JavaScript 不是我的菜，但我愛（上）

JavaScript 從製作動態網頁效果的輔助角色，已經扶搖直上成為程式語言發展史中閃亮耀眼的一顆星，且持續還在發光發熱。學習 JavaScript 不僅是趨勢，對許多程式設計師而言也是一項重要技能。

發展迄今十多年的 JavaScript 語言，已在不同領域出現合適的應用：

- 以 JavaScript 開發跨平台的 Mobile App（例如 Titanium 與 PhoneGap 等）。
- 開發 Modern Web Application，不僅前端用 JavaScript 語言，後端應用程式亦可搭配 Node.js 撰寫。

儘管 JavaScript 有萬般美好，但它不是我的菜。身為經年累月的 Java 開發者，有太多專案、課程都無法輕易跟 Java 說 Good-bye，既然明著說分手不容易，那就只好跟 JavaScript 暗通款曲啦！

JavaScript 對教學的助益

目前 Java 仍是多數學校課程教學的主流程式語言，許多初學者第一次接觸的程式語言，就是 C 或 Java，近年 Android 課程需求大增，有更多學校急忙鼓勵學生速學 Java 語言；但如果沒有良好的學習指引和規劃，學生很可能只學到程式語言的語法，而不是程式設計的基本觀念與邏輯。

約耳談軟體在 2005 年曾發表過一篇《The Perils of JavaSchools（爪哇學校的危害）》，就談到不少 Java 不適合作為「First Programming Language」的問題，如果 Java 不適合初學者，那有什麼更好的語言適合教學嗎？Joel Spolsky 的建議是 Scheme --- 主要用於人工智慧領域的古老程式語言，Scheme 目前仍在 MIT 等多所名校 Computer Science 系所的必修課程用於教學。

我們從 2006 年開始發展第一套 PLWeb 線上教學系統，將 Scheme 語言用於資管必修的程式語言課程。觀察此教學課程帶來的效益，可以發現更多學生在程式設計能力的任督二脈被打通後，日後學習其他程式設計相關課程，進度明顯加快不少。但是在國內推廣 Scheme 課程並不容易，它不容易被多數學校的課程規劃納入，同時對多數業者來說也比較陌生，尚未接觸的學習者不易瞭解 Scheme 帶來的效益。

很幸運地這幾年 JavaScript 的學習蔚為風潮，因為被廣泛使用，將 JavaScript 用於教學課程，將有更多教師與學習者可以接受，從國外教學網站 CodeCademy 的成功案例就可見一斑。對於程式語言基礎的教學而言，JavaScript 具備一些與 Scheme 類似的條件，只要學會很少的幾個重要關鍵字，就可以開始寫程式，因此教學上更容易將焦點放在觀念與邏輯。舉例來說，在傳統的 Java 課程中，我們很難教會學生 Lambda 與 Closure 的概念，在 Scheme 教材中的 High-order Procedure 範例，也相當不容易轉換成 Java 版本；但是 JavaScript 則容易許多。

拜 JavaScript 的快速發展所賜，以 JavaScript 撰寫的簡易程式碼範例，可以更容易在各種行動裝置上被執行；我們甚至能以 BiwaScheme（Scheme interpreter written in JavaScript）在瀏覽器中執行 Scheme R6RS 的範例程式。如果再加上 JavaScript 開發的 Terminal 程式，更能夠將遠端執行編譯測試的程式，透過 tty 在瀏覽器中與操作者互動。

近年我們發展 CodeCanaan 教學系統，就是以 JavaScript 為基礎所建置，同時也極力發展將 JavaScript 用於程式設計課程的基礎教材。我相信這能帶給學習者更多幫助，因為 JavaScript 同時也是實用的語言，透過化繁為簡的教學範例，學習者獲得的不僅有程式設計的基礎能力，過程中也培養未來使用 JavaScript 從事實務開發的基本功夫。

當然 JavaScript as a first programming language 還有許多問題待解決，也絕對不是一蹴可幾，但許多人已經開始努力實現這個目標。若你對目前的教學發展感興趣，請別錯過 Khan Academy 的免費線上教材（http://goo.gl/b5dmH），一個結合互動式程式碼撰寫、圖形動畫展現與影音導覽的先進教材，而 Khan Academy 正採用 JavaScript 作為學習者的第一個程式語言（http://goo.gl/Z1K4O）；John Resig（creator of jQuery）的《Redefining the Introduction to Computer Science（http://goo.gl/l7p0l)》文章，對這項教學方法有更多說明。

當 JavaScript 碰上 Java

據說最初 JavaScript 是受 Java 啟發而設計的，但除此之外，Java 與 JavaScript 可是完全不同的兩個程式語言。

但發展迄今，有趣的事情開始發生了！使用 JavaScript 實作 JVM（Java 虛擬機器），甚至還有 CoffeeScript 的版本！

- BicaVM https://github.com/nurv/BicaVM
- Doppio https://github.com/int3/doppio

面對眾多的新語言，與其談論 A 和 B 哪個好，不如花些時間研究如何各取其優點，讓不同語言的優點併出火花。不管黑貓白貓，能抓到老鼠就是好貓；既然如此，何不一次養兩隻貓？

對於跨平台、跨程式語言的軟體架構，不只能用 Web Services 實現，利用 Message Queuing 機制實現 RPC（Remote Procedure Call）也是相當可行的作法（例如 RabbitMQ 及 ZeroMQ）。

自從 Node.js 賦予 JavaScript 在更多領域的應用，原本使用 Java EE 開發 Modern Web Application 所遇到的難題，也將有更多可行的方法迎刃而解。

事實上 JavaScript 陣營所發展的 Node.js 已經帶給開發者不少啟發，如今我們也希望在 Java 世界同樣擁有 Non-blocking、Asynchronous、Event-driven 的架構設計，例如使用 vert.x（http://vertx.io/）設計類似 Node.js 的 Java 程式。未來 Tomcat 7 或 Jetty 也將會有更多不同過去思維的新特性，如果先瞭解 JavaScript 與 Node.js，勢必也更能掌握 Java 世界的新脈動。事實上，不管使用什麼語言做開發，這個世界都是朝向同樣的方向在前進。

如果你同時喜歡 JavaScript 與 Java 兩種語言，像是 node-java（https://github.com/nearinfinity/node-java）這樣的專案，也許會讓你感到興趣，以下是一個在 Node.js 中搭配 Java 函式庫使用的範例。

```
var java = require("java");
java.classpath.push("commons-lang3-3.1.jar");

java.newInstance("java.util.ArrayList", function(err, list) {
  list.addSync("item1");
  list.addSync("item2");
});
```

混搭不同的程式語言，讓辛苦打造應用程式的開發者，更能將心力專注在重要的功能實作，而不必重新發明輪子。舉例來說，Java 實作的 JasperReport 報表引擎，

或是 Python 實作的 Docutils 與 Pygments 等，無論你有多愛 JavaScript 都不必重新打造這些東西。

待續…

@作者 lyhcode 目前從事程式設計教學與顧問工作。（http://lyhcode.info/）