

程式人



用十分鐘瞭解

陳鍾誠的程式設計課

(採用 JavaScript + C 的原因)

陳鍾誠

2016 年 2 月 22 日

2015 年開始

- 我又回來教大一的程式設計

上一次教大一程式

- 好像已經是八年前的事了！

那時

- 我是用 Java 在教！

但是這次

- 我上學期用的是 JavaScript
- 下學期會加入 C 語言

換言之

- 就是以 JavaScript + C 的
搭配來教

但是為何

- 我要選擇這種組合呢？

主要考量

- 我想是以實用性為主吧！

因為

- JavaScript 和 C 的搭配
- 可以同時涵蓋
 - 《系統、應用與網站》

現在

- 就讓我們來看看

這種組合的優缺點為何！

先講缺點好了

缺點就是

- JavaScript 和 C 都是有點奇怪的語言，有不少語法陷阱
- 並不是最適合初學者的！

最適合初學者的語言

- 我想應該是 Python

由於有些語法陷阱

- 所以使用的時候要特別小心
- 以免誤觸陷阱，讓自己身心受創！

但是

- 有一好必有一壞
- 有一失必有一得

程式語言

- 沒有最好的
- 每種語言都有其優缺點！

而且

- 重要的往往不是語言本身！
- 而是使用《語言》的人所形成的社群！以及相關的支援工具。

JavaScript 和 C

- 這兩個語言，都有龐大的支援工具和社群，而且應用極廣！

讓我們先來瞭解一下

- JavaScript 和 C 這兩個語言發明的背景。

C 語言

- 是 1970 年時，《丹尼斯·里奇》與《肯·湯普遜》兩人（簡稱 K&R），在開發 UNIX 作業系統的期間設計出來的。
- 由於是設計來開發作業系統的，所以對硬體的 control 能力很強，效能也很好。

後來

- C 語言成了《作業系統、系統程式、編譯器、嵌入式系統》等領域的主力語言。
- 像是 UNIX, Linux, Windows, MS.Office, Visual Studio, gcc, FreeBSD, JVM, node.js/V8, QEMU, ... 等等重量級的軟體，都是用 C 語言開發出來的。

另外還有

- 你所看到的所有語言，像是 Java, C#, Python, Ruby, JavaScript 等語言的開發工具和執行環境，也都是用 C 語言開發出來的。

但是

- C 語言沒有物件導向功能
- 所以後來衍生出 C++ 與 Objective C
 - 兩者都是 C 語言的物件導向版本

後來也有很多軟體

- 採用 C/C++ 的搭配組合來開發！
- 像是 gcc 編譯器的競爭對手 LLVM 就是以 C++ 為主的。
- 甚至連嵌入式的 Arduino 也是將 C++ 進一步包裝後的結果

C 語言的特色

- 是靜態編譯式，所以程式會被轉換成《CPU 的原始機器碼》，因此執行速度很快。
- 而且其特有的《指標》，可以透過記憶體映射的方式，直接控制硬體。

這都是因為

- 當年 K&R 兩人就是為了寫 UNIX 而設計 C 語言的原因
- 而這也是為何 C 語言在《系統層》特別強大的理由。

好了

- 介紹完 C 語言，讓我們將焦點轉向 JavaScript

JavaScript 出現在 1995 年

- 那是一個《網路與瀏覽器》
萌芽後大爆發的年代！

全世界的人

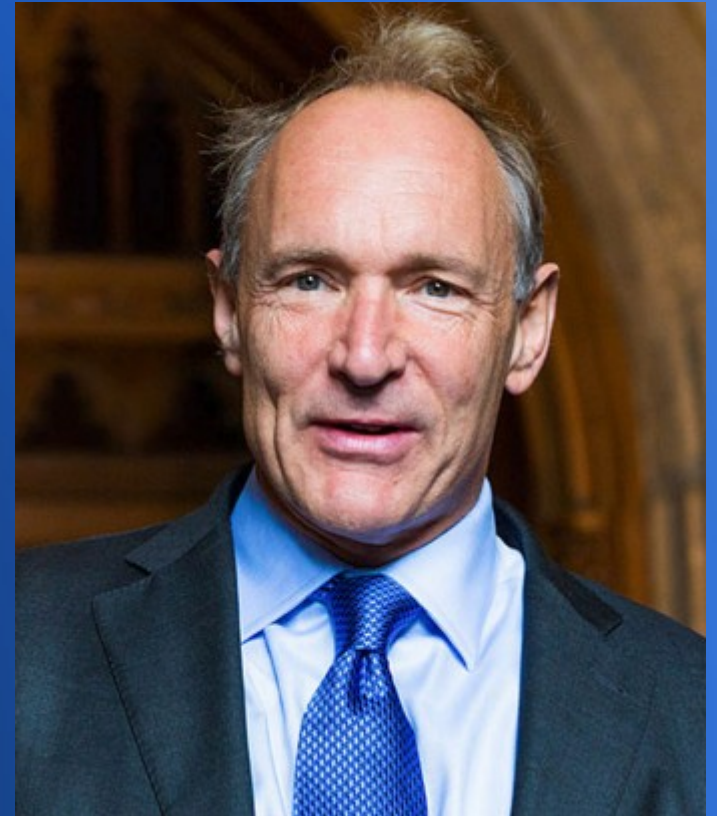
- 都想盡辦法讓他的電腦，能夠連上一種稱為 World Wide Web (www) 的網路。

而 WWW 的出現

- 主要歸功於兩組人馬
- 首先是發明者 Tim Berners-Lee
接著是瀏覽器廠商 Netscape

Tim Berners-Lee

- 當時以契約的方式為 CERN 工作
- 在 1980 年代後期利用《超連結與網路》試圖讓物理學家們可以方便的使用網路。
- 結果在 1990 年釋出了以 url, hyperlink, http 為基礎的 web server 和 client (瀏覽器).



於是後來

- Tim Berners-Lee 被尊稱為 Web 之父。
- 關於他發明 web 的故事與想法，都記載在《一千零一網》（Weaving the web）這本書裏頭。

但是

- Tim Berners-Lee 並沒有把瀏覽器做得很好用，伺服器也不夠強大。
- 後來 Apache 做出了很好的伺服器
- 而在美國超級電腦中心 NCSA 工作的《馬克·安德森》和《埃里克·比納》等人開發出了很好的 mosaic 瀏覽器。

由於 Mosaic 實在太紅了

- 所以《馬克·安德森》乾脆出來成立了 Netscape 瀏覽器公司
- 這引起了美國《創投天使》的瘋狂投資，並且認為有可能藉此打敗微軟，成為軟體網路新霸主！
- 後來微軟為了反制 Netscape，買下了 Internet Explorer 免費發放，而且強制綁在 windows 裏出貨

為了在瀏覽器大戰中爭取勝利

- Netscape 想到要在《瀏覽器》當中，加入一個小型的程式語言，可以用來做《功能表浮現與隱藏》等當時網頁沒辦法做到的功能。
- 而且為了拉攏夥伴《昇陽》對抗《微軟》，結果他們決定把原本稱為 LiveScript 的語言改名為 JavaScript。（因為當初昇陽在推 java 語言）

可惜的是

- 由於當時《微軟》透過 windows 壟斷了整個電腦軟體工業，windows 內建的 IE 成功地打敗了 Netscape 。
- 於是後來 Netscape 賣給 AOL，之後又捐出來給開放原始碼組織成立 Mozilla，現在成了《火狐》Firefox 瀏覽器。

雖然 Netscape 輸了

- 但是 JavaScript 卻沒死
- 而且成了瀏覽器中唯一能使用的程式語言。

到了 2009 年

- Ryan Lienhart Dahl 開發出 Node.js 之後
- JavaScript 開始進入 Server 和應用領域，讓這個原本只在瀏覽器上使用的語言，進入了《文藝復興》階段。

於是出現了更多的 JavaScript 開發環境與應用

- PhoneGap/Cordova 讓你可以用 JavaScript 寫手機應用
- Unity 讓你可以用 JavaScript 寫電腦與手機的遊戲
- Qt/QML 讓你可以用 JavaScript 寫視窗程式
- 微軟將 JavaScript 強型態化，推出 TypeScript 並在 Visual Studio 中大力支援 JavaScript 的開發
- 於是 JavaScript 開始從網頁設計入侵到其他應用領域

現在

- 我們已經介紹完 C 語言和 JavaScript 的特色與歷史了
- 接著讓我們回到主題，談談我為何在《程式設計》課程上採用 JavaScript+C 這個組合的原因。

這是因為

- 採用 JavaScript + C
- 可以得到最大的應用涵蓋度

我的想法是這樣的

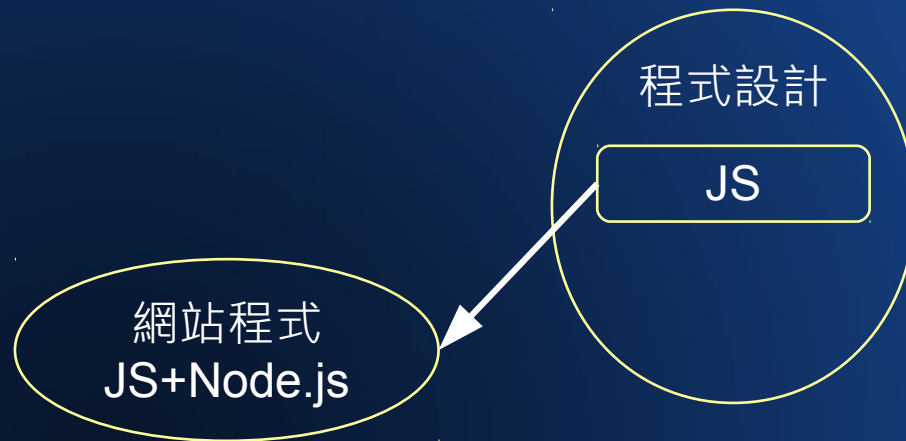
首先、在大一上

- 先從 JavaScript(JS) 入手，只要去除奇怪的語法，並避開陷阱，JS 並不會太難入手



一開始學 JavaScript 語法時

- 我們會採用 Node.js 寫命令列程式
 - 這樣可以避開 HTML/CSS/DOM 等瀏覽器技術
 - 降低學習的門檻！



等到 JS 語法學會了

- 就可以開始學習 HTML/CSS/DOM/ jQuery
等瀏覽器相關技術
- 快速地寫出一些互動網頁，這樣會比較有趣，比較有成就感
- 不會寫了很久，都還只能印一堆字

有了這些有趣的經驗之後

- 你應該會深深感到，自己的程式基礎並不足夠。
- 雖然程式基礎只有《變數、運算式、if、迴圈、陣列、函數》等等，最多再加上《物件》
- 但更重要的是《熟練度》與程式背後的《邏輯思維》
- 這是需要持續不斷地練習與思考，才能學會的

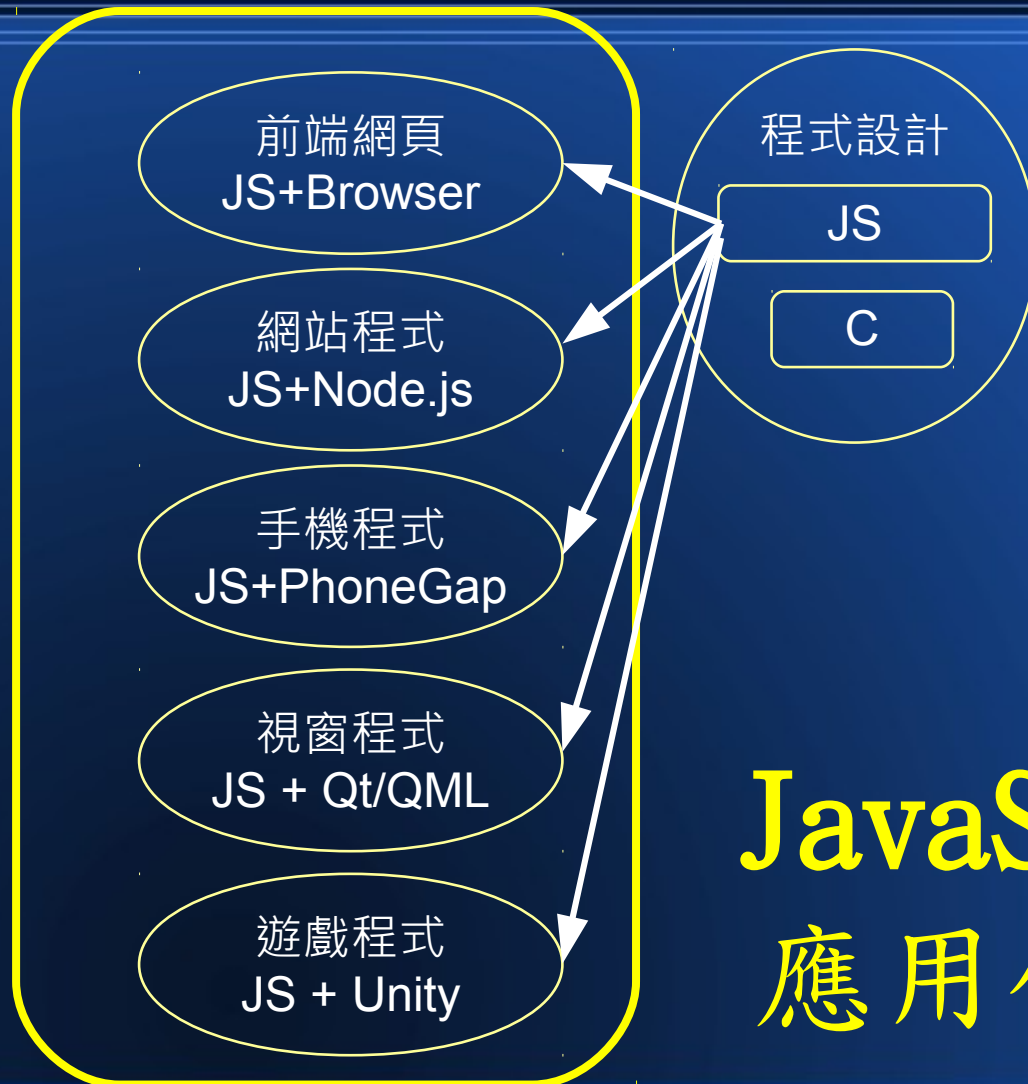
於是在第二學期開始

- 我們會引入《C 語言》
- 除了複習《程式的基礎語法》之外，還讓您能理解《另一種語言的思維模式》
- 然後您可以選擇自己想要繼續精進的方向與語言，繼續不斷地練習。

如果您選擇 JavaScript

- 那麼 Node.js+JavaScript 可以讓您學會完整的《網站前後端技術》
- 甚至還可以進入《視窗與手機》等本地應用程式領域。

這條 JavaScript 的道路 可以快速導向應用領域

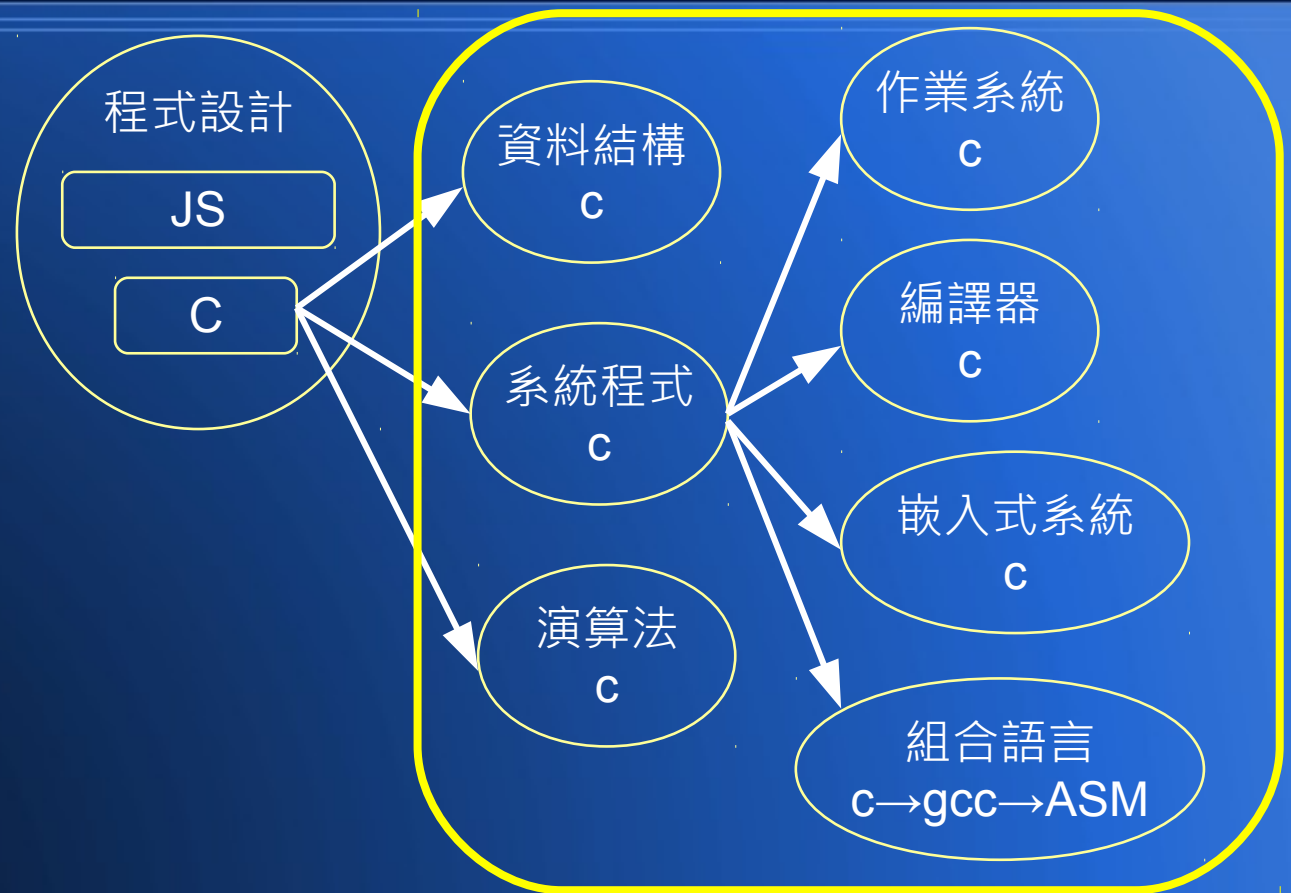


JavaScript 的
應用領域技術

但是如果你選擇 C 語言

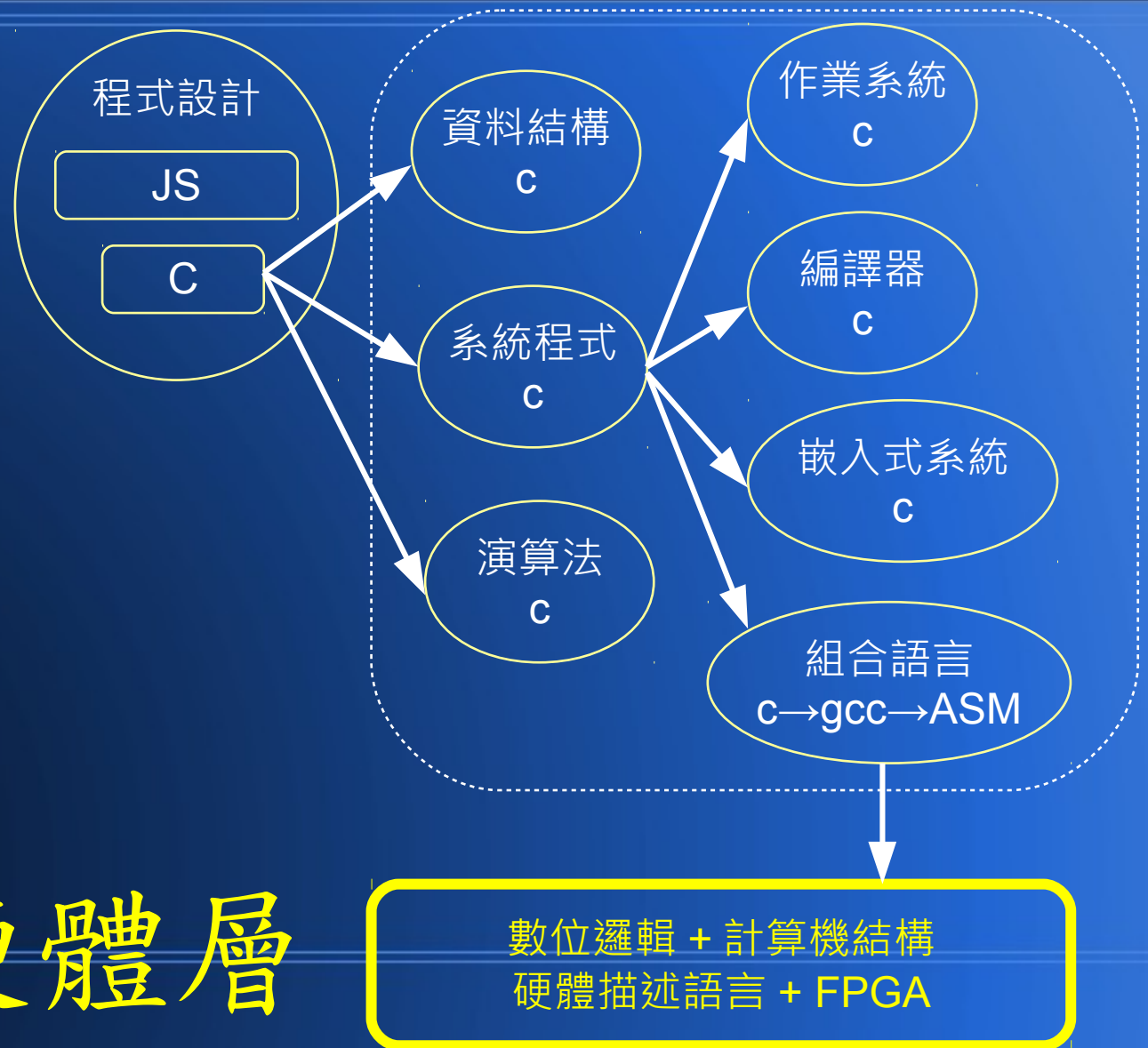
- 那麼你可以透過《資料結構》與《演算法》課程，學好《寫程式所需要的思維模式》。
- 然後進入《系統軟體》與《嵌入式系統》的領域。

C 語言是通往《系統層》的捷徑



《系統層技術》

甚至可以延伸到《硬體層》

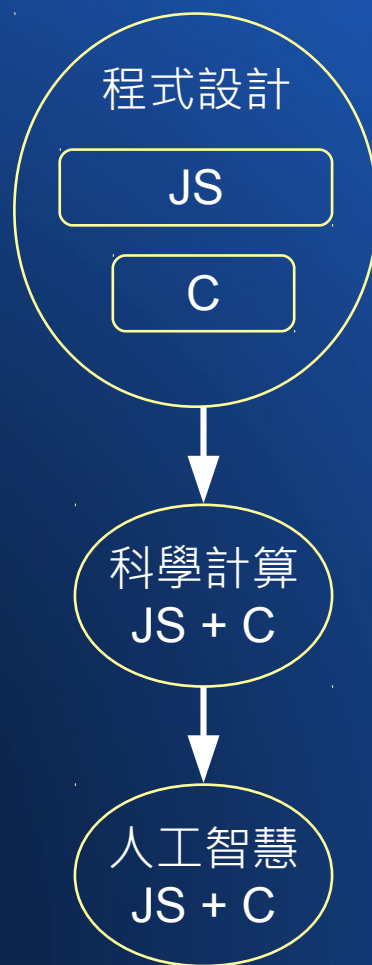


硬體層

然後

- 在三年級的時候，我們會延伸到一些比較困難的研究領域。
- 這時候我們需要很多數學基礎。
- 我們可以從《數學 \Rightarrow 科學計算 \Rightarrow 人工智慧》這條路走進研究領域。

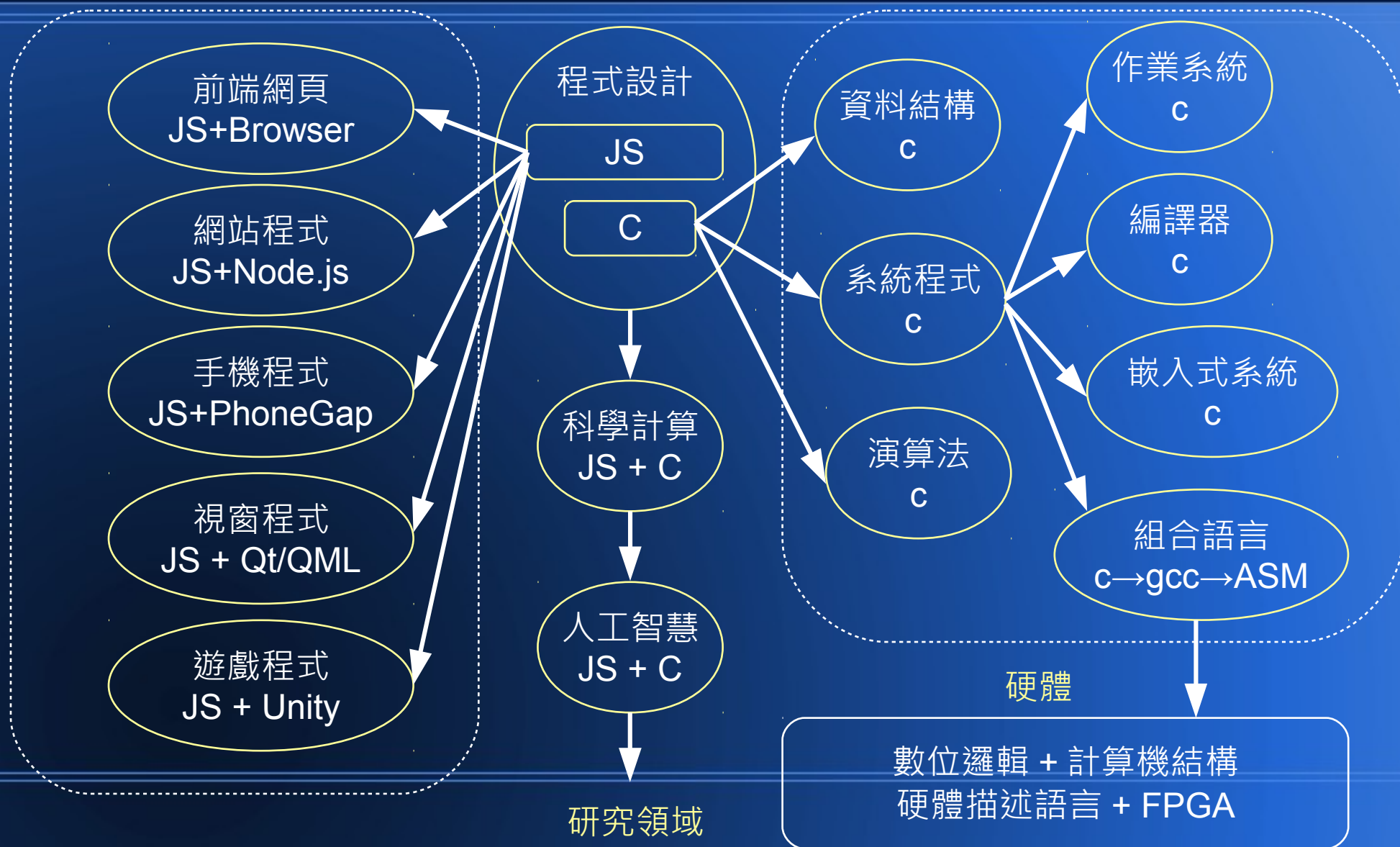
這是一條走向研究領域的道路



於是、整張課程地圖看來像這樣

應用軟體

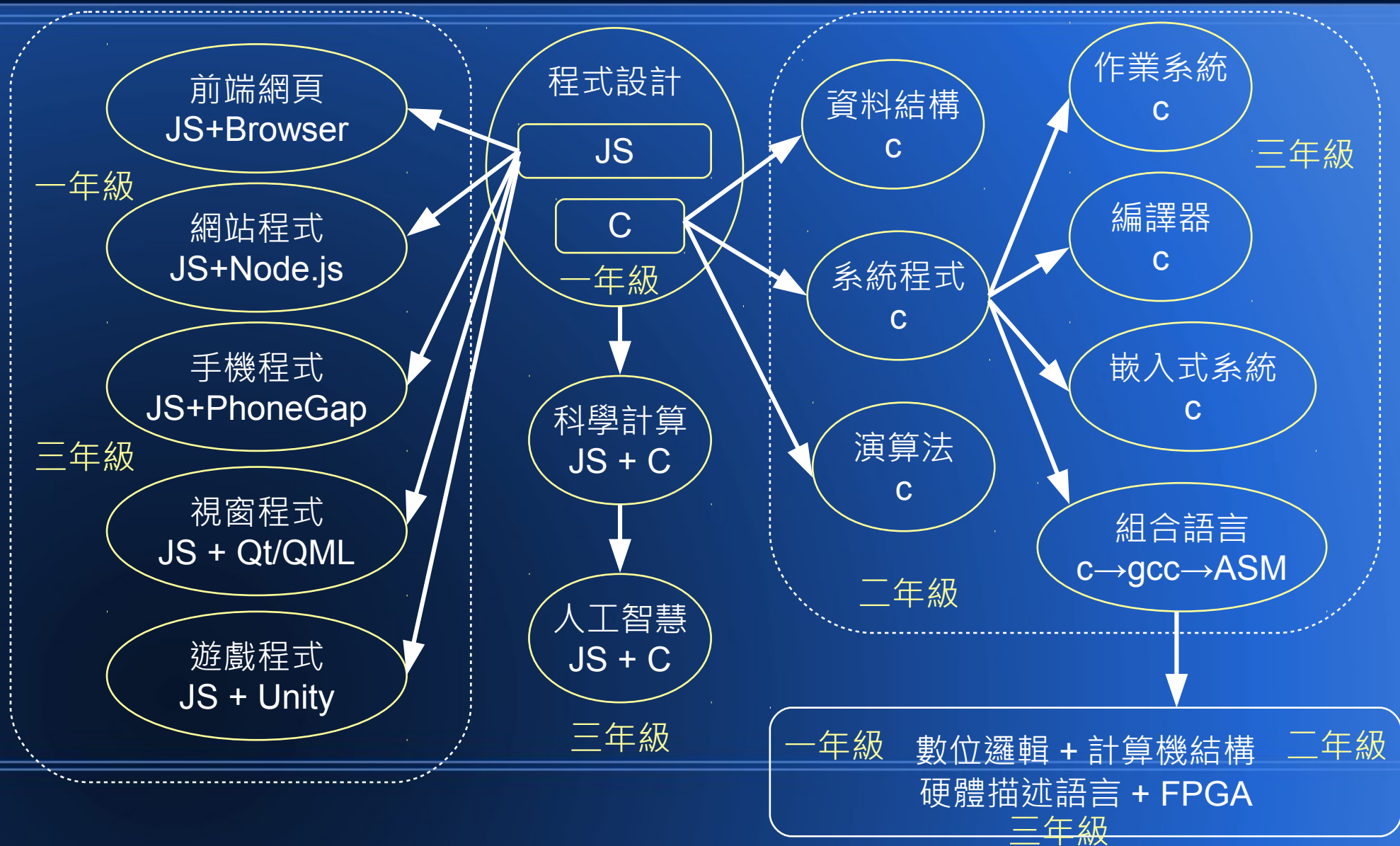
系統軟體



這張地圖

- 應該可以清楚的說明
我對這些課程的想法

您會分別在 1-3 年級學到這些課程



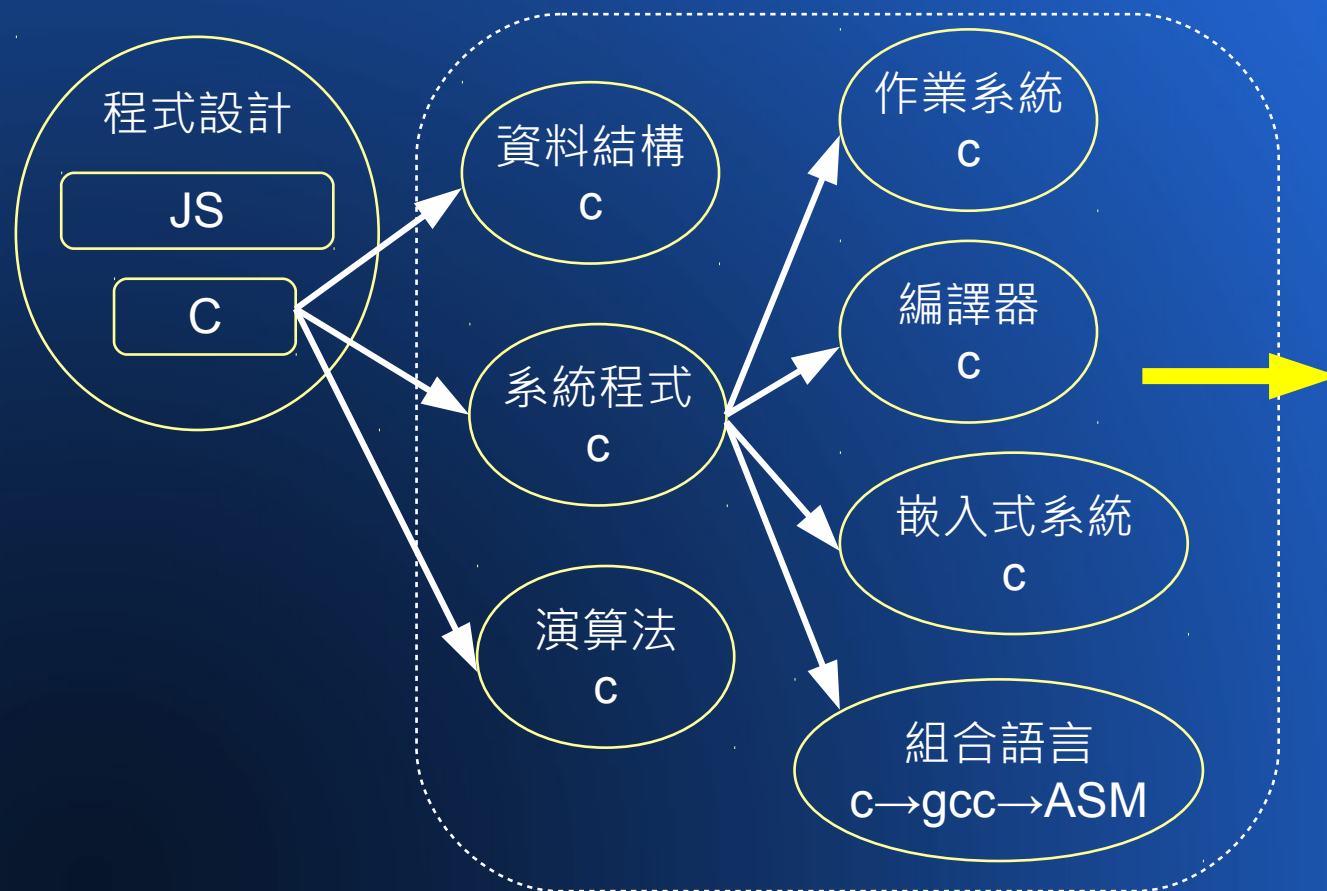
或許

- 您心裡還會有個問題？
 - 那四年級要學些甚麼呢？

我的想法是

- 四年級的時間盡量留給學生自己

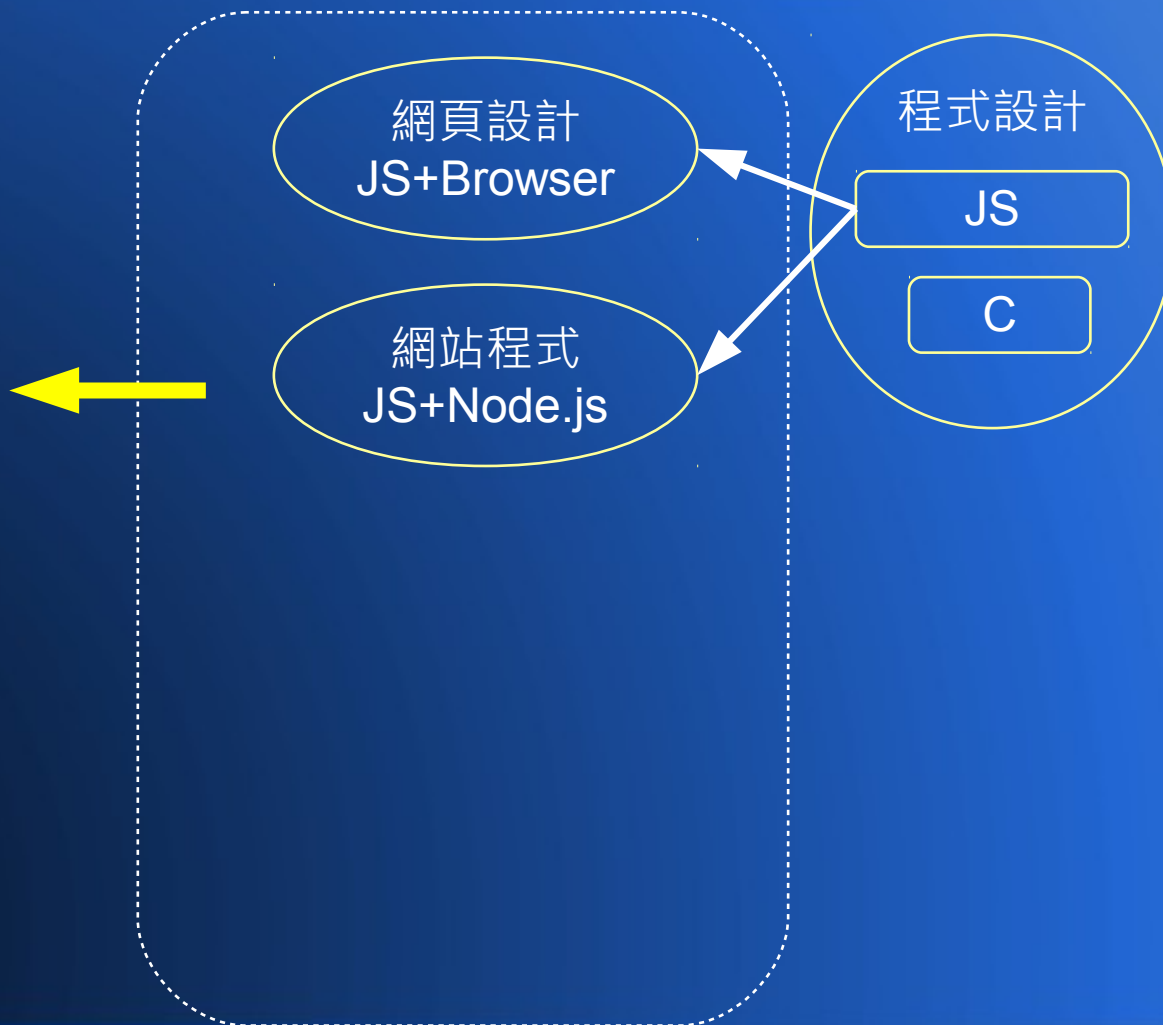
想走系統領域的可以走這條



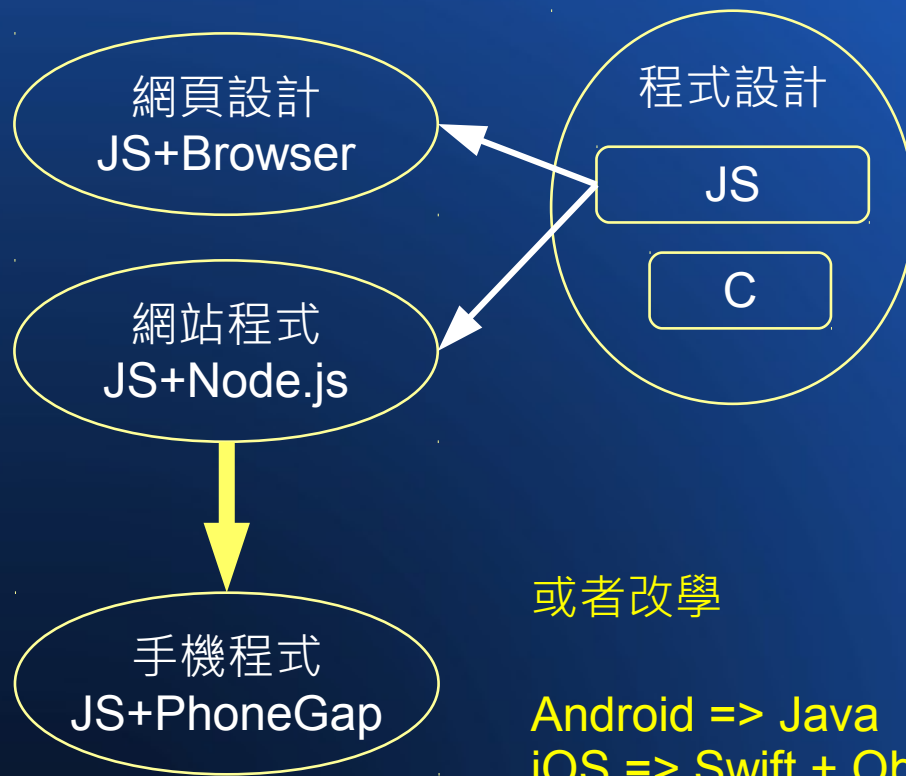
深入
系統層技術

想走《網站設計》的可以往這邊

學習進一步的
網站程式技術



想走手機 APP 的也可以這樣走



或者改學

Android => Java

iOS => Swift + Objective C

...

想走遊戲設計的 可以用 Unity 寫出好遊戲

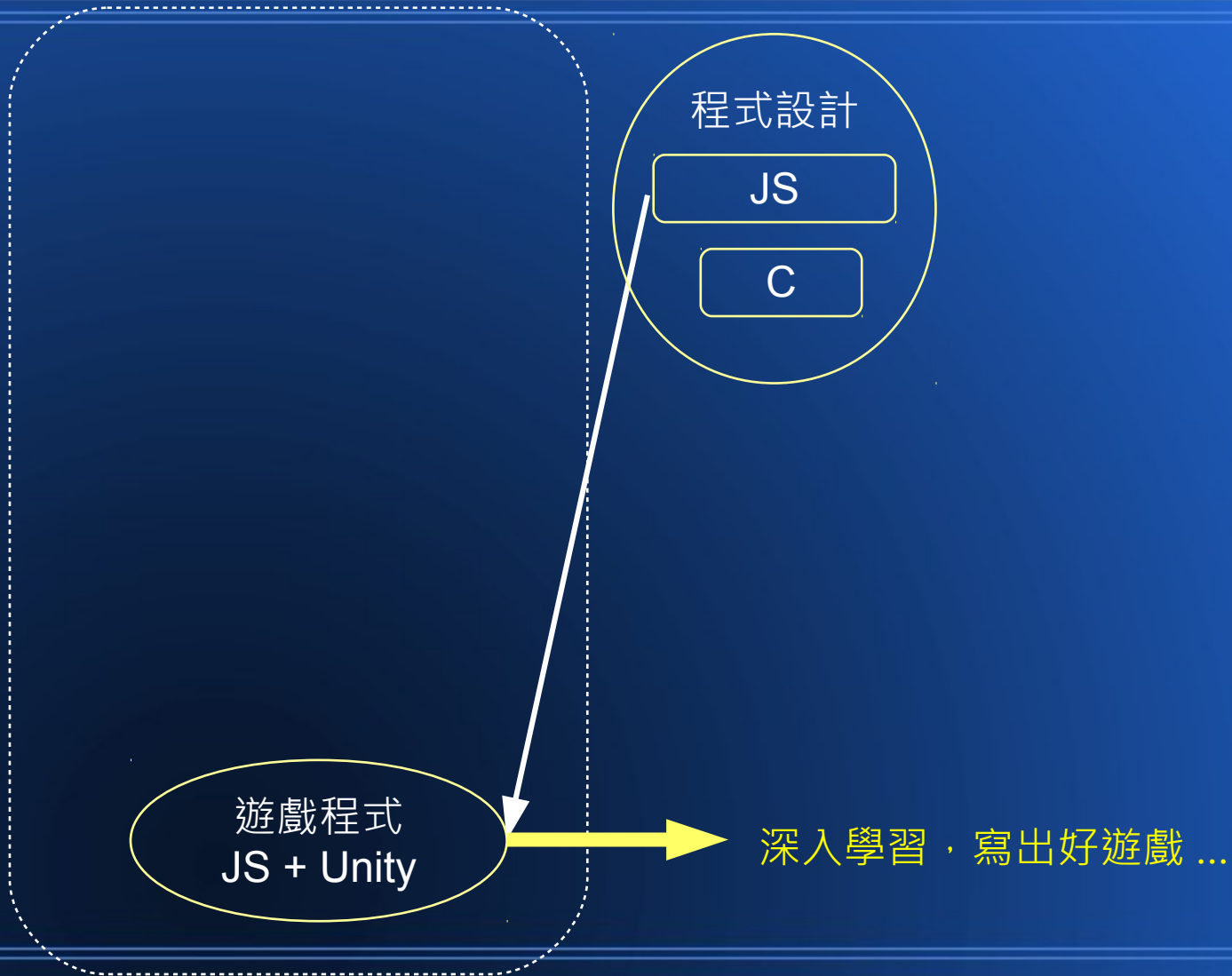
程式設計

JS

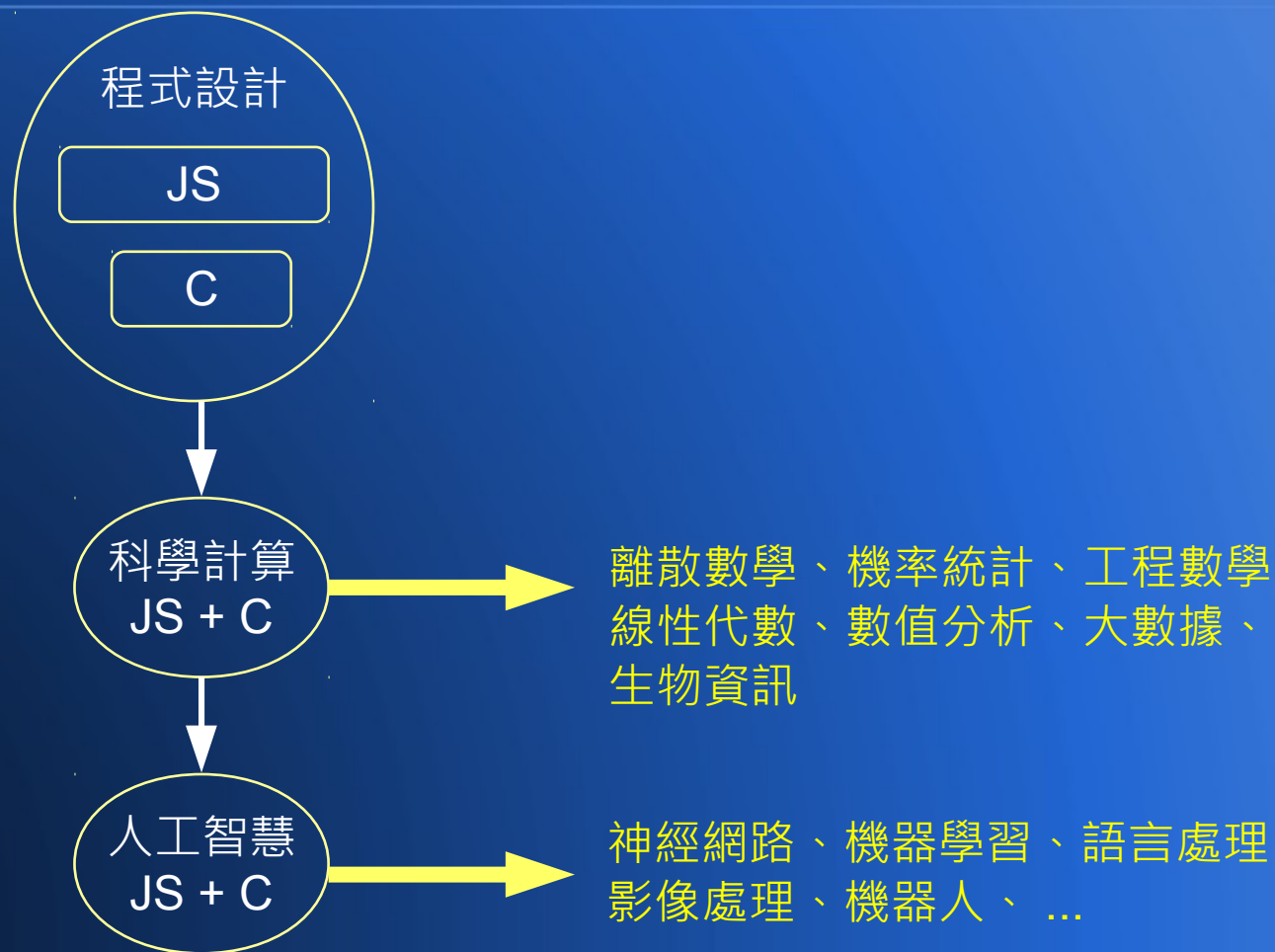
C

遊戲程式
JS + Unity

深入學習，寫出好遊戲 ...



而想走研究路線的可以考慮這條路



當然

- 還有更多我們沒想到的路可以走，像是《金融計算》、《去矽谷創業》、《去好萊塢做動畫》、《發明從來沒有人做得出來的東西》、《移民火星》、《搞政變推翻政府》、《創造自己的貨幣系統主宰世界》....

剩下的

- 那就是你們的事了！

自己的未來

- 還得靠自己去創造！

但是

- 沒有三兩三、怎敢上梁山
- 想要當土匪，也得先練好刀槍

在這個新世界

- 練刀練槍已經不太有用了
- 練習程式能力會有用得多！

不管你走哪一行

- 學會程式設計通常都可以用得上！

因為

- 在這個電腦無所不在的世界裡
- 程式就是最強大的武器！

只要你的程式夠強

- 你可以《去矽谷創業》、可以《寫程式賺錢》、可以《開發出比 Facebook 更棒的網站》、可以寫出《比 Airbnb 更賺錢的 APP》、甚至可以設計出《像夥計一樣的自動駕駛車》、或者《回收你發射的火箭》等等。

而這一切的一切

- 都得先學會《程式設計》

如果你從《金大資工》畢業

- 但是沒學會《程式設計》

記得

- 不要告訴別人你是我的學生

否則

- 我會覺得很丟臉！

因為

- 一個不會程式的資工系畢業生
- 出去絕對會被人歧視！

這就好像

- 一個不會煮菜的廚師一樣
- 是不可思議的一件事情！

但是

- 如果你《微積分、線性代數、離散數學、機率統計、物理化學、電子電路》... 等等科目沒學好
- 但是《程式能力很強》，那我仍然會以你為榮，因為沒有人會認為資工系畢業一定要會那些。

所以

- 請務必在大一的時候
- 學好程式設計這門課

這樣

- 後面學《資料結構、演算法、系統分析、資料庫、系統程式、作業系統、嵌入式系統、人工智慧、機器學習》等等課程的時候，你才知道到底那些程式怎麼寫，為甚麼那樣寫！

換句話說

- 只有學會程式設計

你在資工領域的生命

- 才算是真正有了個開頭
- 也才能夠繼續學習後續的那些課程！

所以

- 千萬要把握一年級的時光
- 努力學好程式！

拜託拜託！

我求您啦！

疑！這好像怪怪的，哪有老師這樣求學生的阿！