Introduction to Web Application Development

https://ihower.tw
2016/10/31

Who Am I?

- 張文鈿 a.k.a. shower
  - https://ihower.tw

- CTO and Instructor at ALPHA Camp
  - https://www.alphacamp.co

- Build web application since 2002

Ruby on Rails 實戰聖經

使用 Rails 4.2 及 Ruby 2.1

電子書和簡體版本準備中。如果您有任何意見、鼓勵或勘誤，歡迎來信給我，謝謝。

我是 ihower，本書介紹 Ruby on Rails 這套開放原始碼的網站開發框架。如果您對這本書有任何意見或勘誤指教，歡迎來信和我聯絡。

關於本書

回首頁

Part 1: 入門實作

1. Ruby on Rails 簡介
2. 安裝 Rails 開發環境
3. Rails 起步走
4. Ruby 程式語言入門
5. 手工打造 CRUD 應用程式
6. RESTful 應用程式
7. Active Record 基本操作與關聯設計
8. RESTful 綜合應用

Part 2: 深度剖析

1. 環境設定與 Bundler
2. 路由 (Routing)
3. Action Controller: 控制 HTTP 流程
4. Active Record: 資料表操作
5. Active Record: 資料庫遷移 (Migration)
6. Active Record: 資料表關聯
7. Active Record: 資料驗證及回呼
8. Active Record: 進階功能
9. Action View: 樣板設計
10. Action View: Helpers 方法
11. Ajax 應用程式
12. Assets Pipeline

alphacamp.cc
ALPHAcamp
ABOUT US
BOOTCAMP
STUDENTS
SEMINARS 講座
APPLY
COUNTRY
網站開發工程師實戰營
FULL TIME | 每週五天
十週 Ruby on Rails 課程實戰訓練，帶你學會 Rails、HTML、CSS 和 JavaScript，
打通前端、後端開發任督二脈，成為全球炙手可熱的網路全端工程師（Full Stack Developer），或親手實現創業的 idea！
瞭解更多
十週內打造全方位的 FULL STACK 開發能力
你將會學到
Ruby 程式語言
Rails 網站開發框架
前端網頁設計
版本控制開發流程
測試驅動開發
Web API 設計
網站效能調校
網站安全

Agenda

- What is Software Application and Why it’s important?
- Choose Software Product Platforms
- Front-End vs. Backend
- Choose Technology stack
- Developer Roles and Skills you need

1. What’s Application Software?

https://en.wikipedia.org/wiki/Application_software

電腦軟體可以分成

- 系統軟體
  - 作業系統、程式語言、編譯器、嵌入式系統等
  - 不希望有任何效能損耗，需要了解硬體、操控硬體，例如記憶體空間
- 應用軟體
  - 各種 App、桌面軟體、手機軟體、Web 應用等
  - 效能上可以有 trade-off，好寫好改 vs. 程式執行效能，來因應變來變去的商務需求

Why Software Is Eating The World

Internet Startup

- Startup is a business build to grow rapidly.
- Startup 都需要軟體 + 網路的技術，Why?

Engineering

- Getting something to work well enough for people to buy
  - 不一定是寫程式，還包括系統整合 (System Integration)
    - 培養對技術的敏感度，了解技術才可以評估和決定是否導入使用
- 用什麼工具都沒差? 用正確的工具才能事倍功半

Technologies?

- Operation System: Linux, iOS, Windows
- Cloud Hosting: AWS, Google, Azure
- DVCS: git
- Language: JavaScript, Ruby, Python, PHP, Swift, Java…
- Web Framework: Ruby on Rails, Django, Laravel
- Database: MySQL, PostgreSQL
- Web Server: Nginx, Apache
- Frontend Framework: React, AngularJS, Vue.js, Bootstrap

2. How to Choose Programming Language?

程式語言這麼多？怎麼選？

程式語言用來描述電腦如何工作

- Powerful
  - 可以處理超多資料、每次數億次操作(operations)
- Stupid
  - 每個操作很簡單機械、沒有見解或理解
  - 自然語言可以不明確，甚至有錯誤仍然可以理解。但是程式語言需要很精確，固定的語法(syntax)結構讓電腦可以了解

Quick Demo

- 打開 Chrome 瀏覽器
- 打開 Inspect 視窗的 Console
- 可以輸入 JavaScript 程式
- 或用 https://jsbin.com/ 也可以線上寫 JavaScript

有哪些程式語言?

- C
- C++
- Java
- C#
- Swift
- Objective-C
- Scala
- Go
- ….
- JavaScript
- PHP
- Python
- Ruby
- Perl
- R
- Erlang
- Haskell
- ….

Nally

Sites | Reconnect | Add Site | ptt.cc (Address) | Emoticon | Arti-Icle | Show Hidden Text | Double Byte

1. Ptt

【板主:femlro】 軟體板 徵才請看板規 系列《Soft_Job》

[←]離開 [→]閱讀 [Ctrl-P]發表文章 [d]刪除 [z]精華區 [i]看板資訊/設定 [h]說明

編號	日期	作者		文章標題	人氣:171
285	1010/28	RunRun5566	□	[閒聊] 想學新語言	
286	2111/23	PyRubyJavaC	□	[閒聊] 目前台灣哪種語言or技術 比較流行?	
287	2012/08	async	□	公司要求學其它的語言	
288	112/08	dlikeayu	R:	公司要求學其它的語言	
289	812/13	ukilue	□	[請益] 換工作/跳槽非已會語言能談高薪水?	
290	m2012/13	NDark	R:	[請益] 換工作/跳槽非已會語言能談高薪水?	
291	m2112/14	DrTech	R:	[請益] 換工作/跳槽非已會語言能談高薪水?	
292	6612/20	RunRun5566	□	[閒聊] 聊聊各位目前在寫的語言	
293	31 1/06	peace9527	□	[請益] 未來想走data science 該學什麼語言?	
294	8 1/15	csfgsj	□	[情報] 有關第四代語言分析的好文	
295	40 1/29	Non	□	[請益] 寫記帳程式該用什麼語言寫?	
296	29 3/16	pockychu	□	[請益] 要如何抉擇學哪一個程式語言...?	
297	12 4/13	dnabossking	□	[請益] 語言真的不重要?	
298	2 4/14	csfgsj	R:	[請益] 語言真的不重要?	
299	18 4/14	LaPass	R:	[請益] 語言真的不重要?	
300	4 4/15	batista0630	R:	[請益] 語言真的不重要?	
301	7 4/15	superpai	R:	[請益] 語言真的不重要?	
302	59 5/31	SSiuan	□	該衝刺哪個程式語言	
303	m 4 6/02	AmosYang	R:	該衝刺哪個程式語言	
304	~25 6/30	wilson50101	□	[討論] [討論] 關於程式語言用到的網路觀念	

文章選讀 (y)回應(X)推文(^X)轉錄 (=[]<>)相關主題(/?a)找標題/作者 (b)進板畫面

為什麼這麼多種語言?

- 不同的程式語言有不同的抽象化方式，用來增加效率
- 效率的定義(是執行快? 還是開發快?)、語法風格每個人看法不同
- 因應開發不同類型的軟體，有不同的語言設計
- 因為上帝要阻止人類興建巴別塔

低階語言

- 機器語言、組合語言
- 程式碼寫起來很費事，但電腦跑起來很快
- 不同硬體的 CPU 指令集不一樣，例如
  - Intel x86 和 ARM 不一樣
  - 64位元和32位元也不一樣

機器語言 Machine code

- 描述 CPU 指令 + CPU 暫存器 + 記憶體位址
  - 000000 00001 00010 00110 00000 100000
  - 100011 00011 01000 00000 00001 000100
  - 000010 00000 00000 00000 10000 000000

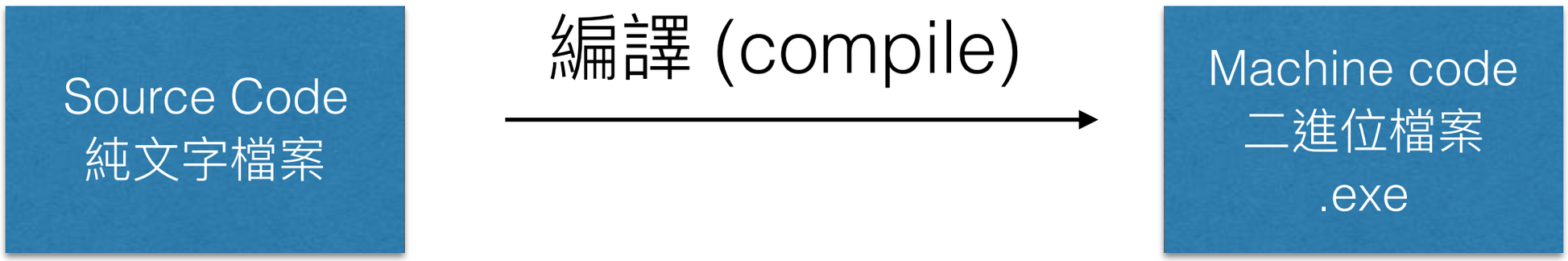
組合語言 Assembly

```
MOV eax, 1
ADD eax, 4
SUB eax, 2
MOV num, eax
INVOKE printf, ADDR formatStr, num
ret    0
```

高階語言

- C 語言、Java 語言、PHP/Python/Ruby 等等
- 透過結構化程式設計，開發比較好寫
    - function 子程式、程式碼區塊、for迴圈以及while迴圈等等結構
- 經過編譯後(Compile)轉成機器碼

編譯 Compile



C 語言

- 自組語之後，最重要的開發效率進步
- 可用 Pointer 指標操作記憶體
- 最重要的系統程式語言
  - 高效能
  - 經過編譯可以移植到不同硬體上
- C++ 不是更好的 C，別搞混了

```c
#include <stdio.h>
int main() {
    int n, i, sum = 0;

    printf("Enter a positive integer: ");
    scanf("%d",&n);

    for(i=1; i <= n; ++i) {
        sum += i;   // sum = sum+i;
    }

    printf("Sum = %d",sum);
    return 0;
}
```

電腦軟體可以分成

- 系統軟體
  - 作業系統、程式語言、編譯器、嵌入式系統等
  - 不希望有任何效能損耗，需要了解硬體、操控硬體，例如記憶體空間
- 應用軟體
  - 各種 App、桌面軟體、手機軟體、Web 應用等
  - 效能上可以有 trade-off，好寫好改 vs. 程式執行效能，來因應變來變去的商務需求

Java, C#, Swift, Objective-C

- 開發應用軟體的程式語言
- 支援物件導向
- 不需要管理記憶體，開發效率提高
- 需要編譯

JavaScript, PHP, Python, Ruby

- 也是開發應用軟體的語言
- 支援物件導向
- 也不需要管理記憶體
- 不需要編譯，開發效率更高
- 缺點是執行效能較慢
- Startup 愛用

直譯器 (interpreter) 直接執行 Source Code 程式碼

JavaScript, PHP, Python, Ruby

- JavaScript 託瀏覽器的福佔領了世界，近年來用 Node.js 擴展到後端
- PHP 因為 Web 而發明
- Ruby/Python 從 scripting 的需求到 web application 時代的興盛

最熱門？最建議?

https://www.sitepoint.com/whats-best-programming-language-learn-2015/

TIOBE Index for June 2016

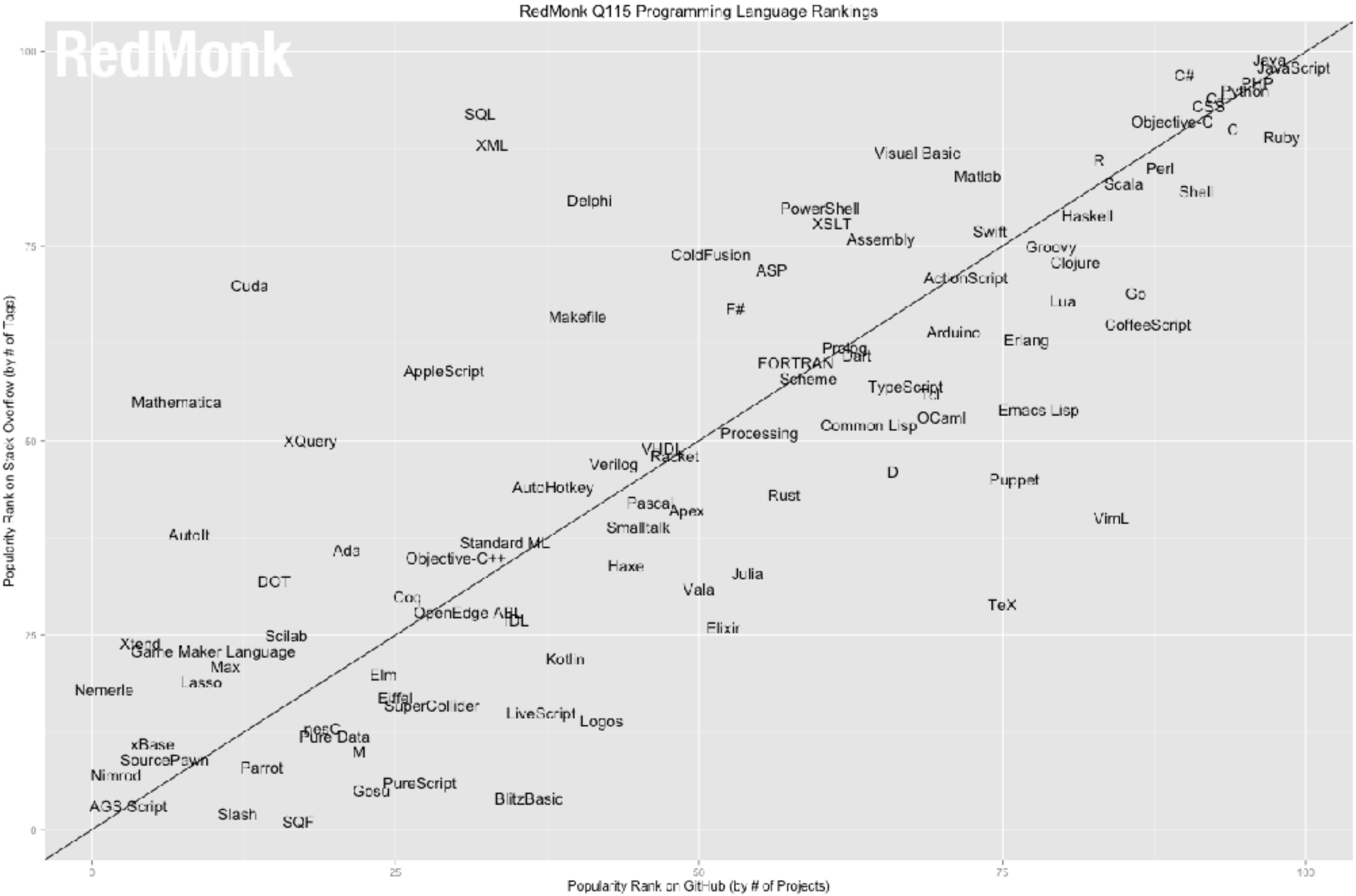
June Headline: The Long Tail of Programming Languages

For the first time in the history of the TIOBE index a language needs to have a rating of more than 1.0% to be part of the top 20. What does this mean? This indicates that the number of real market leaders is going down. The set of languages to chose from is getting bigger and more and more less well-known programming languages are being adopted. About 10 years ago, the first 8 language covered 80% of the market, now this is reduced to 55%. This phenomenon is also called the long tail, a term that has been popularized by Chris Anderson of Wired in 2004.

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found here.

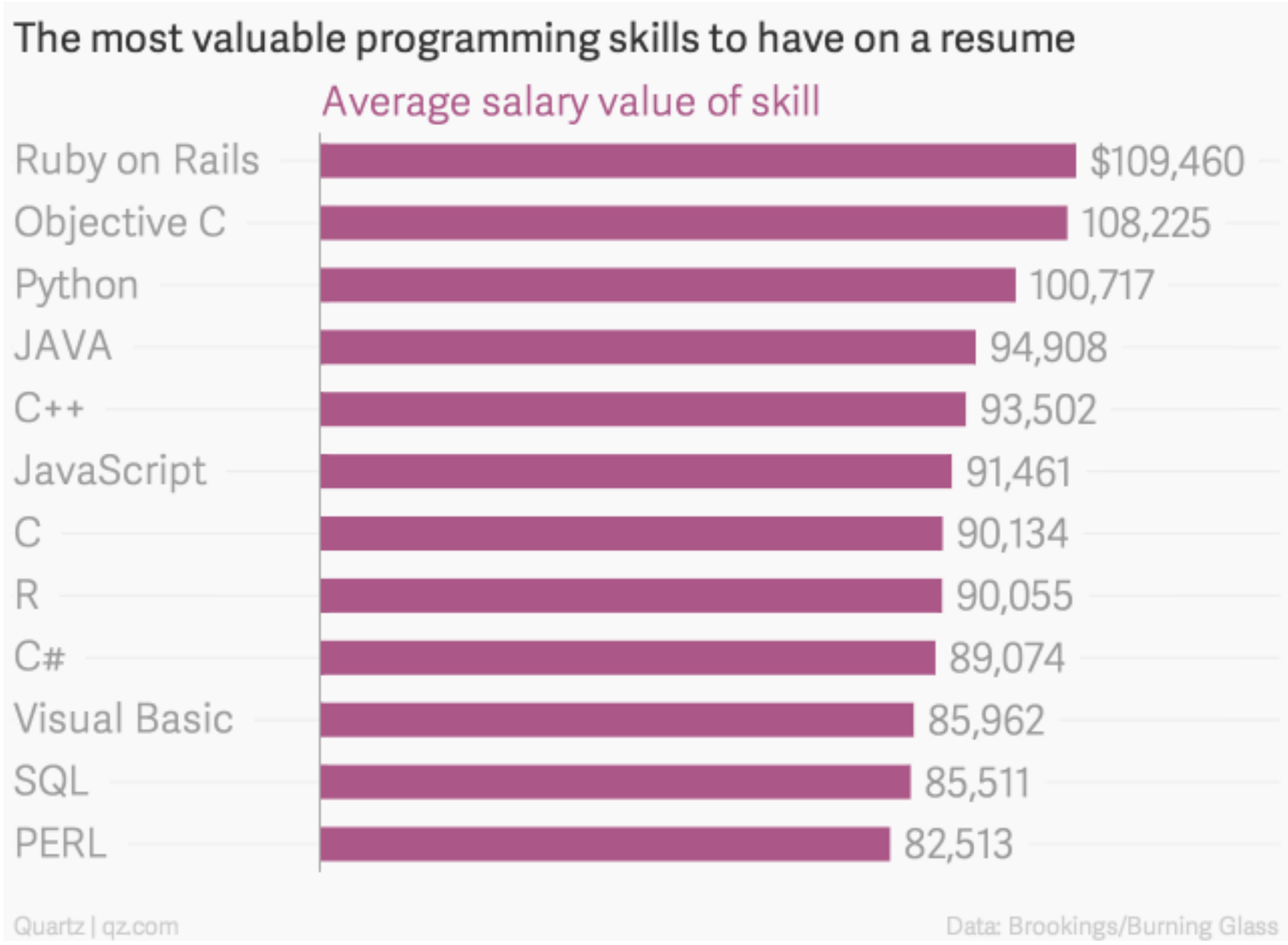
Jun 2016	Jun 2015	Change	Programming Language	Ratings	Change
1	1		Java	20.794%	+2.97%
2	2		C	12.376%	-4.41%
3	3		C++	6.199%	-1.56%
4	6	↑	Python	3.900%	-0.10%
5	4	↓	C#	3.786%	-1.27%
6	8	↑	PHP	3.227%	+0.36%
7	9	↑	JavaScript	2.583%	+0.29%
8	12	⇈	Perl	2.395%	+0.84%
9	7	↓	Visual Basic .NET	2.353%	-0.82%
10	16	⇈	Ruby	2.336%	+0.96%
11	11		Visual Basic	2.254%	+0.41%
12	23	⇈	Assembly language	2.119%	+1.36%
13	10	↓	Delphi/Object Pascal	1.939%	+0.07%
14	14		Swift	1.831%	+0.39%
15	5	⇊	Objective-C	1.704%	-2.64%


RedMonk Q115 Programming Language Rankings
RedMonk
Popularity Rank on Stack Overflow (by # of Tags)
Popularity Rank on GitHub (by # of Projects)
0
25
50
75
100
Java
JavaScript
C#
PHP
Python
CSS
Objective-C
C
Ruby
SQL
XML
Visual Basic
R
Perl
Matlab
Scala
Shell
Delphi
PowerShell
XSLT
Haskell
Swift
Assembly
Groovy
ColdFusion
Clojure
ASP
ActionScript
Cuda
Go
Lua
F#
Makefile
CoffeeScript
Arduino
Erlang
Prolog
Dart
FORTRAN
AppleScript
Scheme
TypeScript
Mathematica
Emacs Lisp
OCaml
Common Lisp
Processing
XQuery
VHDL
Racket
Verilog
D
Puppet
AutoHotkey
Rust
Pascal
Apex
VimL
Smalltalk
AutoIt
Standard ML
Ada
Objective-C++
Haxe
Julia
DOT
Vala
Coq
TeX
OpenEdge ABL
IDL
Elixir
Scilab
Xtend
Game Maker Language
Kotlin
Max
Elm
Lasso
Nemerle
Eiffel
SuperCollider
LiveScript
Logos
nesC
Pure Data
xBase
M
SourcePawn
Parrot
Nimrod
PureScript
Gosu
BlitzBasic
AGS Script
Slash
SQF

Programmer 一定知道的網站

- Github: 開源軟體大本營
- Stackoverflow: Q&A 網站

挑戰年薪 300 萬，學這些程式語言就對了！



http://www.inside.com.tw/2014/11/21/these-programming-skills-will-earn-you-the-most-money

別人用什麼?

- Java: Google, Oracle
- Swift, Objective-C: Apple
- C#: Microsoft
- PHP: wikipedia, vimeo, facebook
- Ruby: airbnb, shopify, github, twitter, groupon, basecamp, hulu+
- Python: youtube, quora, google, instagram, pinterest

學哪種語言有差嗎?

- 程式概念的確可以跨語言，有很多 common concept
- 但是 ecosystem 不一樣，擅長的情境不一樣，一般來說：
  - 開發系統程式(例如作業系統)，適合 C 語言
  - 開發 Web 後端應用，適合 PHP/Ruby/Python/Node.js
  - 開發 Web 前端應用，得用 JavaScript
  - 開發 Android 應用，得用 Java
  - 開發 iOS 應用，得用 Swift 或 Objective-C

衡量因素

- 社群支援: 用的人多嗎? 可以找到學習資源嗎? 找的到人問嗎?

- 開發難易度: 開發你要的 app 適合嗎? 開發 chat app? 開發 mobile app? 不同應用適合不同的語言

- 你身邊有的資源，誰可以幫助你?

    - 例如 ALPHCamp 教 Swift 跟 Ruby、JavaScript，沒有教 C#、PHP

http://carlcheo.com/wp-content/uploads/2014/12/which-programming-language-should-i-learn-first-infographic.png

WHICH PROGRAMMING LANGUAGE

SHOULD I LEARN FIRST?

WHAT IS PROGRAMMING?

Writing very specific instructions to a very dumb, yet obedient machine.

LANGUAGES

PYTHON · JAVA · C · PHP · C++

JAVASCRIPT · C# · RUBY · OBJECTIVE-C

START

WHY DO YOU WANT TO LEARN PROGRAMMING?

I don't know, just pick one for me

Just for fun

I'm interested!

Improve myself

Make money

I have a startup idea!

Have a brilliant idea/platform in mind?

YES

Which platform/field?

Nope. Just want to get started

I prefer to learn things...

The easy way

The best way

The slightly harder way

Auto or Manual car?

Auto

Manual

The really hard way (but easier to pick up other languages in the future)

NOT SURE

NO

Which one is your favourite toy?

Lego

Play-Doh

I've an old & ugly toy, but I love it so much!

Do you want to try something new with huge potential, but less mature?

YES

NO

Does your web app provides info in real-time, like twitter?

Web

Mobile

Which OS?

IOS

Android

3D/Gaming

Enterprise

Start with Scratch, then move on to...

For my kids

Which platform/field?

I want to work for big tech companies

Get a job

Front-end (web interface)

Web

Back-end ("brain" behind a website)

Doesn't matter, I just want $$$

I want to work for...

Corporate

What do you think about Microsoft?

I'm a fan!

Not Bad

Suck

Startup

THE LORD OF THE RINGS ANALOGY TO PROGRAMMING LANGUAGES

3. How to choose platforms?

Platforms (依照技術分類)

- Native
  - 直接跑在作業系統上
  - 例如 MS Word、瀏覽器、手機上的 LINE app 等
- Web
  - 在瀏覽器環境中執行
  - 例如 Gmail、Facebook、Twitter

Native

- Desktop 桌面軟體
  - Windows 作業系統
  - MacOS 作業系統
- Mobile 移動式裝置
  - iOS 作業系統
  - Android 作業系統
- Cross-platform Native
  - 例如 React Native, Ionic, NativeScript, RubyMotion, Appcelerator 等等，透過編譯器工具轉成 Native 程式。

Web

- Web
  - 桌機的 Browser: Chrome, Safari , Firefox, IE
  - 手機的 Browser (Mobile Web): Safari, Chrome
  - 包裝成手機的 App，但 App 裡面跑得其實是 Browser 環境 (Hybrid Mobile)

What’s web?

- HTML/CSS (Markup Language)
- JavaScript
- Quick Demo
    - https://startuprookies.alphacamp.co/

Native vs. Web

- Native
  - 執行速度快
  - 100% 善用硬體資源
- Web
  - 跨平台
  - 佈署方便快速

	Native Mobile	Web	Hybrid/Cross-platform
執行效能	快	慢	慢/快
佈署	App Store 上架	自己的伺服器	App Store 上架
相機、Notification、通訊錄、離線應用	支援	支援程度不一	支援
Geolocation	支援	支援	支援
觸控	支援	支援程度不一	支援
開發語言	Swift, Java	HTML/CSS/ JavaScript	HTML/CSS/ JavaScript

	Web 2013	Web 2014	Web (2015)	Native
Deep Linking	YES	YES	YES	KINDA
Single click install and launch	YES	YES	YES	NO
Geo	YES	YES	YES	YES
Gyro	YES	YES	YES	YES
Offline	KINDA	KINDA	YES	YES
Camera	KINDA	YES	YES	YES
Push	NO	NO	YES	YES
Contacts	NO	NO	NO	YES
Auth	NO	NO	NO	YES
Payments	NO	NO	NO	YES

https://paul.kinlan.me/the-headless-web/

Native App or Web App?

- 安裝程序和軟體發佈流程
- 跨平台
- 硬體存取權限
- 效能和使用者體驗
- 離線應用
- 買下軟體 or 租用軟體的感覺
- Open vs. Vendor lock-in

iOS or Android?

- Startup 通常偏好先 iOS 來作 MVP 產品?
- iOS 比較好開發
- iOS 使用者比較會付錢
- Android 手機版本破碎

Desktop App?

- 越來越少人在談了?
- 商務應用逐漸轉移到 Web Apps
- 需要顯示卡運算能力的應用
  - 例如遊戲

Website vs. Web app

A Website is Not an application software

Website vs. Application

- Frequency of Use
- Direction of Data and Content
- Navigation vs. Participation
- Presence of Accounts
- Pages or Flows
- Beyond the Browser

Web 的兩種特性

- Document retrieval system
- Application delivery system

從 Website 進化到 Webapp

- 靜態網頁
- PHP&MySQL 動態網頁
- Flash
- AJAX
- Progressive Enhancement with JavaScript
    - HTML for structure, CSS for appearance, JS for behavior
- Rich Internet application (RIA) 較廣義包括 Flash, Silverlight
- Single Page Application (SPA) 專指 HTML-based solution

Platforms (依照應用分類)

- Desktop
  - Native
  - Web 瀏覽器
  - Hybrid app
- Mobile
  - Native app
  - Hybrid web app
  - Mobile web

Desktop Apps

- Native
  - Windows (C#)
  - MacOS (Swift)
- Cross-platform Native
- Web 瀏覽器: Chrome, Safari, Firefox
- Hybrid 技術 (包裝成 App，但是裡面程式碼是 Browser 環境，執行 JavaScript)
  - 例如 Electron http://electron.atom.io/

Mobile Apps

- Native
  - iOS (Swift)
  - Android (Java)
- Cross-platform Native
- Mobile Web (HTML5)
- Hybrid mobile 技術，例如 Phonegap, Cordova, Framework7, Turbolink3 等等

https://www.agingcoder.com/mobile/2016/09/25/the-state-of-html-mobile-frameworks-in-2016/

Mobile web

- HTML5
- 除了桌機版本的網頁，另外再製作手機版本
- 或是採用 (RWD) Responsive Web Design 設計
  - 隨著螢幕大小，變化樣式為適合手機大小
  - http://www.ibest.tw/page03.php

	桌面應用	移動式裝置
Native	MacOS (Swift) Windows (C#)	iOS (Swift) Android (Java)
Cross-Compile (Native)	React Native、Ionic、NativeScript、Appcelerator (JavaScript) RubyMotion (Ruby)	
Hybrid (Web)	Electron NW.js	Phonegap Cordova Framework7 Turbolink3
Web	JavaScript: Chrome IE/Edge Firefox	JavaScript: Chrome Safari

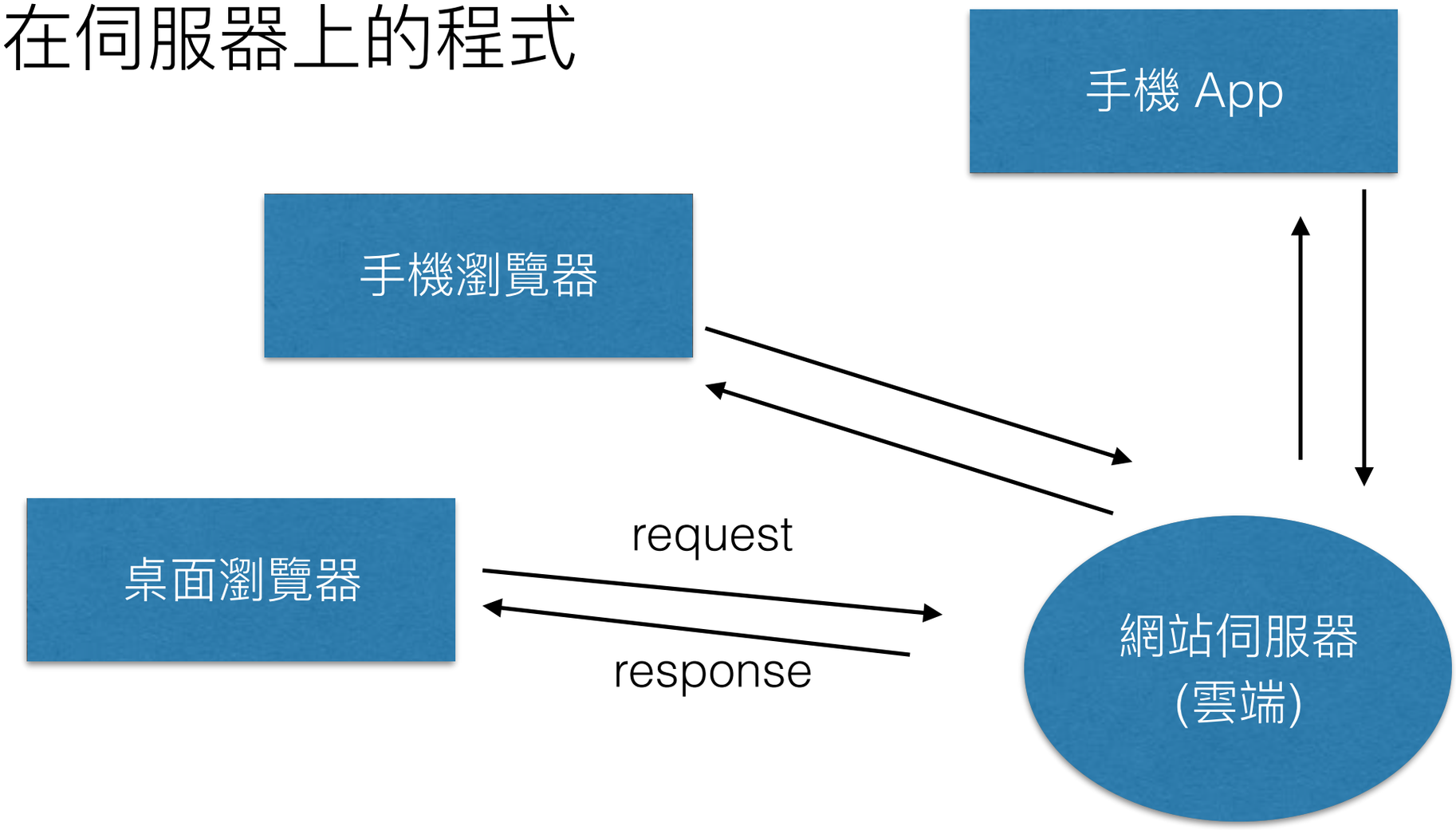
Cross-platform 和 Hybrid 技術

- 有些是 Open Source、有些則需要付費購買編譯器的軟體授權，這些技術看起來很 cool?
- 優點
  - 團隊可以用偏好的程式語言，例如 JavaScript 或 Ruby，減少團隊學習新程式語言的成本，部分程式碼也可能共享
  - Cross-platform 可以讓 Mobile App 與 Web App 有類似的架構(例如 React Native、NativeScript)、或是跟後端一樣的語言 (例如 RubyMotion)
  - 若已經有 Web App，那再開發 Hybrid App 的成本較低。而且佈署可以很即時，因為部分內容可以跟 Web App 一樣用網路即時下載執行。
- 缺點
  - Vendor-lock 需要依賴一間廠商，有些甚至需要授權費用
  - 學習曲線不一定比較低，畢竟多一個工具抽象層就會帶來一些新的坑 (Bugs)
  - Hybrid 的效能較差，使用者未必能夠接受長的像 App 但跑起來像 Web

3. Front-end vs. Back-end

前端 vs. 後端

- 前端指的是跑在用戶端的軟體
- 後端跑在伺服器上的程式



前端 + 後端

- 前端 Client-Side
    - 提供使用者 User Interface 和 User Interaction
- 後端 Server-Side
    - 儲存、計算和分析資料

前後端技術

- 前端：需要考量使用者用什麼平台
  - Mobile: Swift 和 Android
  - Web: JavaScript/HTML/CSS
- 後端：除了程式語言，還需要考慮作業系統、Web 伺服器、資料庫
  - 基本上是 whatever you like
  - 程式語言常用 PHP/Ruby/Python 和 Java/C#

Web 前端後端怎麼分?

- HTML = Template + Data
- 那由誰負責 Rendering? Browser (JS) 或 Server-side
  - Browser 和 Server-side rendering 的優缺點?
    - 執行效率、開發難易、SEO...etc
  - Server-side rendering 的話，前後端如何分離? 小團隊需要分離嗎?

https://hashnode.com/post/server-side-rendering-vs-client-side-rendering-which-one-is-faster-and-why-cilwa02js00ah0s53de09rbdg

Template 技術分類

- 後端 Template，例如 PHP, Ruby on Rails
- 前端 Template
  - Backbone
  - AngularJS
  - React

Demo

https://github.com/ihower/intro-to-webdev-code

- 安裝 Homebrew 和最新版 Ruby
  - http://brew.sh/
  - brew install ruby
    - 重開一個視窗，輸入 ruby -v 應該顯示 Ruby 2.3.1
- 安裝 http://get-serve.com/
  - gem install serve
  - 執行 server 就可以打開 http://localhost:4000 瀏覽該目錄
  - 附檔案 .erb 則是 Ruby 的 template 會被 render

4. How to Choose Tech Stack?

https://www.flickr.com/photos/paulafunnell/8653411426

開發框架

程式語言

資料庫

Web 伺服器

作業系統

Technology Stack

- Server Operation System
- HTTP Server
- Hosting
- Development Tools
  - Mac v.s. Windows
  - Version Control
  - Editor v.s. IDE
- Browsers
- Programming Languages
- Front-end web framework & library
- Back-end web framework & library
- Framework v.s. Library
- Database
- Licenses

看看別人用什麼？

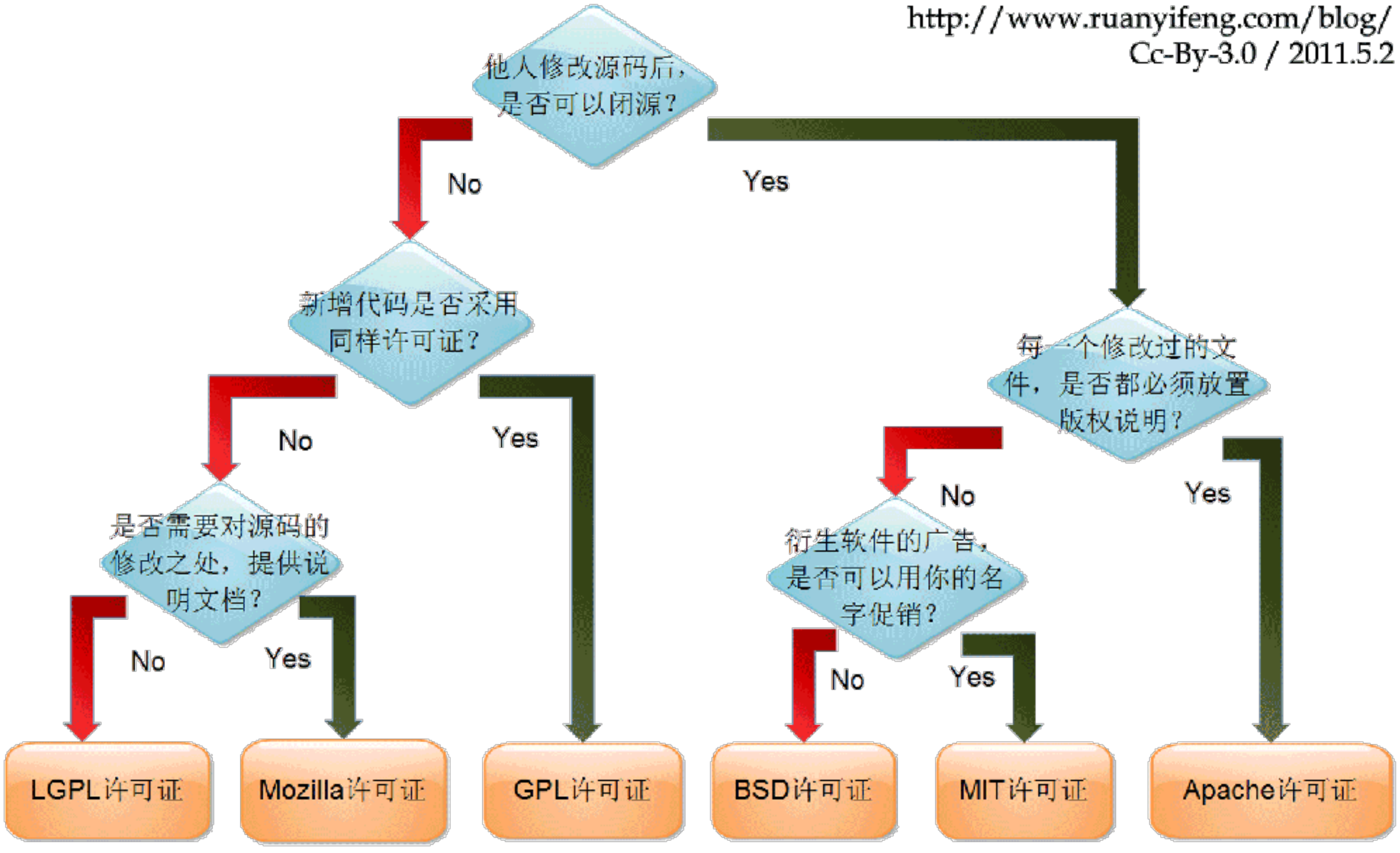
- http://builtwith.com/
- http://stackshare.io/
- Facebook 用的技術，就是適合我的技術嗎？
- Google 用的技術，就是適合我的技術嗎？

用現成的還是自己開發?

- in-house
- commercial
- or use open source?

Open Source

- 你可以自由使用、下載、修改與散佈的程式軟體
- 自由軟體(Free Software) v.s. 開源軟體 (Open Source Software)
- Open Source License
  - GPL, LGPL, AGPL 等 不會授予授權條款接受人無限的權利。再發行權的授予需要授權條款接受人開放軟體的原始碼，及所有修改。且複製件、修改版本，都必須以GPL為授權條款。
  - BSD, MIT 等，可以閉源



Why Open Source?

- 共享和開源有助於自己更嫻熟地掌握相關知識
- 共享和開源共享有助於提高自己的計畫品質
- 開源和共享能夠讓你免費利用大家的智慧與勞動
- 開源與共享是推銷自己的最好方式
- 開源和共享能夠讓你獲得對計畫的擁有權

https://share.inside.com.tw/posts/27427

OS

- OS X for local development and daily use
  - Mac 的作業系統和 Linux 系出同源，同屬 Unix-like 系統
- Linux for production server
- Windows for testing IE

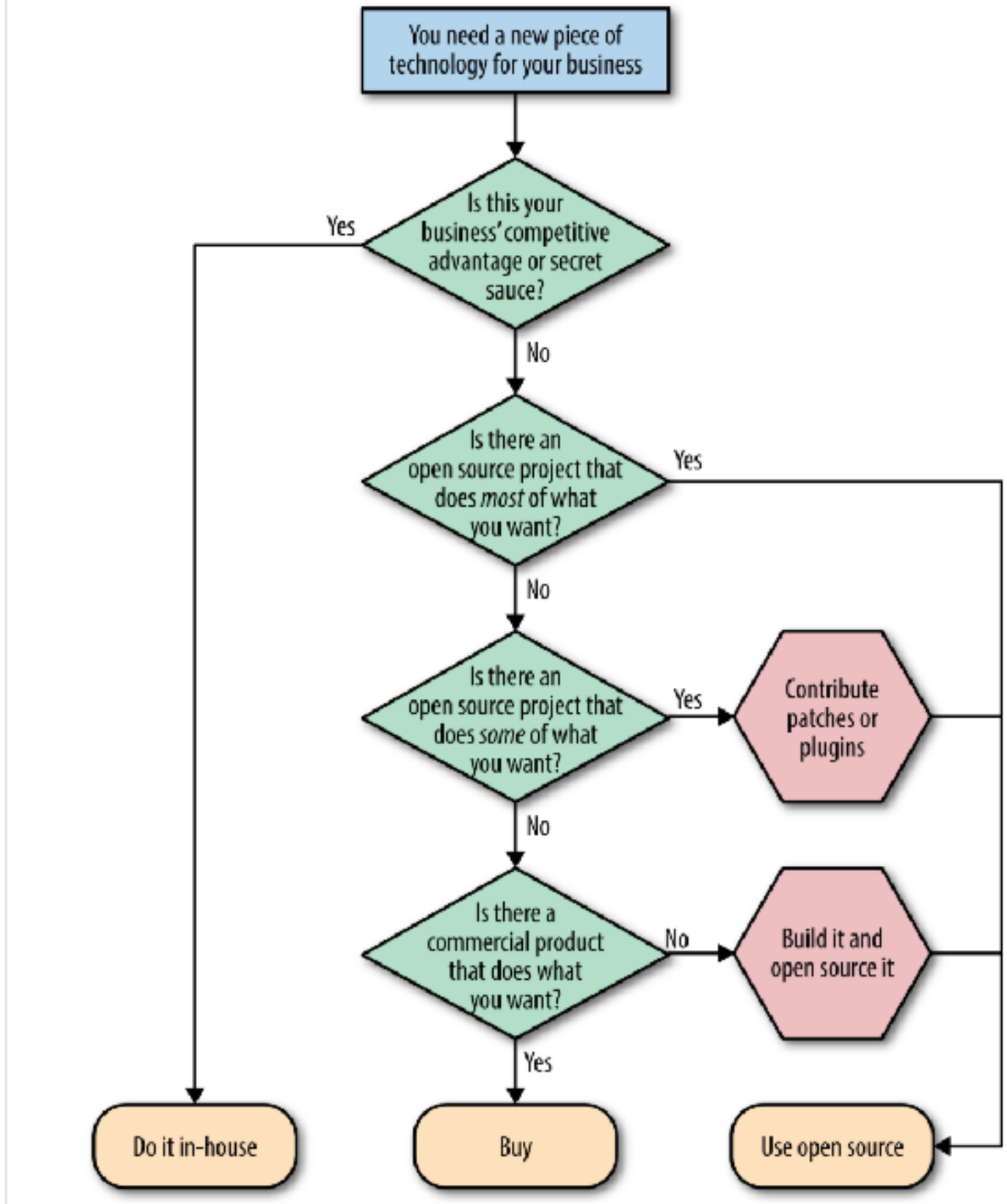
Considerations

- License (open source?)
    - cost, vendor lock-in
- Developer-friendly (i.e. happy)
    - development speed
    - CLI and API?
- Learning Curve
- Community and Ecosystem
    - documentation, library
- Performance

Don't reinvent the wheel

這些東西不要自己蓋

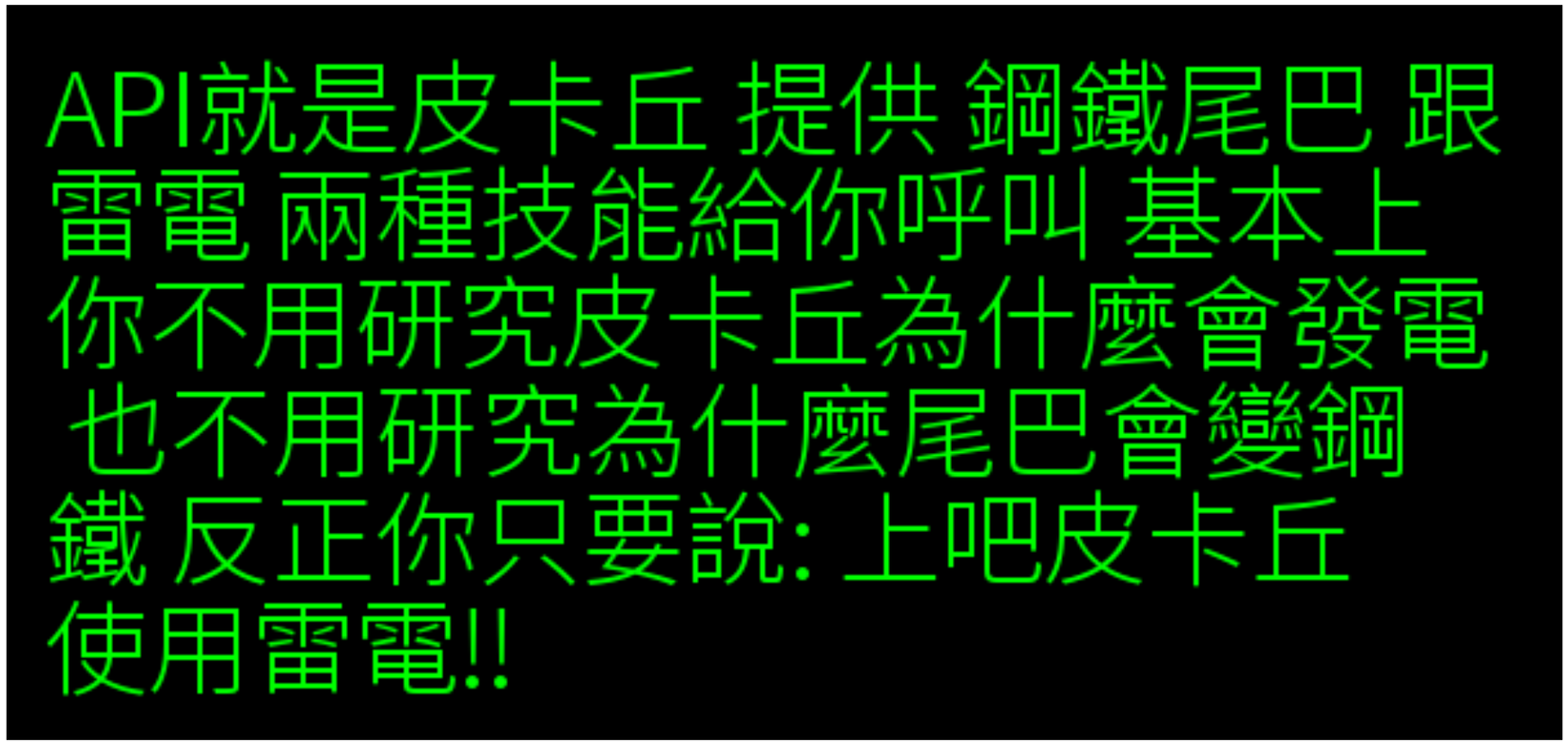
- Operating systems
- Programming languages
- CS 101: basic data structures (map, list, set), sorting algorithms
- Web technologies: HTTP servers, server-side and client-side frameworks
- Data systems: databases, NoSQL stores, caches, message queues
- Software delivery: version control, build systems, deployment automation
- Libraries for common data formats: XML, HTML, CSV, JSON, URLs
- Utility libraries: date/time manipulation, string manipulation, logging
- Secure Libraries: cryptography, password storage, credit card storage
- Operations/Office software: SaaS (google, gmail, slack, quip, dropbox…etc)


You need a new piece of technology for your business
Is this your business' competitive advantage or secret sauce?
Yes
No
Is there an open source project that does *most* of what you want?
Yes
No
Is there an open source project that does *some* of what you want?
Yes
Contribute patches or plugins
No
Is there a commercial product that does what you want?
No
Build it and open source it
Yes
Do it in-house
Buy
Use open source

Choose Programming Language

- iOS 開發: Swift or Objective-C
- Android 開發: Java
- Web 前端: JavaScript
- Web 後端: Ruby? PHP? Python? Java? .NET?

Use Library 和 Framework



https://www.facebook.com/kobeengineer/photos/a.1634720690097535.1073741828.1632027893700148/1697488120487458/?type=3

Choose library/framework

- Web 前端
  - jQuery
  - Bootstrap
  - ReactJS
  - AngularJS
- Web 後端
  - Ruby on Rails
  - Python: Django
  - PHP: Laravel

Choose Database

- Open Source
  - MySQL
  - PostgreSQL
  - NoSQL
- Commercial
  - Oracle
  - MSSQL

Domain Name 網域名稱

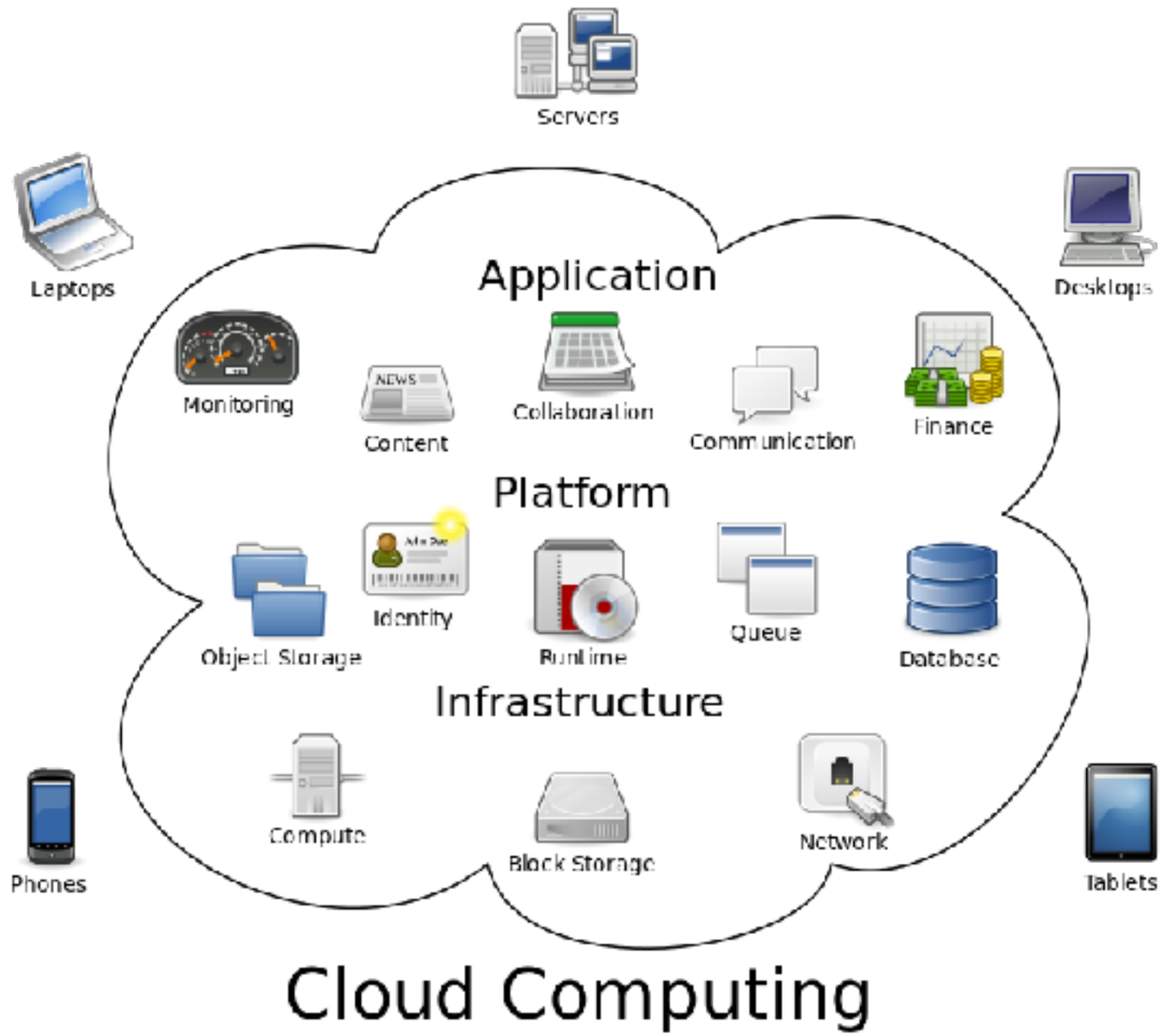
- 買一個順眼的 domain name，例如 alphacamp.co
  - https://www.namecheap.com
  - http://godaddy.com/
    - 除此之外還有很多域名營運商
    - 設定你的 DNS 將 domain name 對應到伺服器的 IP Address
- 一般使用者通常會使用 ISP 的 DNS，例如中華電信 168.95.1.1，或自己改成 Google 的 8.8.8.8。這些 DNS 會去查詢你的 DNS 獲得答案

情境

- 誰註冊了這個 Domain Name: Whois
- 標售哄抬現象 http://www.inside.com.tw/2011/04/07/domain-name-front-runnng
- 網路好像故障了，瀏覽器無法連線，但是可以 ping ip address 這是什麼情況? 可能是中華電信的 DNS 壞了。
- 牆內的 DNS 污染
- 釣魚網站: 利用網址很像、畫面弄成一樣騙你輸入帳號密碼

Choose Hosting

- BaaS
  - Firebase
- PaaS
  - Heroku
- IaaS
  - Amazon Web Services
  - Google
  - Microsoft Azure
  - Linode or Digital Ocean



CMS

- Content Management System 是一種現成的 Web 軟體，提供後台可以編輯 HTML/CSS，內容存進資料庫。

- Wordpress 或 Drupal

- 不需要改 HTML code 就可以編輯頁面，而且有很多 theme 可以裝

- 雖然有很多 plugins 可以裝，但是需要特別客製化就有困難

Our Web Technology Stack

Type	Name
Server OS	Ubuntu Linux
Dev OS	Mac OSX
IAAS	Linode and AWS
PAAS	Heroku
Shell	bash
Text Editor	Sublime Text and Vim
DVCS	Git
DVCS Hosting	Github
Server-Side Language	Ruby
Client-Side Language	JavaScript
Backend Web Framework	Ruby on Rails
Database	MySQL or PostgreSQL
Web Server	Nginx
Dev Browser	Chrome
Frontend CSS Framework	Bootstrap
Frontend Javascript Library	jQuery
Web Standards	HTML5/CSS3/HTTP2/JSON

5. Developer Roles and Skills you need

角色與技能

- Startup 團隊中，不是先有決定有那些角色位置，而是先有那些事要做，誰會那個技能誰就去作。

- 一個人是多技能的，前端、後端等不同角色在不同團隊中，自然有不同的工作涵蓋。

先看做出一個網站產品有哪些事情?

- 定義和決定需求、規格
- 網頁視覺設計
- Client-side 程式設計 e.g 頁面互動
- Server-side 程式設計 e.g. 資料庫存儲
- 測試
- 伺服器安裝、佈署、監控和管理
- 使用者行為分析、資料分析

一個人會什麼技能？

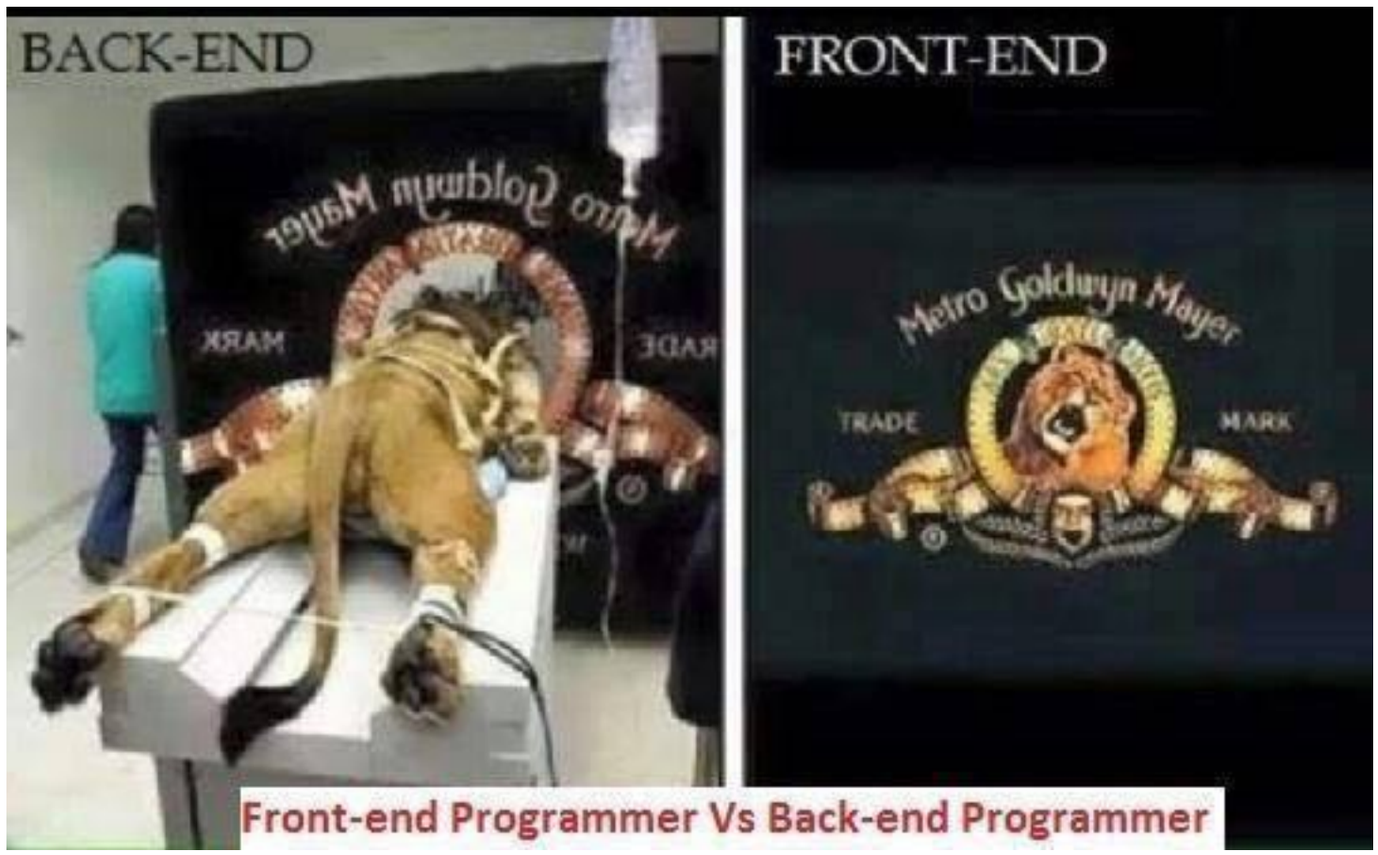
- 技能樹概念
  - http://www.dungeonsanddevelopers.com/
- http://blog.udacity.com/2014/12/front-end-vs-back-end-vs-full-stack-web-developers.html

定義 Engineering Team 的角色

- Product Manager or Product Owner
- Back-end developer
- Front-end developer
- Graphic designer
- System administrator
    - DBA
- QA/TE (Test Engineer)
- SET (Software engineer in Test)
- Experts: Security, UX, SEO
- Data scientist: 統計分析、視覺化、大數據存儲、機器學習

Full-stack Web developer

- Front-end v.s. Back-end
  - 有多 full-stack? 前一頁的技能都會嗎?
- buzzword?



Back-End (Rails) Developer

- Back-End 的程式語言很多...
- 熟悉 Ruby 程式語言
- 熟悉 Ruby on Rails 框架
- 熟悉資料庫和 SQL
- Linux 伺服器管理

Front-End Developer

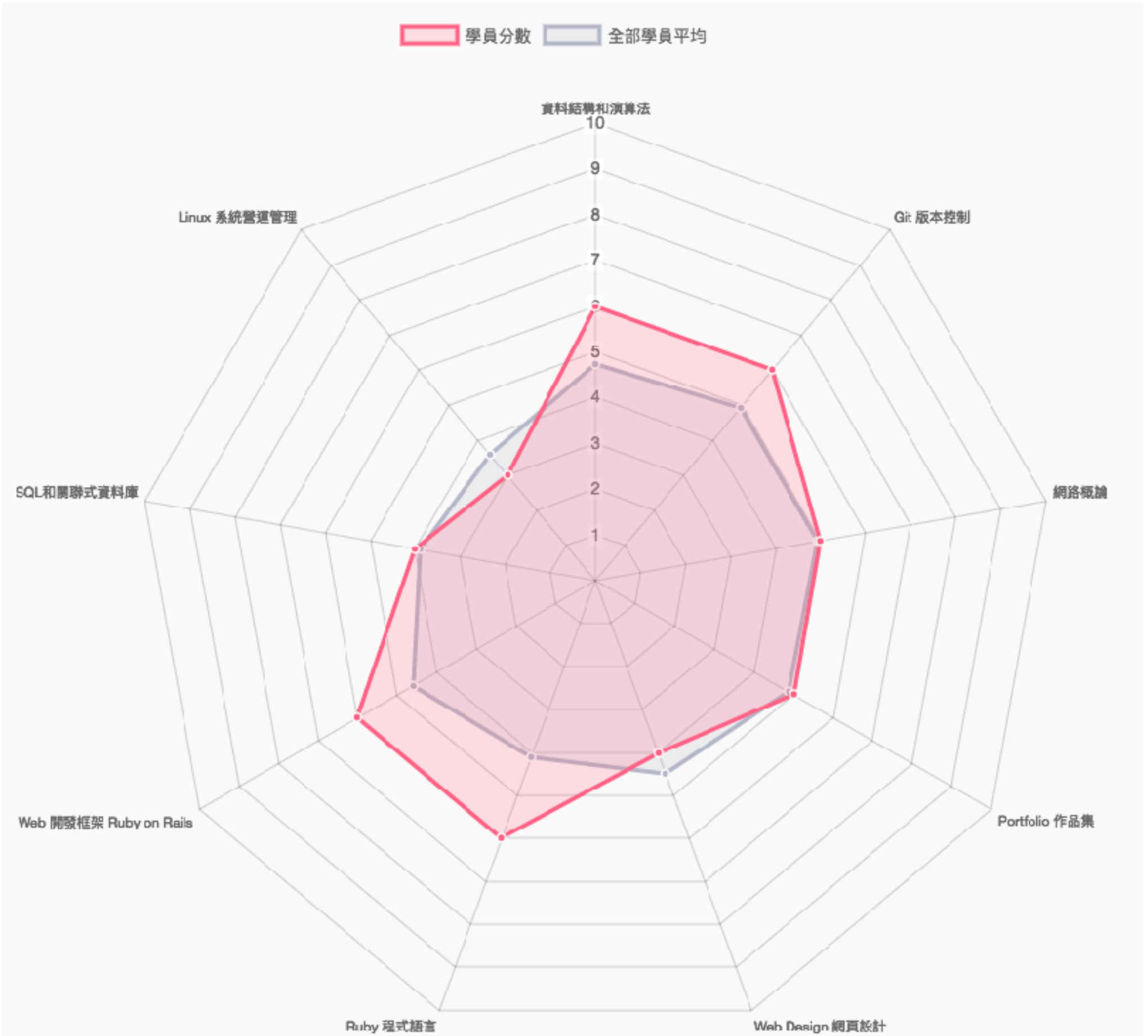
- F2E (Front-End Engineer)
- 熟悉 HTML/CSS
- 熟悉 JavaScript
- SPA (Single Page Application)
- UX/UI

Web Designer 和 Front-End developer

- 拆版誰拆? 誰會 CSS?
- Designer 只會視覺設計，不會 CSS?
- Front-End developer 不會視覺設計
- 不拆版的 Front-End developer?

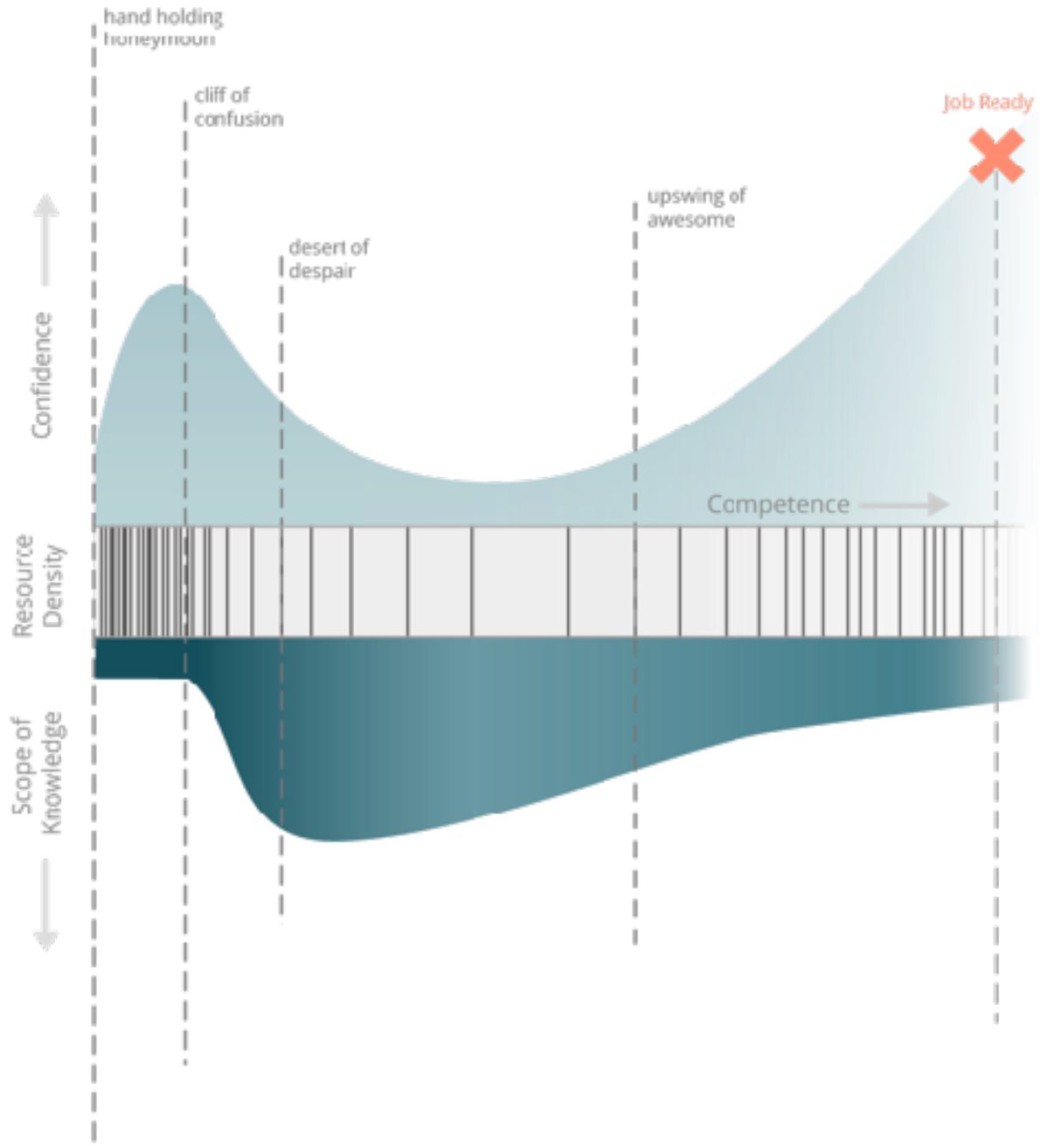
DevOps

- Developer + Operations
- Yet another buzzword?
  - System Administrator
  - Site Reliability Engineer


學員分數
全部學員平均
資料結構和演算法
Git 版本控制
網路概論
Portfolio 作品集
Web Design 網頁設計
Ruby 程式語言
Web 開發框架 Ruby on Rails
SQL和關聯式資料庫
Linux 系統營運管理
10
9
8
7
5
4
3
2
1

Some Advice

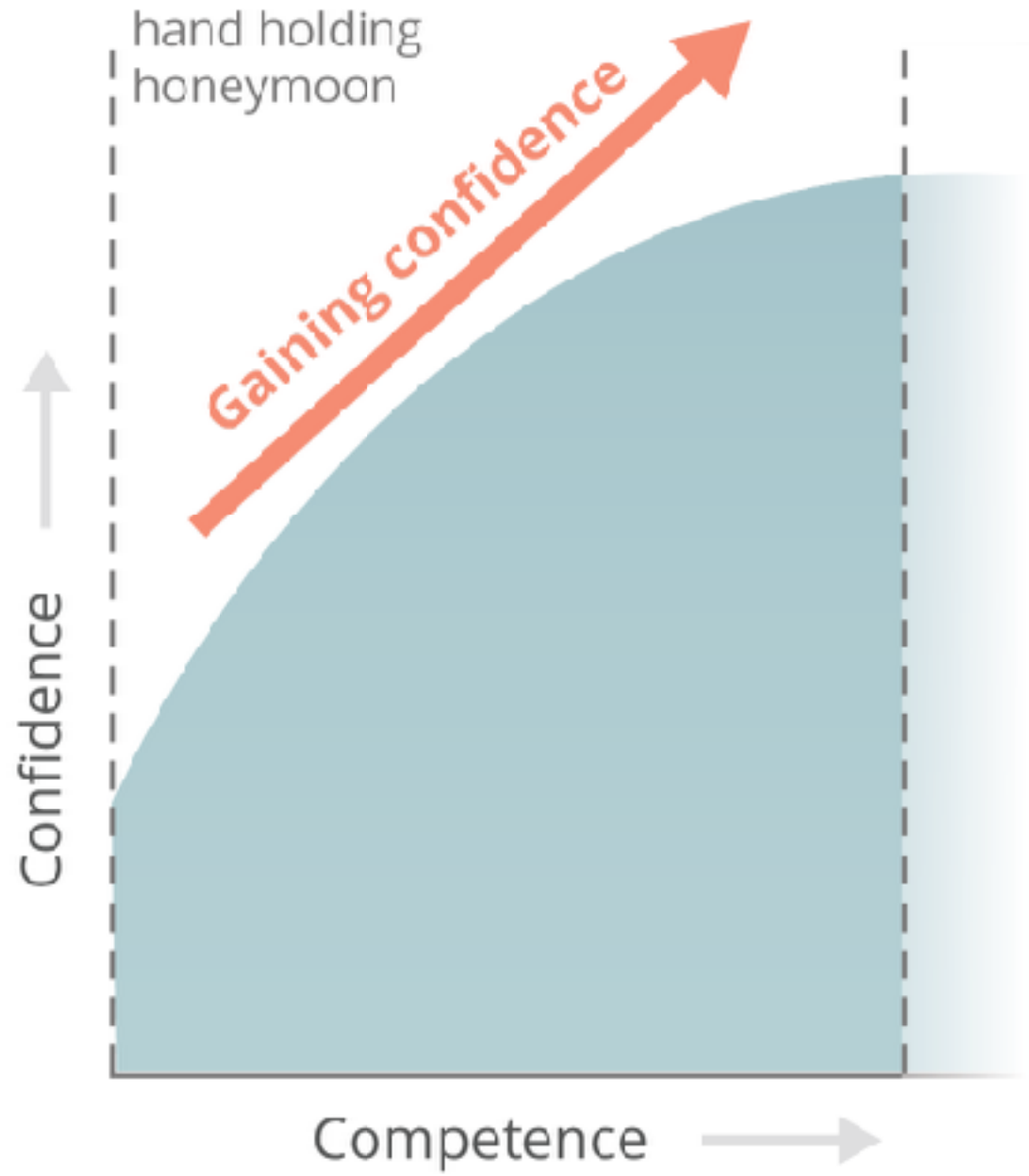
為什麼成為一名工程師這麼難



http://www.inside.com.tw/2015/03/27/why-learning-to-code-is-so-damn-hard

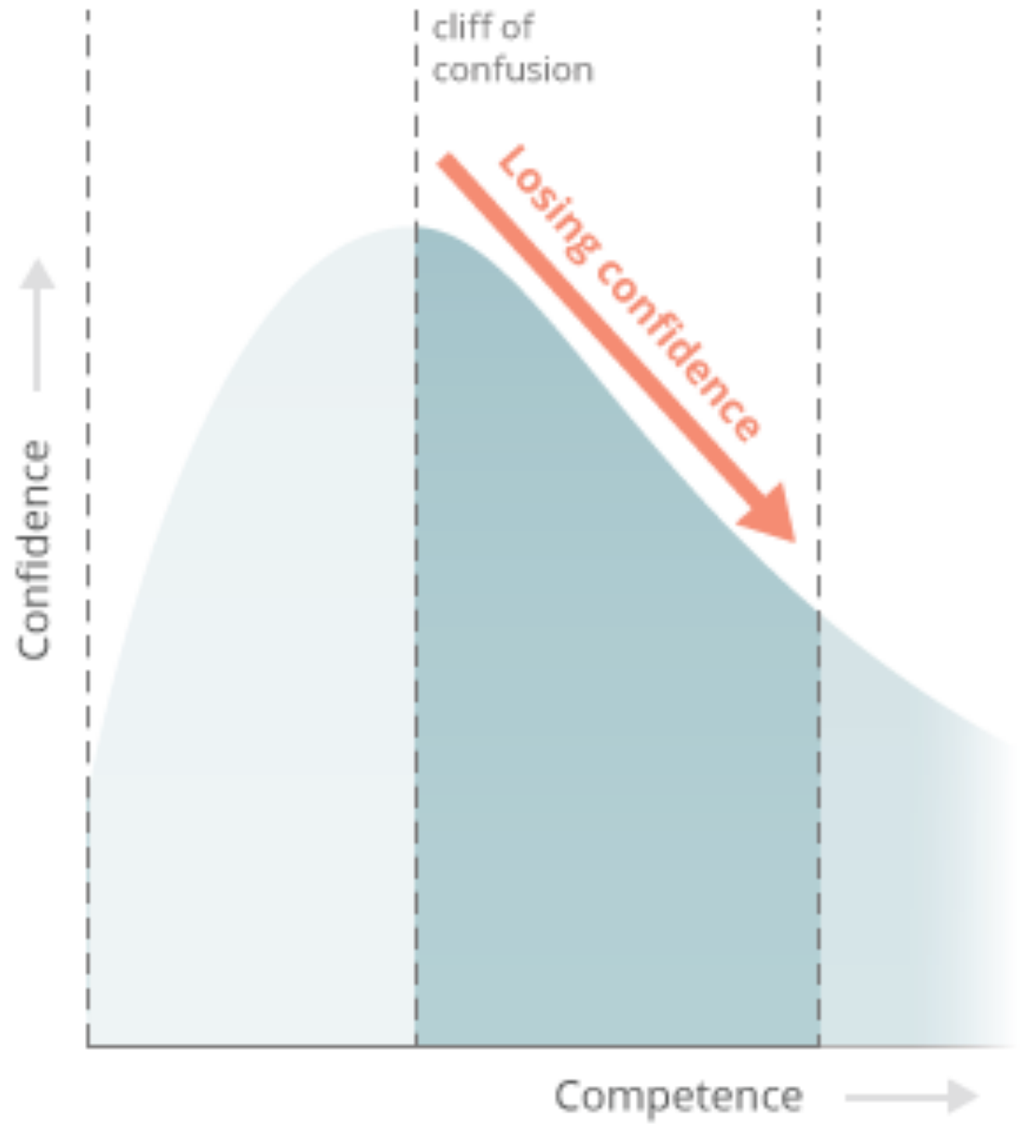
第一階段：手牽手心連心蜜月期

- Week 3~4
- 照著教材做就做出來了
- 坊間短期課程大多如此



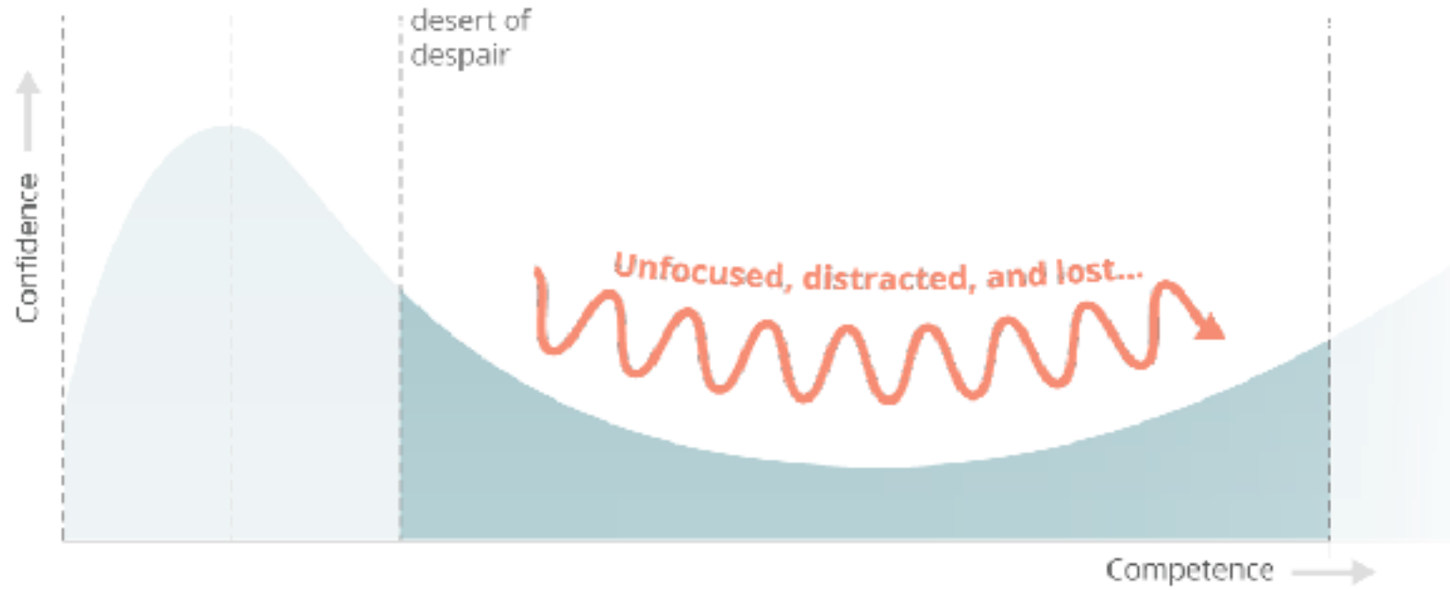
第二階段：困惑之崖

- Week 6~8
- 作業開始不會寫
- 不知道錯在哪裡
- 不知道怎麼 Debug 除錯
- 不知道該問什麼問題
- 決定進入這行的轉捩點



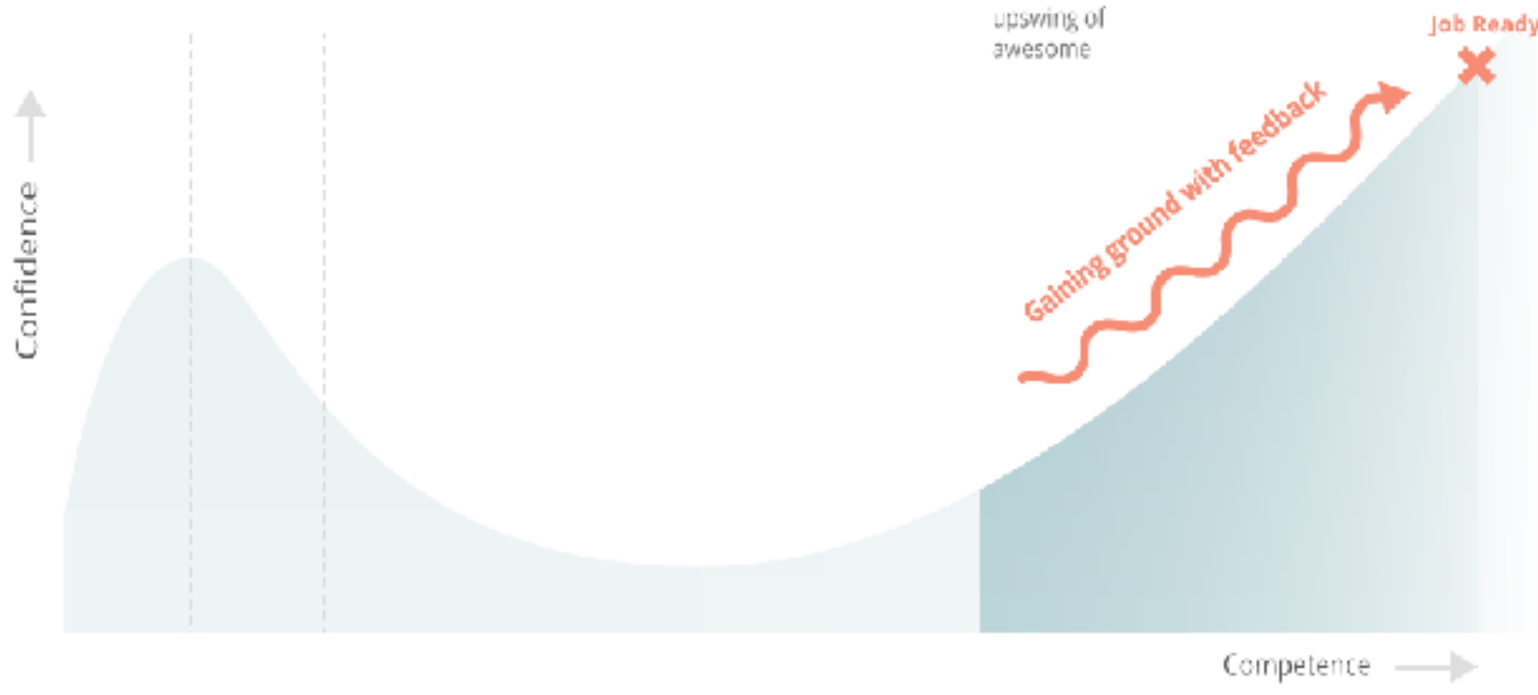
第三階段：絕望沙洲

- Week 9~10
- 你不知道你什麼不知道
  You don't know what you don't know
- 不知道要 google 什麼
- 學也學不完



第四階段：創傷後的恢復期

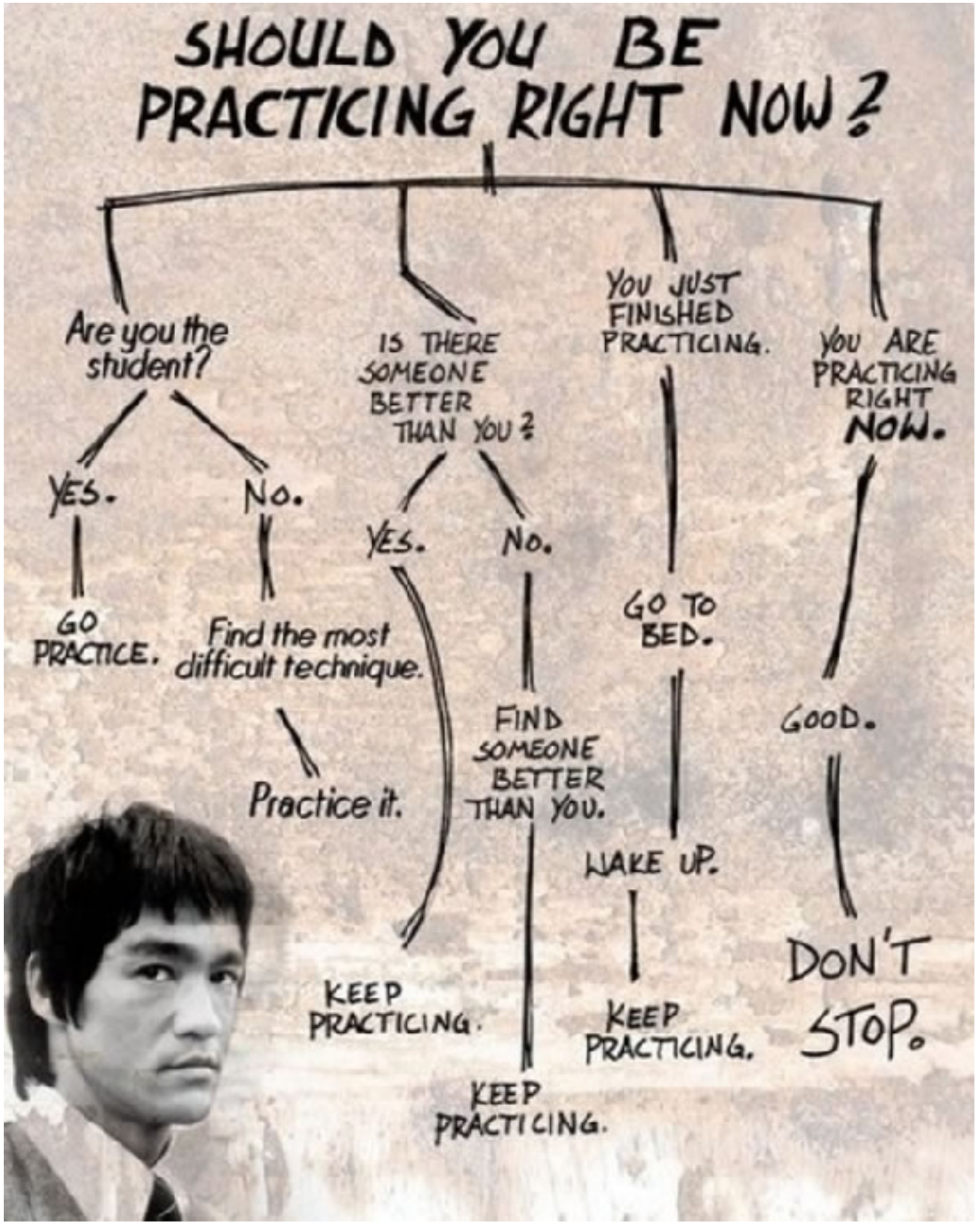
- Week X
- 開始有自信心了
- 知道該 google 什麼
- 選定自己的技能樹



軟體工程師的鄙視鏈？

https://vinta.ws/blog/695

- 萬丈高樓，起於平地
- 千里之行，始於足下



https://twitter.com/AccordionGuy/status/538736213071384577

總結和課後問題

- 開發不同類型的應用，需要不同的程式語言
- 不同 Platforms 的不同特性
  - (Native, cross-compile, hybrid, Web) x (desktop, mobile)
  - 你會如何選擇用 Web 或 Mobile 開發你的應用?
    - Web Sites 和 Web Applications 有什麼不一樣?
    - Web 應用和 Mobile 應用有哪些使用者體驗(UX)上的差異? 如果你有一個 idea，你會如何選擇用 Web 或 Mobile 應用?
- 該如何區分前端和後端開發? 請以 mobile app 和 web app 舉例。
  - 為前端、後端開發者分別強調熟悉那些技能?
- Tech stack 的選擇
- 有哪些 developer role?

課後問題

- Web Sites 和 Web Applications 有什麼不一樣?
- 該如何區分前端和後端開發? 請以 mobile app 和 web app 舉例。
- 你認為前端、後端開發者分別強調熟悉那些技能?
- Web 應用和 Mobile 應用有哪些使用者體驗(UX)上的差異? 如果你有一個 idea，你會如何選擇用 Web 或 Mobile 應用?

參考資料

- https://www.coursera.org/course/startup
- https://playbook.thoughtbot.com/
- Hello, Startup (O'Reilly)
- Product Design for the Web, Randy J. Hunt
- 不再聽不懂！圖解網站建置與開發, 碁峰