 是 Write Once , Run Anywhere

# 這對那些

- 怕被《微軟》綁住的大企業，很有吸引力！

# 也對想學網路的程式人

- 很有吸引力！

# 所以很多公司和程式人

- 都紛紛投入 Java 的懷抱

(我也是其中一位 XD)

# 奇妙的是

- 明明行銷很成功！
- 但是昇陽的技術部門，就是不懂得搞瀏覽器圖形程式。

# 結果

- 在 1997 年時 Macromedia 推出了一個 Flash 技術，把 Java 明明已經放到嘴邊的肉給搶走了！
- 後來 Flash 取代了 Java Applet 的地位，成了瀏覽器內繪圖程式的霸主！



# 然後

- Applet 就逐漸消失了
- Applet 成了一個很多人都知道，但是卻不能用的技術！

# 不過昇陽公司的 Java

- 卻沒有因此而消失！

# Java 失去了瀏覽器市場

- 但是卻在《企業市場》成了主流

# 這就是所謂的

- 《失之東隅、收之桑榆》 了吧！

# 昇陽甚至野心勃勃地

- 推出 Java 的微型版 J2ME ，想要進軍手機市場！

# 然後

- 歷史又開始出現弔詭！

# J2ME 主打傳統手機

- 但是當時全世界只有日本的 NTT  
DOCOMO 有辦法創造出成熟的手機行動商務市場！
- 其他國家的手機基本上只拿來打電話

# 日本人當然

- 會有自己的技術和方法！



# 日本的 FOMA 行動電話



# 非常的早熟且成功

- 這讓日本人在 2000 年時就能夠用手機上網購物與消費！



# 但是這個成功卻是導致

- 日本《弱電系產業全面衰敗》的種子！

# 日本的手機公司

- 從此眼界都只鎖在日本國內，  
所有技術也都是日本單行的規格

# 等到 2007 年

- iPhone+Android 出現之後
- 手機市場開始全面洗牌

# 日本手機開始全面潰敗

- 而且沒有跟上 Android 的浪潮

# 那時台灣的 HTC 宏達電

- 透過 2000 年 Compaq 的 iPAQ、微軟的 Windows Mobile，以及 2007 年的 Google Android，實現了三波的接力式跳躍成長，終於成了台灣股王！





# 但是、電腦的領域

- 幾乎沒有永遠的贏家！



# 日本的手機產業兵敗如山倒

- 台灣的 HTC 現況大家也都知道！
- 而那個搶下 Java Applet 市場的 Flash，卻在 2010 年賈伯斯的極力阻攔下，從手機市場敗陣下來，連帶影響到桌上型市場，現在也快消失了。

# 那麼、究竟是誰

- 取代了 Flash 的地位呢？

答案有兩個

# 一個是在手機市場

- 原生 App 打敗了 Flash，成為手機上的主力平台！

# 另一個是在瀏覽器內

- JavaScript 透過 HTML5 捲土重來，搶下了網頁內的繪圖市場，成為新一代 web 技術的主力！

# 於是

- 那個原本應該屬於 Java 的市場
- 竟然被他沒有血緣關係的小老弟  
JavaScript 給奪走了！

# 回想這段歷史

- 還是讓我覺得歷歷在目！

# 那是因為

- 我實在是夠老的了！
- 親身經歷了那些年的技術演變。



# 現在

- 讓我們把焦點移回主題！

# 也就是

- JavaScript 的程式世界這件事情！

程式人《十分鐘系列》

本文衍生自維基百科



用十分鐘瞭解

《JavaScript 的程式世界》

陳鍾誠

2016 年 9 月 10 日

# 經過前面的介紹

- 您應該知道，JavaScript 現在是瀏覽器內繪圖技術的霸主！

# 而且

- 基本上在 Applet 與 Flash 垮掉之後，JavaScript 是瀏覽器內唯一可用的技術了！

# 不過

- 雖然目前瀏覽器內 JavaScript 是主流，但是也不是沒有潛在敵人的！

而且、最可能的潛在敵人

- 又是 JavaScript 的表弟

# 那就是

## • Web Assembly ( 網路組合語言 )

### WebAssembly [編輯]

維基百科，自由的百科全書

**WebAssembly**或稱**wasm**是一個實驗性的低階程式語言，應用於瀏覽器內的客戶端。WebAssembly是可攜式的抽象語法樹<sup>[1]</sup>，被設計來提供比JavaScript更快速的編譯及執行<sup>[2]</sup>。WebAssembly將讓開發者能運用自己熟悉的程式語言（目前主要是C/C++）編譯，再藉JavaScript引擎在瀏覽器內執行<sup>[3]</sup>。WebAssembly的開發團隊分別來自Mozilla、Google、Microsoft、Apple，代表著四大網路瀏覽器Firefox、Chrome、Microsoft Edge、Safari<sup>[4]</sup>。

### 參考資料 [編輯]

- ↑ design/Rationale.md at master · WebAssembly/design [↗](#). GitHub.
- ↑ design/HighLevelGoals.md at master · WebAssembly/design [↗](#). GitHub.
- ↑ Google, Microsoft, Mozilla And Others Team Up To Launch WebAssembly, A New Binary Format For The Web [↗](#). TechCrunch. 2015-06-17.
- ↑ Bright, Peter. The Web is getting its bytecode: WebAssembly [↗](#). Ars Technica. Condé Nast. 2015-06-18.

### WebAssembly

<b>型態系統</b>	靜態
<b>系統平台</b>	跨平台
<b>許可證</b>	Apache授權條款
<b>常用副檔名</b>	.wast, .wasm
<b>網站</b>	<a href="http://webassembly.github.io">webassembly.github.io</a> <a href="#">↗</a>
<b>啟發語言</b>	
<a href="#">asm.js</a> 、 <a href="#">PNaCl</a>	

# Web Assembly 這個技術

- 一直讓我覺得，很像 Applet 與 Flash 虛擬機捲土重來的感覺！



# 如果 Web Assembly 成功

- JavaScript 雖然還是可以用來寫網頁繪圖程式。
- 但是將不會繼續保有壟斷性的地位！

# 不過 Web Assembly

- 還處於相當早期的階段

# 所以暫時

- 我們還是只能用 JavaScript 來寫
- 或者用其他語言，但最後都還是轉成 JavaScript 才放上瀏覽器！  
(像是 CoffeeScript 就是一個長得有點像 Python 的語言，最後會轉成 JavaScript)

# 現在、你應該明白

- JavaScript 最大的優勢在哪裡了！

# 接著

- 就讓我們介紹一下，JavaScript 的世界裡到底有些甚麼好了！

# 不過

- 在此我們並不是要介紹  
《程式語法》！

# 而是要介紹

- JavaScript 的社群、專案、工具、還有如何尋找資源的問題！

# 首先、讓我們認識一下

- JavaScript 除了在瀏覽器內，  
還有甚麼用途？



# 第一個用途是

- Server 上的伺服器程式
- 這個用途以 node.js 為主流！
- 目前常用的套件有 express 與 koa 。

# 以下是 node.js 的官網



<https://nodejs.org/en/>

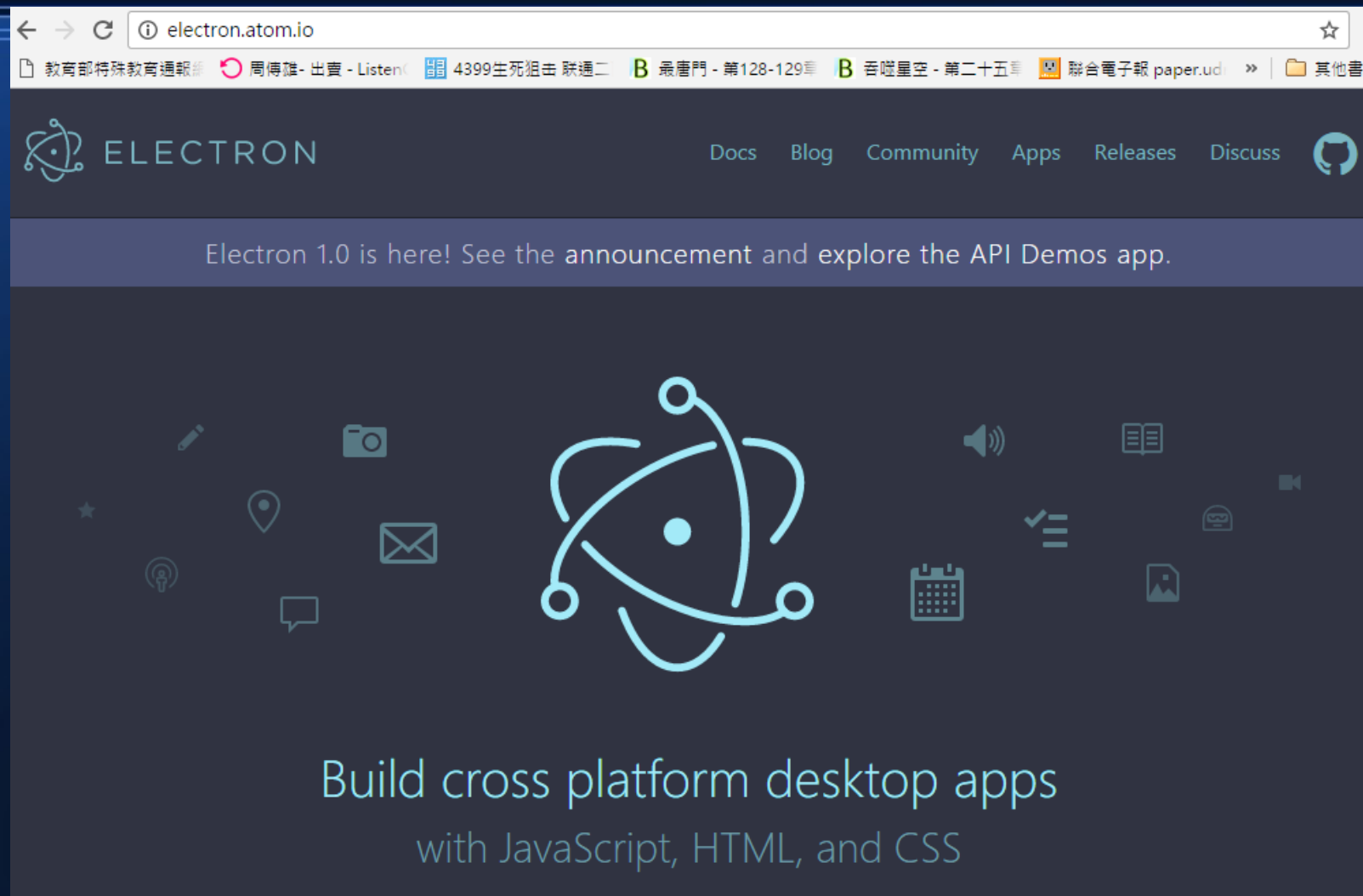
# 必須強調的是

- 雖然 node.js 常被用來撰寫伺服器端
- 但是拿來寫一般的命令列工具程式，也是非常方便的。（就好像你用 python 來寫命令列程式一樣）

## 第二個用途是

- JavaScript 可以用來寫桌上型的網頁式應用。
- 這個方面的解決方案目前有兩個主力，一個是 Intel 主導的 NW.js，另一個是 github 主導的 Electron

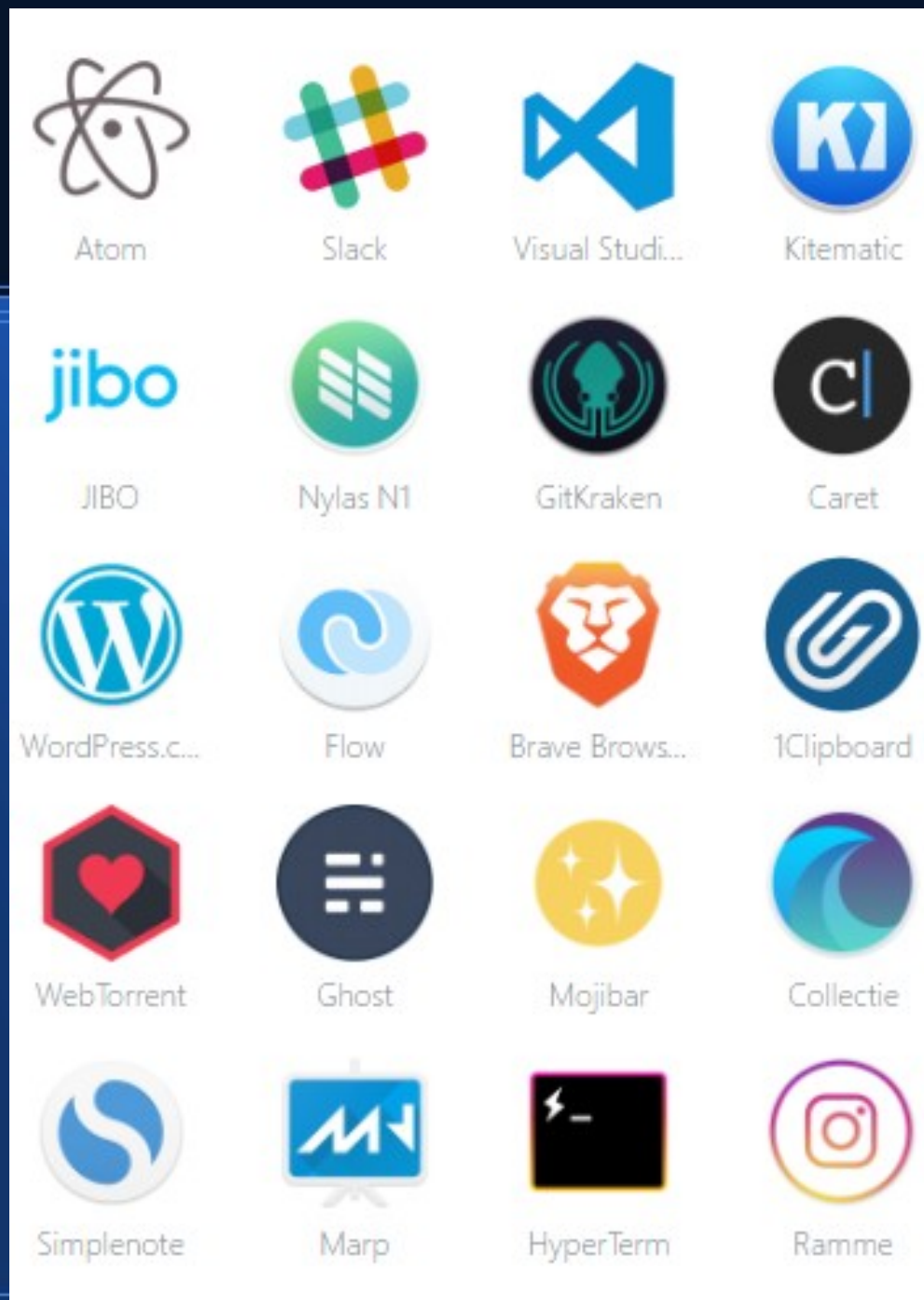
# 以下是 Electron 的官網



<http://electron.atom.io/>

# 還有右圖中是

- 用 Electron 所建立的桌面型瀏覽器應用！



# Electron 的程式碼長相如下

```
10  var initMenu = function () {
11      try {
12          var nativeMenuBar = new Menu();
13          if (process.platform == "darwin") {
14              nativeMenuBar.createMacBuiltin && nativeMenuBar.createMacBuiltin("FileExplorer");
15          }
16      } catch (error) {
17          console.error(error);
18          setTimeout(function () { throw error }, 1);
19      }
20  };
21
22  var aboutWindow = null;
23  var App = {
24      // show "about" window
25      about: function () {
26          var params = {toolbar: false, resizable: false, show: true, height: 150, width: 400};
27          aboutWindow = new BrowserWindow(params);
28          aboutWindow.loadURL('file://' + __dirname + '/about.html');
29      },
30  }
```

# 如果你想寫的不是瀏覽器行應用

- 而是傳統的經典視窗應用的話，那麼可以使用 Qt/QML 的技術。
- QML 是一種符合 ECMAScript 標準的 JavaScript 程式。



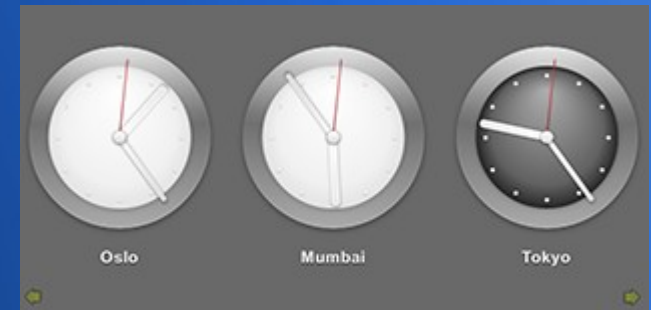
# QML 程式長相如下

```
import "content" as Content

Rectangle {
    id: root
    width: 640; height: 320
    color: "#646464"

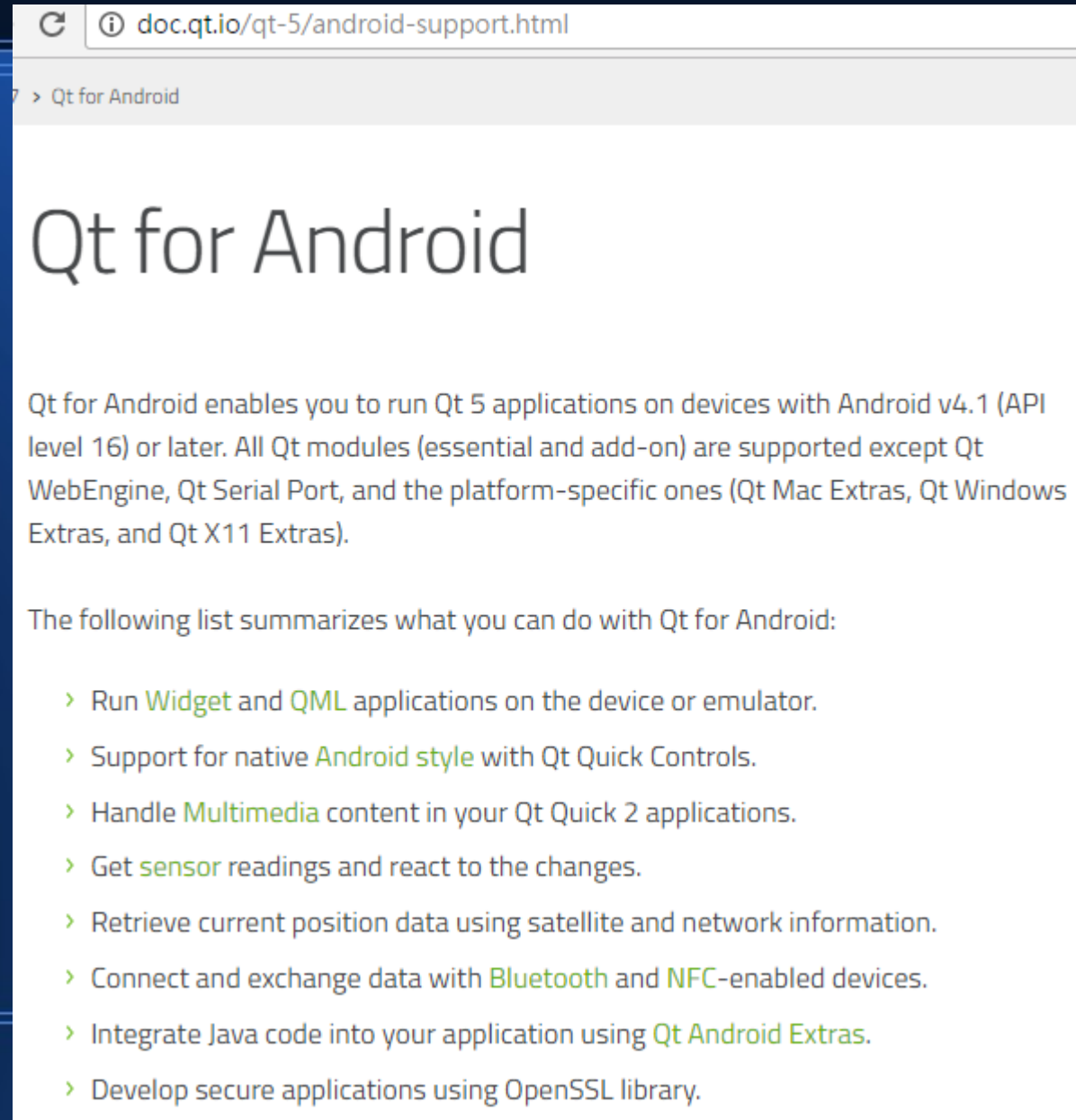
    ListView {
        id: clockview
        anchors.fill: parent
        orientation: ListView.Horizontal
        cacheBuffer: 2000
        snapMode: ListView.SnapOneItem
        highlightRangeMode: ListView.ApplyRange

        delegate: Content.Clock { city: cityName; shift: timeShift }
        model: ListModel {
            ListElement { cityName: "New York"; timeShift: -4 }
            ListElement { cityName: "London"; timeShift: 0 }
            ListElement { cityName: "Oslo"; timeShift: 1 }
            ListElement { cityName: "Mumbai"; timeShift: 5.5 }
            ListElement { cityName: "Tokyo"; timeShift: 9 }
            ListElement { cityName: "Brisbane"; timeShift: 10 }
            ListElement { cityName: "Los Angeles"; timeShift: -8 }
        }
    }
}
```



# 值得一提的是

- Qt/QML 有 Android 版，所以也可以用來寫手機 App。



# 最近好像又出了 iOS 版本

- 所以 Qt/QML 好像是想桌面手機全通吃的樣子！

## Qt Documentation

Google™ Custom Search

Qt 5.7 > Qt for iOS

### Qt for iOS

Qt's iOS port allows you to run Qt applications on iOS devices, such as iPhones and iPads.

### Getting Started

Development and deployment is done using Xcode. The supported workflow is to maintain a .pro file based project, which generates an Xcode project. Building and deploying can be done using either Xcode or Qt Creator. We will look at that in more detail in the next section.

The minimum deployment target for Qt applications is iOS 5.0. Qt then supports the following devices (iOS might need to be upgraded on some devices).

# 如果要開發手機板的瀏覽器應用

- 可以採用 PhoneGap/Cordova 方案！

## Adobe PhoneGap [編輯]

維基百科，自由的百科全書

**PhoneGap**是一款開放原始碼的行動裝置開發框架，旨在讓開發者使用HTML、Javascript、CSS等Web APIs開發跨平臺的行動裝置應用程式<sup>[3]</sup>。原本由Nitobi公司開發，現在由Adobe Systems擁有<sup>[4][5]</sup>。

### 目錄 [隱藏]

- 1 發展歷程
- 2 設計理念
- 3 支援平台
- 4 參見
- 5 注釋
- 6 外部連結

## 發展歷程 [編輯]

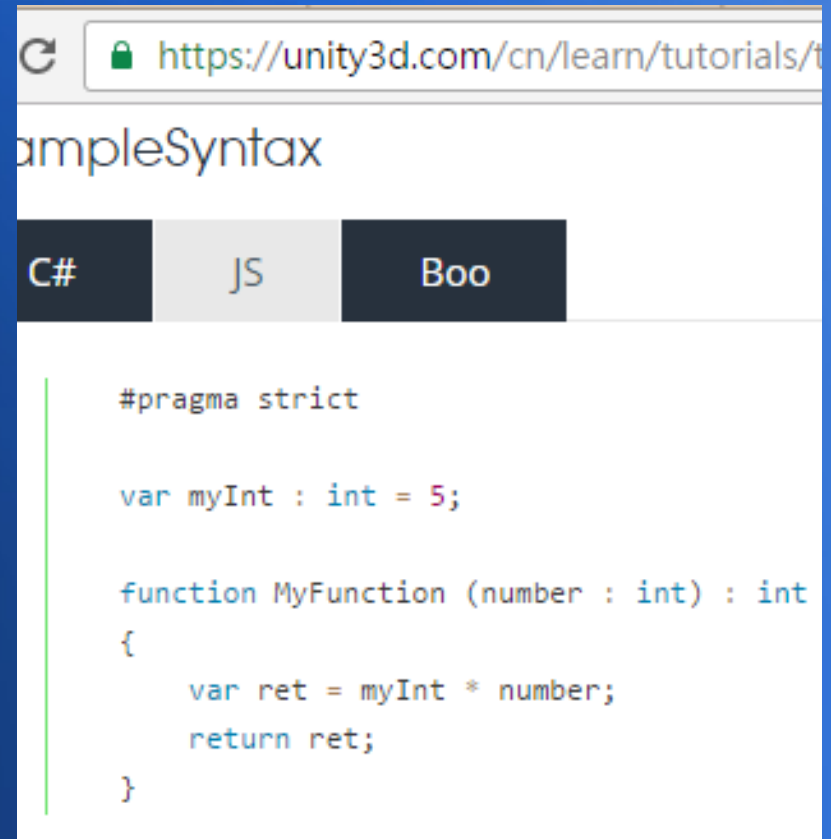
最早的PhoneGap起步於2009年在舊金山召開的iPhoneDevCamp大會<sup>[6]</sup>，Nitobi的工程師Brock Whitten、Rob Ellis和Andre Charland在iOS系統內架設

## PhoneGap(Cordova)

<b>原作者</b>	Joe Bowser、Michael Brooks、Rob Ellis、Dave Johnson、Anis Kadri、Brian Leroux、Jesse MacFadyen、Filip Maj、Eric Oesterle、Brock Whitten、Herman Wong、Shazron Abdullah
<b>開發者</b>	Adobe Systems
<b>穩定版本</b>	3.5.0（2014年5月23日）
<b>開發狀態</b>	開發中
<b>程式語言</b>	JavaScript、HTML5、CSS3、Java、C++、C#、Objective-C
<b>作業系統</b>	iOS、Android、webOS、Symbian、BlackBerry、Ubuntu Touch、Windows Phone、Windows 8、Firefox OS
<b>語言</b>	英語
<b>類型</b>	行動裝置開發框架
<b>授權條款</b>	Apache 2.0 License <sup>[1][2]</sup>
<b>網站</b>	<a href="http://www.phonegap.com">www.phonegap.com</a> <a href="http://cordova.apache.org">cordova.apache.org</a>

# 如果要開發手機遊戲

- 可以採用 Unity 遊戲引擎搭配 Unity Script (一種加上型態標示的 JavaScript) 來寫程式

A screenshot of a web browser displaying the Unity3D website. The address bar shows the URL <https://unity3d.com/cn/learn/tutorials/t>. The page title is "Sample Syntax". Below the title, there are three tabs: "C#", "JS", and "Boo". The "JS" tab is selected. The code editor shows the following JavaScript code:

```
#pragma strict

var myInt : int = 5;

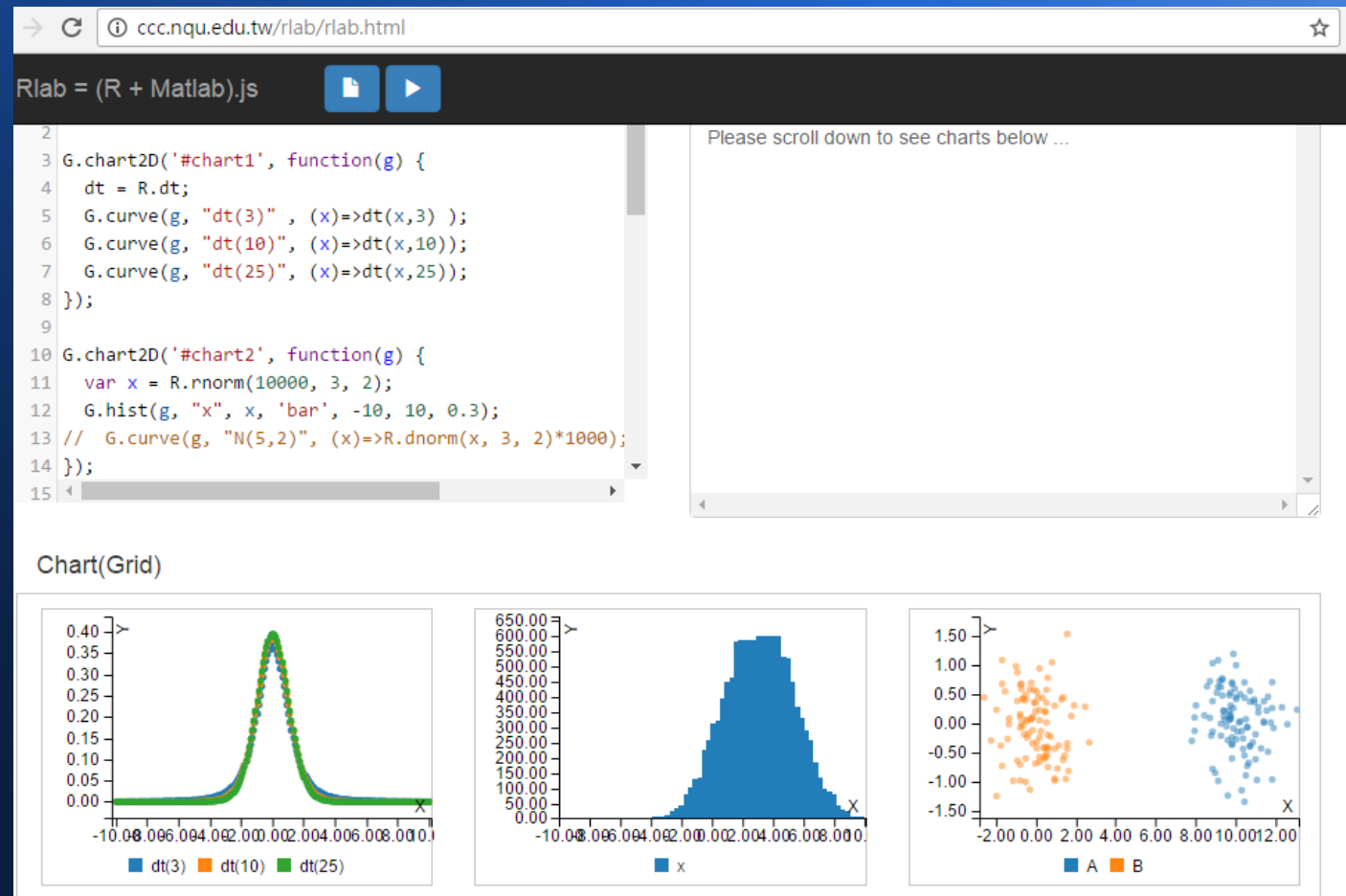
function MyFunction (number : int) : int
{
    var ret = myInt * number;
    return ret;
}
```

# 當然

- 我沒有那麼多時間，可以玩那麼多平台！
- 以上我其實只有玩瀏覽器端，還有 node.js( 伺服器端 + 命令列 ) 而已！

# 最近因為寫 rlab 科學計算平台

- 所以有在考慮要用 electron 來做桌面版。





# 另一個我開發的 wikidown 專案

- 主要是自己用來架設網站，寫網誌的專案！

<https://github.com/ccckmit/wikidown>

A screenshot of a web browser displaying the GitHub repository page for 'wikidown' by 'ccckmit'. The browser's address bar shows the URL 'https://github.com/ccckmit/wikidown'. The page content includes the title 'Wikidown = Wiki + Markdown', an 'Introduction' section stating 'Wikidown is a Wiki System for Markdown.', and an 'Install' section with instructions to install 'node.js+git' first. Below the install instructions is a code block showing the command to clone the repository. The 'Run' section shows terminal commands to run the server and the output of those commands, including the server URL and a database connection error.

GitHub, Inc. [US] | <https://github.com/ccckmit/wikidown>

## Wikidown = Wiki + Markdown

### Introduction

Wikidown is a Wiki System for Markdown.

### Install

You should install 《node.js+git》 first.

```
$ git clone https://github.com/ccckmit/wikidown.git
```

### Run

```
$ cd wikidown
$ node server
```

Server started: <http://localhost:80>  
Ssl Server started: <https://localhost:443>  
db connect fail



# 我的網站就是用 wikidown 架的



Browser address bar: [ccc.nqu.edu.tw/wd.html#main/home.wd](http://ccc.nqu.edu.tw/wd.html#main/home.wd)

Header: ※ 陳鍾誠

主題	連結
陳鍾誠	<a href="#">課程地圖</a>   <a href="#">教科書</a>   <a href="#">程式</a>   <a href="#">知識地圖</a>   <a href="#">十分鐘系列</a>
著作	<a href="#">UDN專欄</a>   <a href="#">Wikidot網誌</a>   <a href="#">小說</a>   <a href="#">散文</a>   <a href="#">詩</a>   <a href="#">秘笈</a>
雜誌	<a href="#">程式人雜誌</a>   <a href="#">少年科技人雜誌</a>
課程	<a href="#">程式</a>   <a href="#">劍宗</a>   <a href="#">氣宗</a>   <a href="#">數學</a>   <a href="#">中文</a>   <a href="#">跨界</a>
蒐錄	<a href="#">人物</a>   <a href="#">影片</a>   <a href="#">工具</a>   <a href="#">書籍</a>   <a href="#">圖靈獎</a>   <a href="#">考題</a>   <a href="#">程式</a>

本文部份內容與大部份圖片修改自 [維基百科](#)，使用時請遵守 [姓名標示](#)、[相同方式分享](#) 授權。  
編輯：陳鍾誠 email: [ccckmit@gmail.com](mailto:ccckmit@gmail.com)

# 因為自己寫的軟體

- 當然自己一定要去用的阿！

# 否則學了很多技術

- 最後的命運就是，很快就會過時了！

那些年  
我們還沒學會就已經過時的那些技術

陳鍾誠

2015 年 8 月 31 日

# 在做上述兩個專案時

- 我使用到的套件有：
- 後端：以 koa 為主！
- 前端：以 jQuery 和 bootstrap 為主！

# 我並沒有去學那些

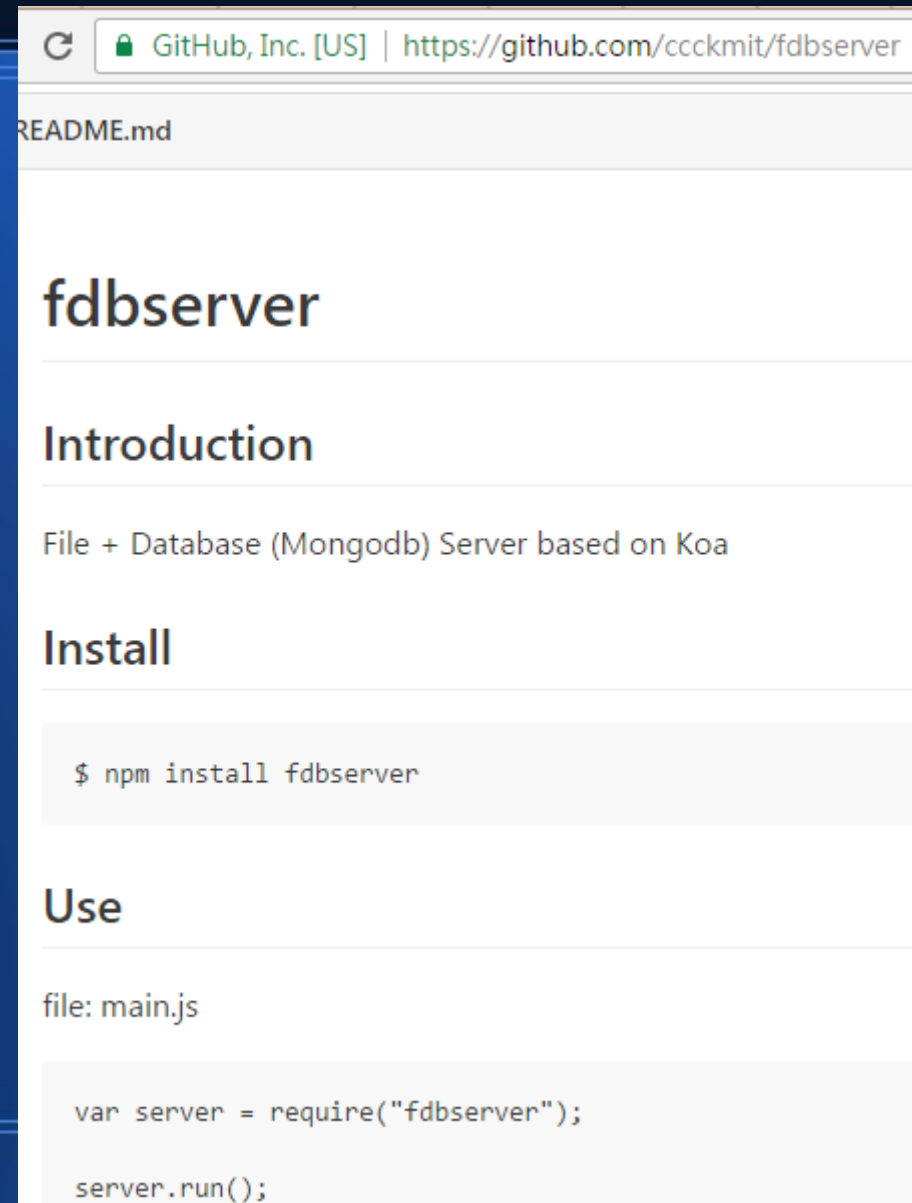
- 比較新或強大的前端，像是 Angular.js, React.js, Vue.js 等等。
- 因為我的應用通常不會有很複雜的介面，所以目前還不覺得有使用上面這種強大前端的需求。

# 在儲存技術上

- 我目前主要用檔案系統儲存
- 必要時會使用 MongoDB 儲存在文件資料庫。

# 因此我寫了 fdbserver

- 一個 Server 端的專案
- 用來處理後端的《檔案與資料庫儲存動作》



# 我通常用 AJAX 的方式

- 透過前端的《單頁應用》(Single Page Application)，將需要儲存的資料送給 fdbserver 去儲存。



# 這種模式

- 目前對我還算夠用！

# 在開發 wikidown 時

- 我去尋找了幾個套件，像是：

Showdown: 將 markdown 轉為 HTML

MathJax: 呈現 latex 數學式

Highlight: 為程式碼加上顏色！

# 然後最近開發 **rlab** 科學計算平台時

- 我用了下列的套件：

lodash.js: 基本常用函式庫

numeric.js: 線性代數矩陣函式庫

jstat.js: 機率統計函式庫

algebrite: 符號運算函式庫

# 另外還用了兩個前端繪圖套件

- C3.js (2D 繪圖)

- 這個用到 d3.js

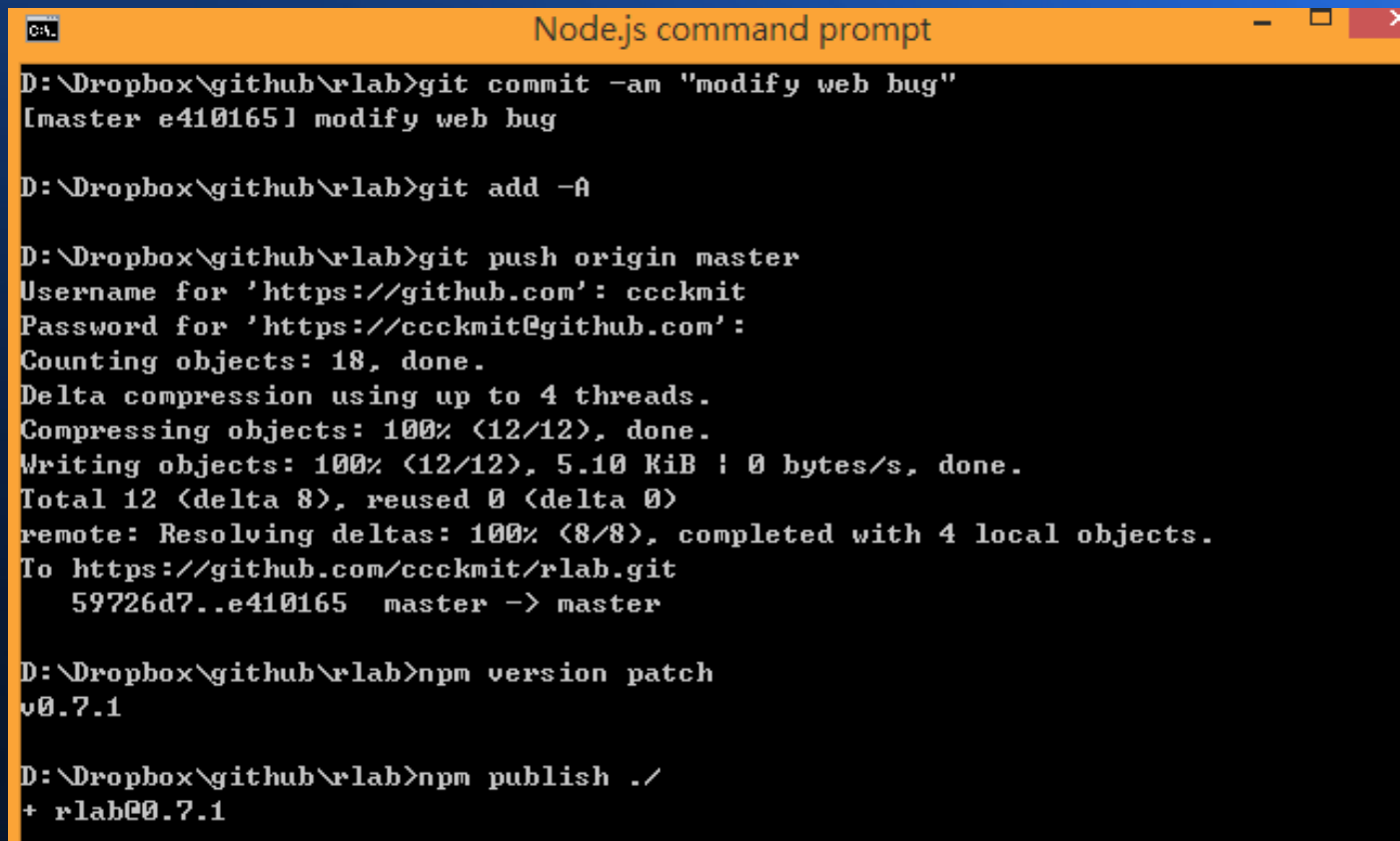
- Vis.js (3D 繪圖)

# 還有用到一個

- 讓程式編輯框顯示行號與顏色的  
CodeMirror.js 套件！

# 而且我都有上傳到

- github 與 npm 的開源專案庫中



```
Node.js command prompt
D:\Dropbox\github\rlab>git commit -am "modify web bug"
[master e410165] modify web bug

D:\Dropbox\github\rlab>git add -A

D:\Dropbox\github\rlab>git push origin master
Username for 'https://github.com': ccckmit
Password for 'https://ccckmit@github.com':
Counting objects: 18, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (12/12), done.
Writing objects: 100% (12/12), 5.10 KiB | 0 bytes/s, done.
Total 12 (delta 8), reused 0 (delta 0)
remote: Resolving deltas: 100% (8/8), completed with 4 local objects.
To https://github.com/ccckmit/rlab.git
  59726d7..e410165  master -> master

D:\Dropbox\github\rlab>npm version patch
v0.7.1

D:\Dropbox\github\rlab>npm publish ./
+ rlab@0.7.1
```

# 關於 npm 模組的使用上傳方法

- 可以參考另一篇十分鐘系列！



# 在做這些專案的過程

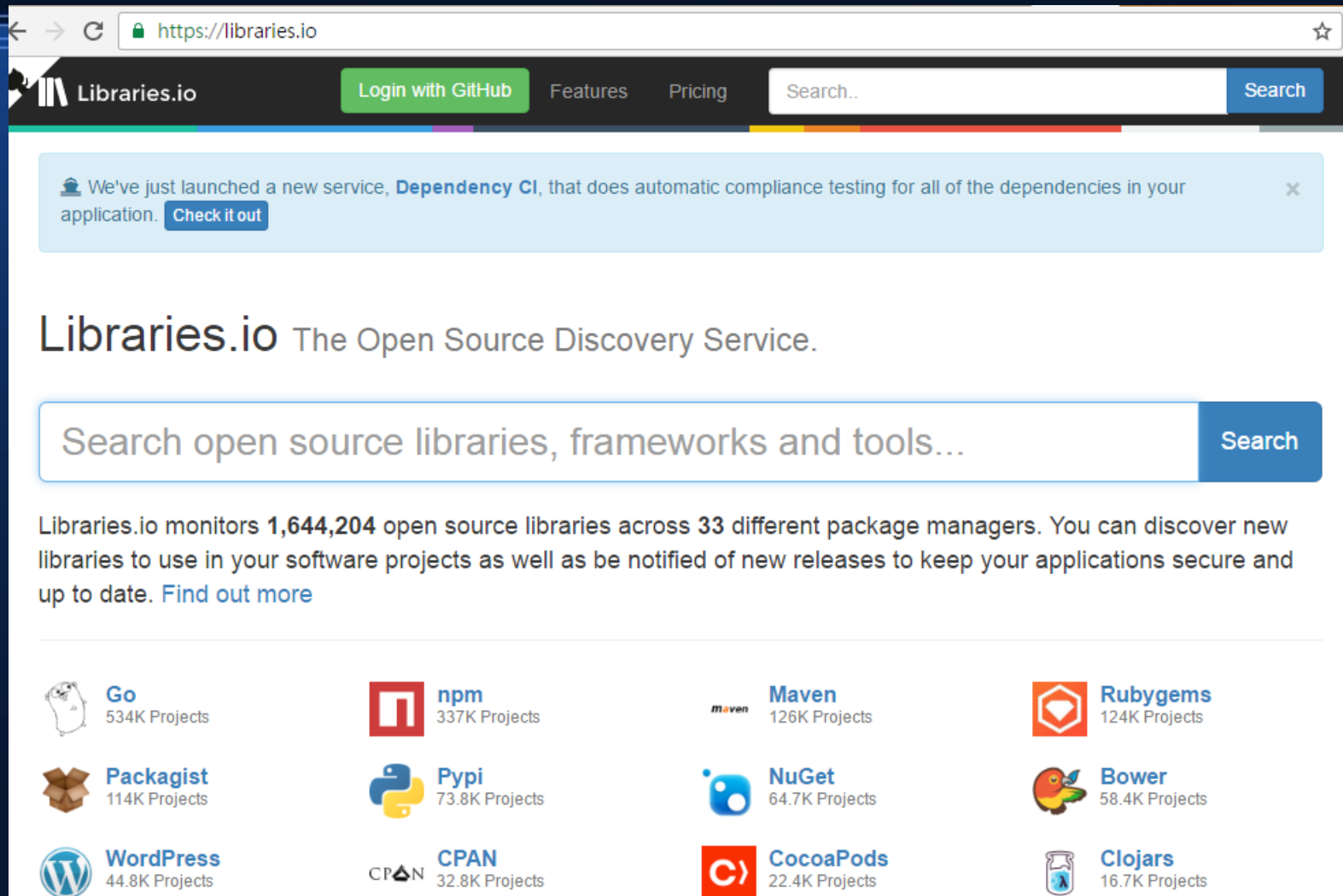
- 大約 30% 的時間是在找合適的專案
- 剩下的 70% 時間才是寫程式。



# 現在如果我有空

- 就會上 google, libraries.io, npm 等專案搜尋引擎上，看看有沒有好用的套件！

# libraries.io 是專門蒐尋開源專案的



# 而 google 可以搜尋到

- 不只是專案，還有別人的推薦或討論文章，像是 stackoverflow 上對專案的評價，就很有參考價值！

# 另外

- npm 和 github 也都可以找到很多專案！

# 有時候

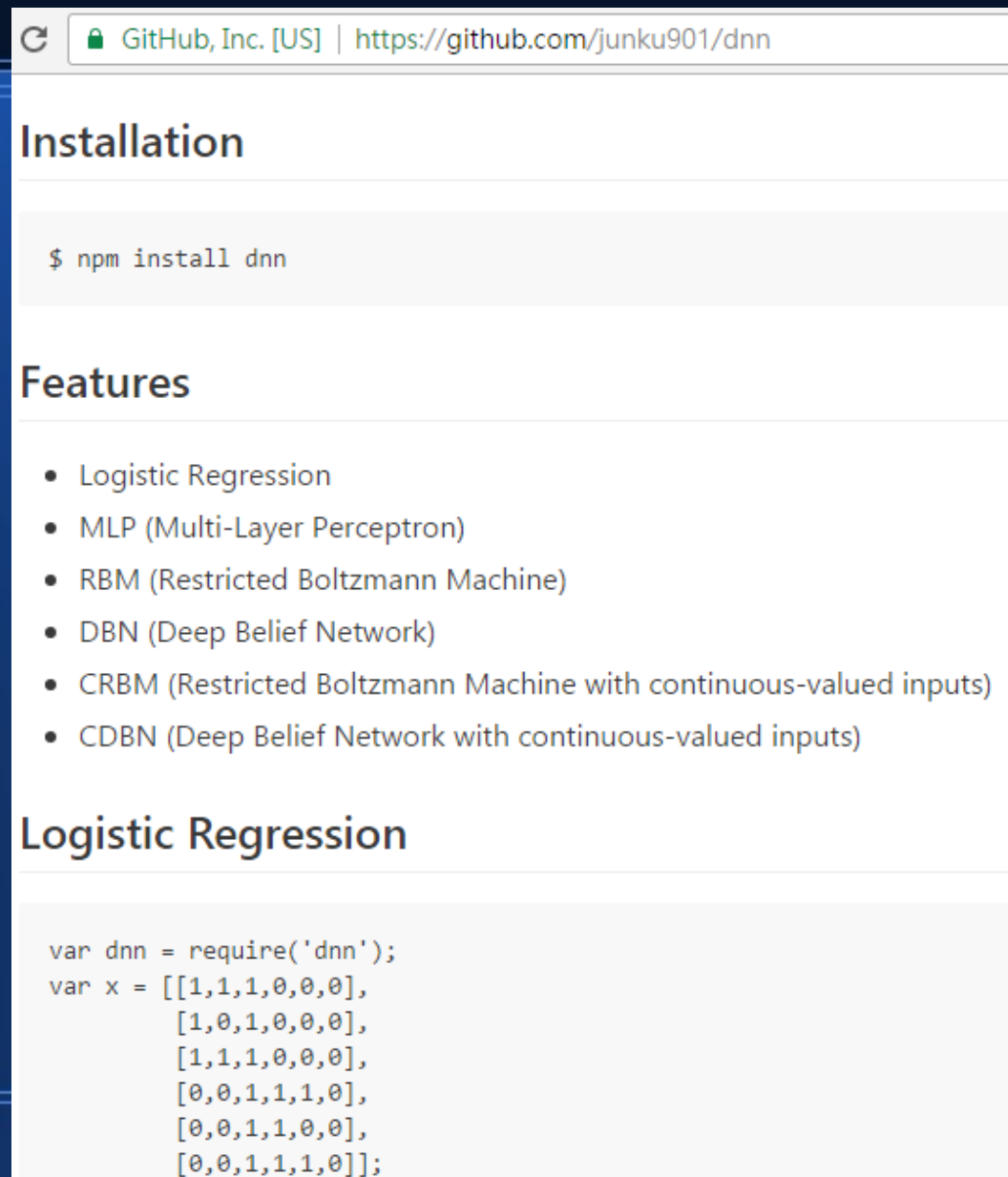
- 甚至當我想了解一些，比較學術性的議題時，我也會去搜尋開源專案。

# 像是要學習

- 人工智慧、機器學習、科學計算的方法時
- 我也會從《開源專案》去理解那些方法。

# 例如我想瞭解深度學習技術

- 於是找到一個稱為 dnn.js 的 javascript 專案
- 安裝執行並閱讀程式碼後，讓我對深度學習技術比較有實際的感受！



# 另外像是

- 決策樹、K-Mean、K 近鄰法、支持性向量機 SVM，分類分群等技術，以及人工智慧方法等等，也通常都可以找到簡單易懂的專案。



# 以上這些

- 就是我用 JavaScript 程式世界的資源所做的事情！

# 這些 javascript 的資源

- 對我有很大的幫助！

# 或許也會對你

- 有所幫助也說不定！

# 這就是今天的

- 十分鐘系列！

希望你會喜歡

我們下回見！

Bye Bye!