

Ruby on Rails

thegiive in COSCUP
Taiwan Ruby User Group

(License by CC 姓名標示 2.0)

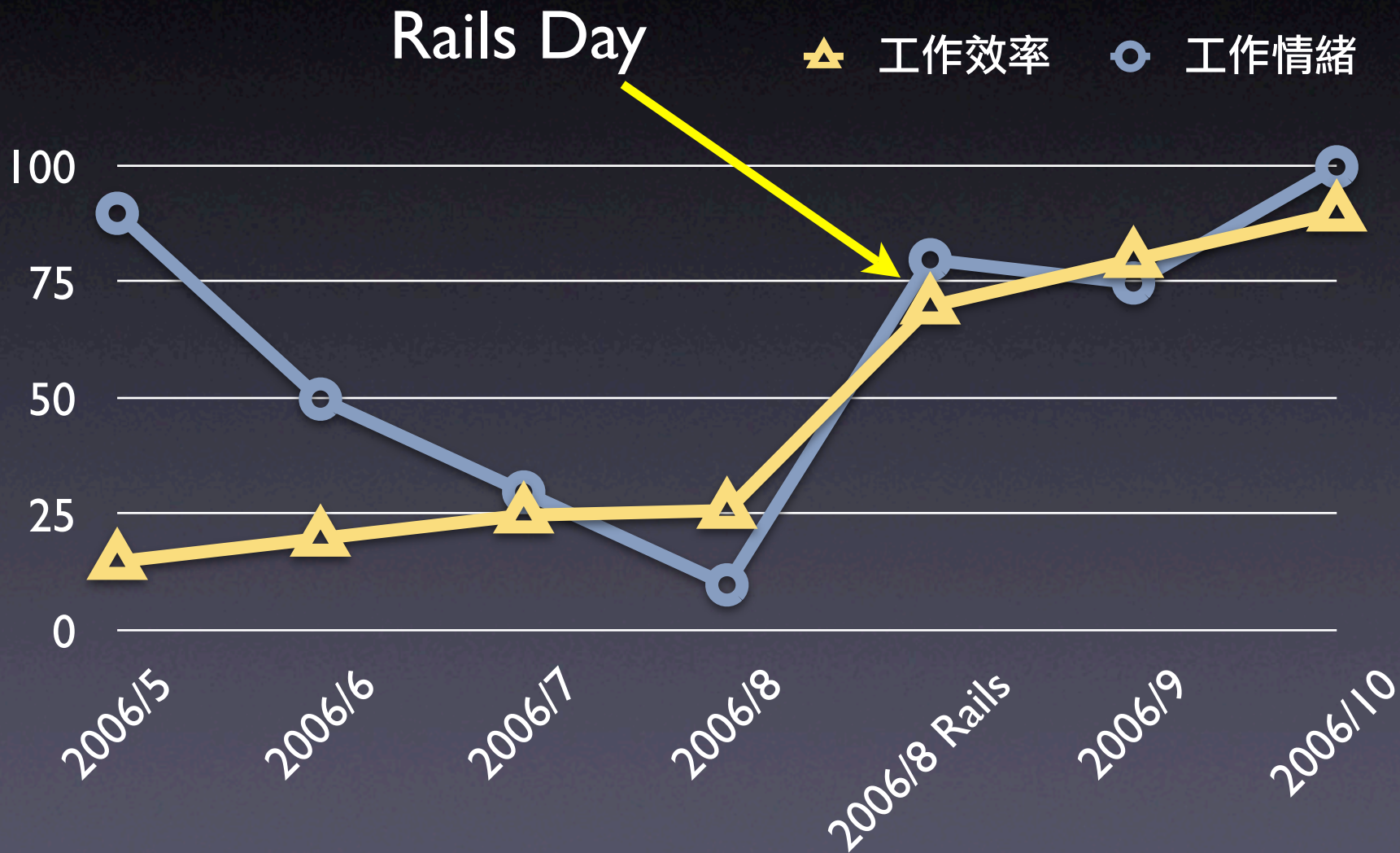
About me

- 網路 ID : thegiive
- 現任某網路公司 RD
- 使用 Ruby on Rails 開發網頁程式
- Blog : <http://lightyror.blogspot.com/>
- Email : thegiive at gmail . com

Outline

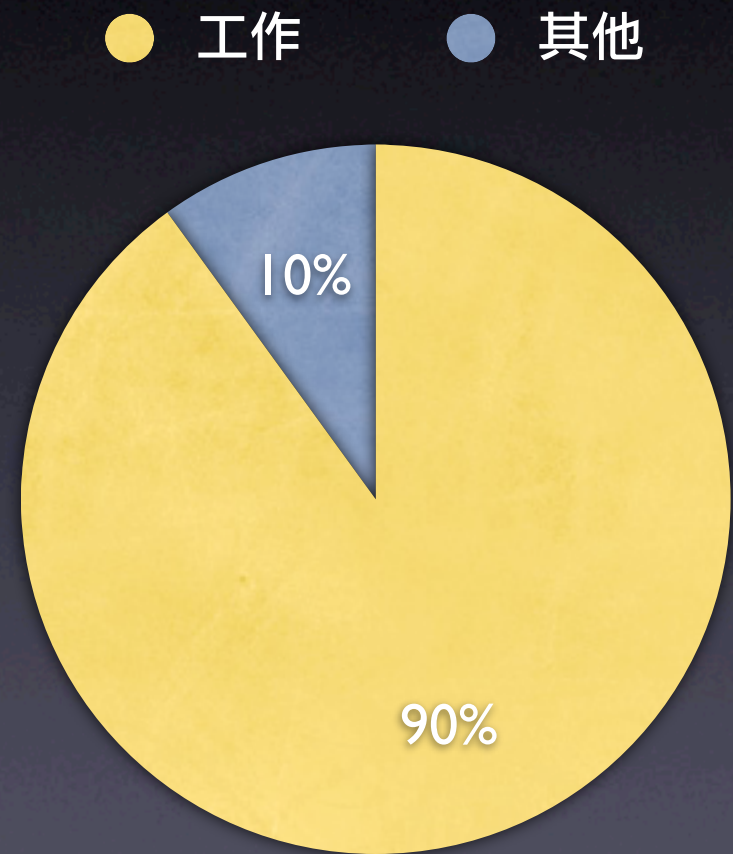
- 關於我使用 Ruby on Rails 的故事
- Ruby on Rails 簡介
- Rails 高生產力的來源
- Active Record 火力展示
- 一些議題
- Active Record 進階火力展示

關於效率跟情緒的故事



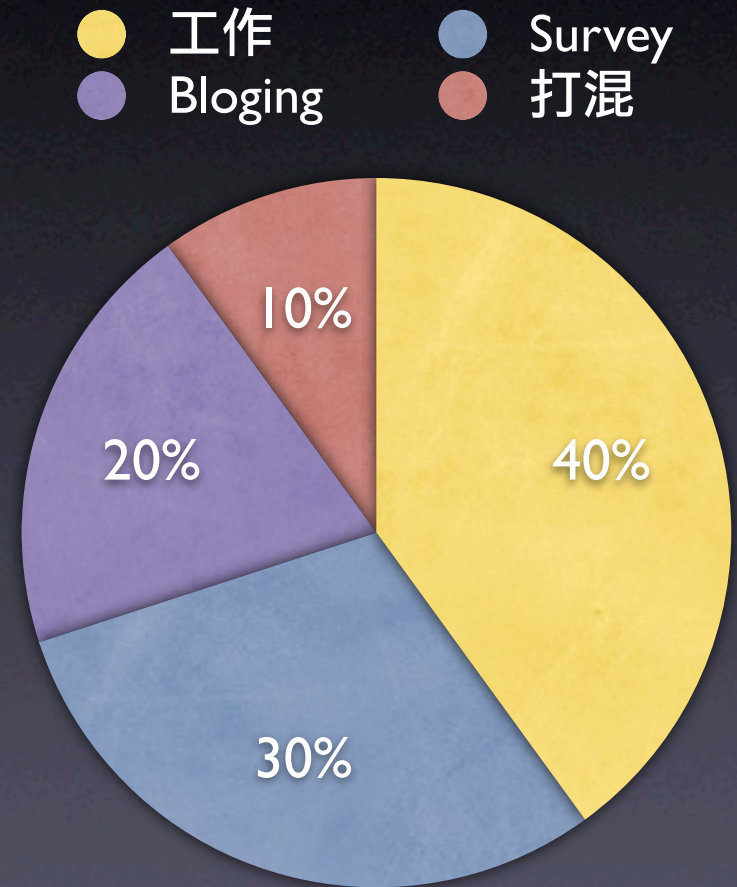
Before Rails day

- 使用 PHP
- 每天都被催進度
- 上班時間被工作佔滿



After Rails day

- 使用 **2 天** 的時間改寫了一個用PHP寫了兩週的 Project
- 換成我在催美工進度
- 小型頁面平均一兩個小時可以完成



Phone time project

- 完成了“Blog換裝，國片啟航”的後端程式
- 撰寫約 20 行 code
- 讀取 3 個資料表



Blog換裝·國片啟航

http://events.roodo.com/twmovie

COSCUP/Program - TOS... conclusion fy - Google... Blog換裝·國片啟航

Roodo! Your Blog 國片啟航

盛夏光年
ETERNAL SUMMER

友情·團結
真實最感動
你一定愛上這片!!

9.22 最佳電影 /
奇蹟的夏天
My Football Summer More Than Just a Game

楊力州 張榮吉 導演

醫生
DOCTOR

現在就幫自己的BLOG換裝，表達對國片的支持！還有機會得到特映券或其他贈品

活動說明

你熱愛國片嗎？你是否對國片有深切的期待與熱忱，今年夏末，在此為你獻上三部不同類型的國片強作！
從現在起，登入Blog管理，至「網頁樣式設定」的地方，在上列三齣電影（盛夏光年、奇蹟的夏天、醫生）共四套的國片樣式裡，選擇其中一種套用在自己的blog上，就有機會得到相關贈品唷！

- 奇蹟的夏天 楊力州導演親筆簽名明信片30張
- 醫生 特映券30張
- 盛夏光年 特映券30張

(以上獎項除醫生特映券將每日抽出並立即通知得獎者之外，其餘獎項將於9/30統一抽出並公告，並公告於“因為有您，國片起飛”部落格)

Ruby on Rails

- MVC 框架
- 以 **Ruby** 為主要語言
- DHH 為了 Basecamp 這個 Project 而開發出來



創造者

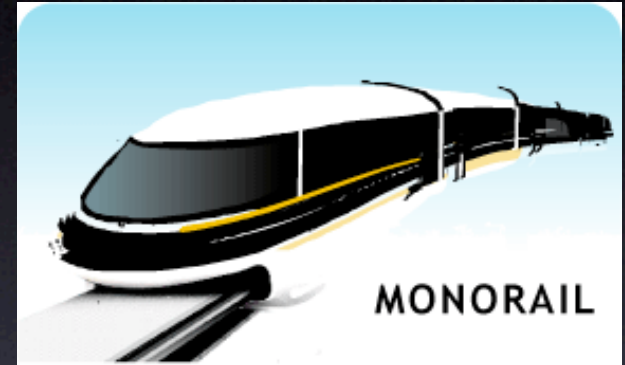
- David Heinemeier Hansson
- 網路簡稱 DHH
- 當時才 26 歲



圖片出自DHH Blog

How Hot is Rails

- Rails 2006研討會 400個座位一周內賣完，加開的 150個也在 24小時內賣完
- 2005年 Ruby on Rails 書籍銷售成長 1500%，2006年又成長了 700%
- Ruby 佔有率一年內從 26名暴增到13名



Rails Clone



席捲全球的原因

- **超快速**網頁程式開發

開發速度有多快？

| | JAVA (Spring/Hibernate) | Rails |
|--------------|-------------------------|-------------|
| 時間 | 4 個月 (20小時/周) | 4 晚 (5小時/晚) |
| 程式碼 | 3293 | 1164 |
| 設定檔 | 1161 | 113 |
| Class/method | 62/549 | 55/126 |

Justin Gehtland 在他的Blog發表的數據

快速開發到底有多重要

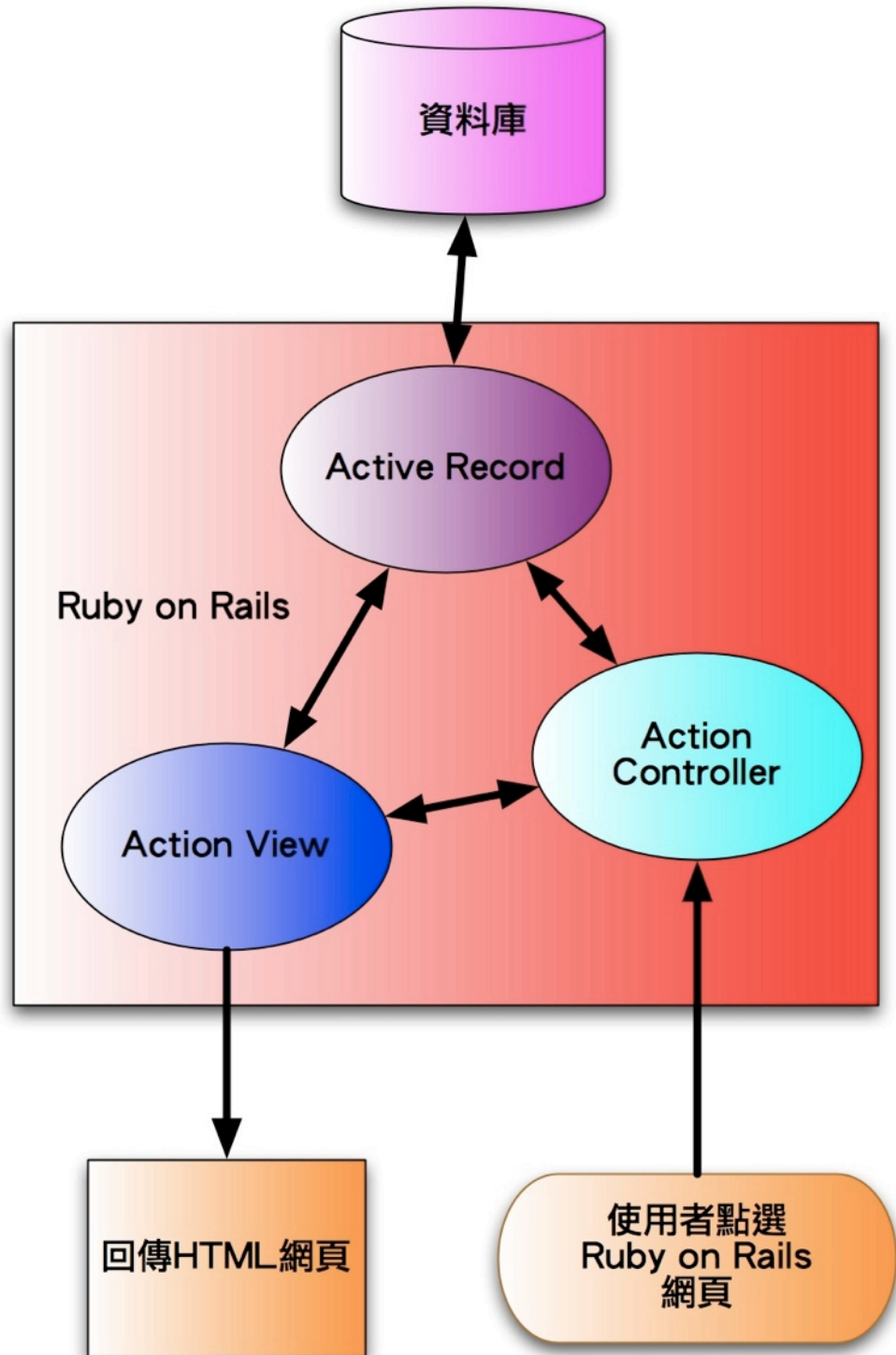
- 有Performance之後，要錢就比較容易！！
- 你可以有時間打混，而且老闆還會稱讚你
- 可以早點下班出去玩

程式碼大量減少

- Justin Gehtland 提出 JAVA : Rails = 3.5 : 1
- Proc.net 提出 PHP : Rails = 10 : 1
- JavaEye 站長估計 JAVA : Rails = 10 : 1
- 我實驗的結果 PHP : Rails = 8 : 1

網址 <http://lightyror.blogspot.com/2006/10/rails.html>

Rails 架構



高速開發的原因

- ORM
- Convention over Configuration
- Don't Repeat Yourself
- Domain Specific Language

ORM

- Object **R**elational **M**apping
- 將資料庫操作作用物件導向的形式來呈現
- Propel ， Hibernate ， Active Record ... 等

Convention over Configuration

- 不需要額外設定檔，一切以預設值加快研發速度
- 但是總是有辦法可以更改預設的方式

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE hibernate-mapping  
PUBLIC "-//Hibernate/Hibernate Mapping DTD//EN"  
"http://hibernate.sourceforge.net/hibernate-mapping-2.0.dtd"  
>  
<hibernate-mapping>  
<class name="test.hibernate.Order" table="orders">  
<id name="id" type="string" unsaved-value="null" >  
<column name="id" sql-type="integer" not-null="true" />  
<generator class="native" />  
</id>  
<key column="order_id" />  
<one-to-many class="test.hibernate.OrderItem" />  
</set>  
</class>  
</hibernate-mapping>
```

DRY

- Don't Repeat Yourself
- View 裡面有 Helper , Partial 方便 DRY
- Controller 裡面有 Module , Lib 等方便 DRY

Domain Specific Language

- Ruby 是一個很好的 DSL 製造語言
- Rails 是 DB-Driven Web App 的 DSL
 - `has_one` , `has_many` , HABTM
 - 大量使用 symbol (`:abc`)

Active Record

- Rails 的**心臟**
- 目前數一數二的 ORM 技術
- Convention over Configuration
- 可以處理 1:1 ， 1:m ， n:m 等資料庫關係
- 支援 MySQL ， SQL Server ， Oracle ， SQLite 等資料庫

ORM Example

| People | |
|---------|---------|
| id | int |
| name | varchar |
| phone | varchar |
| address | text |
| sex | enum |



```
a = People.new
```

```
a = People.find(1)
```

```
a.name = 'David'
```

```
a.destroy
```

```
a.save
```

Relation Table Example

| People | |
|---------|---------|
| id | int |
| name | varchar |
| phone | varchar |
| address | text |
| sex | enum |

| Article | |
|------------|----------|
| id | int |
| person_id | int |
| title | varchar |
| content | text |
| created_at | datetime |



Relation ORM

`a = People.find(1)`

`a.articles`

代表這個人寫的所有文章列表

`a.articles.find(:first)`

這個人寫的第一篇文章

`a.articles << Article.new(...)`

Person a 新寫一篇文章

缺點？

- 速度
- 國際化
- Legacy系統的支援
- Template 支援度

大家都會這樣講

“ Ruby on Rails 是好東西，可惜他不是 xxx 寫的 ” 這句話我至少聽過 JAVA，Python，Perl 三個版本

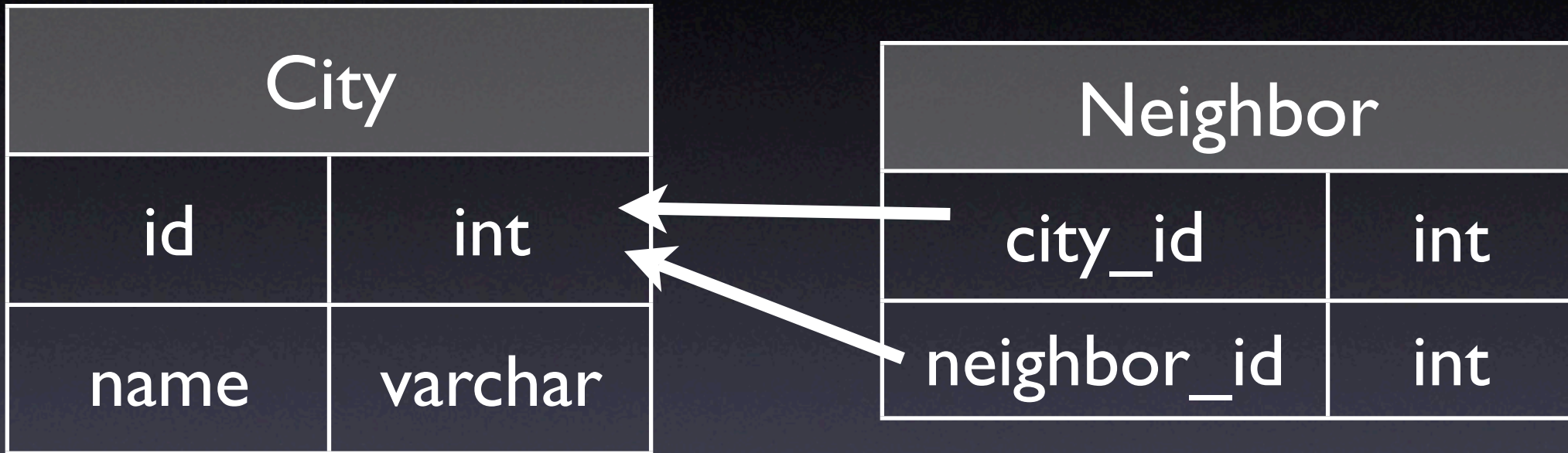
其他語言可以作出 Rails 嗎？

- 任何事情都有可能.....**但**.....
 - DHH 曾經解釋使用 Ruby 原因
 - Ruby 的 DSL 能力
 - 我曾經使用其他框架的體驗
 - 程式碼完成度，文件，參與人數多寡

“如果一個框架可以讓你僅變快 20%，或許你應該繼續使用比較保險的語言，像是 JAVA。但是如果你可以變快 300%，甚至更高，那麼其他的差異都變得不重要了。”

From Beyond Java

小習題



程式會給兩個城市，目的是判斷“**隔壁**”，
”**鄰近**“還是”**很遠**“？

難度

- city 跟自己是呈現自我 n:m 關係
(neighbor 是 join table)
- 要找到兩層以上的深度
- 要寫的容易閱讀

```
function 判斷 (A城市，B城市) {  
    if( B城市是A城市的鄰居) {  
        return '隔壁'  
    }else{  
        for ( A城市的每個鄰居城市，設為城市C) {  
            if( B城市是C城市的鄰居 ) {  
                return '鄰近'  
            }  
        }  
        return '很遠'  
    }  
}
```

Pseudo Code


```
def check ( city_a , city_b )
  if ( city_a.neighbors.include?(city_b)
    return '隔壁'
  else
    city_a.neighbors.each do | neighbor_of_a |
      if (neighbor_of_a.neighbors.include?(city_b) )
        return '鄰近'
      end
    end
    return '很遠'
  end
end
```

Rails Code