


BUILD RUBY ON RAILS APPLICATION PROTOTYPING LESSION 2



開始之前...

- » 儘量發問
- » 回去繼續要多練習
- » 補充資料
 - » <https://railsbook.tw/>
 - » 第 05 ~ 08 章

主題

» Ruby for Rails

RUBY FOR RAILS

- » Ruby 跟你學過的程式語言不太一樣...
- » 不會介紹所有 Ruby 相關的知識
- » 夠用就好

目標

» 完成今天的課程，你應該可以開始「看得懂」 Rails 專案的程式碼

RUBY FOR RAILS

- » Ruby 是一種物件導向程式語言(Object-Oriented Programming, OOP)
- » 在 Ruby 的世界裡，所有的東西都是物件...幾乎啦！
- » Rails 不是一種程式語言(重要！)

RUBY FOR RAILS

HELLO, WORLD

```
puts "Hello, World"
```

RUBY FOR RAILS

HELLO, WORLD

- » `$ ruby -e "puts 'hello, world'"` (要注意單引號跟雙引號)
- » 在 `irb` 裡練習
- » 寫在某個檔案裡
 - » 例如檔名是 `abc.rb` (寫完別忘了存檔)
 - » 請 Ruby 來執行它：`$ ruby abc.rb`

RUBY FOR RAILS

常數與變數

» 變數有好幾款

區域變數

全域變數

實體變數

類別變數

username

\$username

@username

@@username

RUBY FOR RAILS

常數與變數

- » 為什麼要使用變數？
- » 變數本身沒有型態
- » 變數命名慣例

RUBY FOR RAILS

常數與變數

» 變數指定

```
a = 1
```

```
x, y, z = [1, 2, 3]
```

RUBY FOR RAILS

常數與變數

» 常數 Constant，大寫字母開頭

```
BOOK = "ruby book"
```

```
User = "hello, user"
```

RUBY FOR RAILS

常數與變數

» 所以，常數跟變數有什麼不同？

1. 命名上，常數是大寫字母開頭，變數不是。
2. 變數的值可以改變，但常數的值不能改變(?)

RUBY FOR RAILS

字串安插(以 PHP 為例)

» 在 PHP 要串接字串可能是這樣寫：

```
$name = "kk";
```

```
$age = 18;
```

```
echo "hi, I am " . $name . ", and I am " . $age . " years old";
```

RUBY FOR RAILS

字串安插(RUBY)

» 在 Ruby 你可以這樣寫：

```
name = "kk"
```

```
age = 18
```

```
puts "hi, I am #{name}, and I am #{age} years old"
```

那如果這樣呢？

```
puts 'hi, I am #{name}, and I am #{age} years old'
```

RUBY FOR RAILS

邏輯判斷與流程控制

```
» if .. else ..
```

```
weather = "下雨"
```

```
if weather == "下雨"
```

```
  puts "宅在家裡"
```

```
elsif weather == "出太陽"
```

```
  puts "出去玩!"
```

```
else
```

```
  puts "睡覺!"
```

```
end
```


RUBY FOR RAILS

邏輯判斷與流程控制

- » 只有 `false` 跟 `nil` 是假的
- » 注意：是 `elsif`，不是 `elseif`

RUBY FOR RAILS

邏輯判斷與流程控制

» unless == if not

```
weather = "出太陽"
```

```
if not weather == "下雨"  
  puts "太好了，我們出去玩!"  
end
```

與以下程式相等

```
unless weather == "下雨"  
  puts "太好了，我們出去玩!"  
end
```

RUBY FOR RAILS

邏輯判斷與流程控制

» 把 `if` 丟後面去

```
weather = "出太陽"
```

```
if weather == "出太陽"  
  puts "太好了，我們出去玩!"  
end
```

與以下程式相等

```
puts "太好了，我們出去玩!" if weather == "出太陽"
```

RUBY FOR RAILS

邏輯判斷與流程控制

» case ... when ...

```
weather = "下雨"
```

```
case weather
when "下雨"
  puts "宅在家裡"
when "出太陽"
  puts "出去玩!"
else
  puts "睡覺!"
end
```

RUBY FOR RAILS

邏輯判斷與流程控制

» 原來一堆的 `if .. elsif ...`

```
age = 10
```

```
if age > 0 && age <= 3
  puts "寶寶"
elsif age > 3 && age <= 10
  puts "小孩"
elsif age > 10 && age <= 17
  puts "青少年"
else
  puts "成人"
end
```

RUBY FOR RAILS

邏輯判斷與流程控制

» 可用 `case ... when ...` 改寫

```
age = 10
```

```
case
when age > 0 && age <= 3
  puts "寶寶"
when age > 3 && age <= 10
  puts "小孩"
when age > 10 && age <= 17
  puts "青少年"
else
  puts "成人"
end
```

RUBY FOR RAILS

邏輯判斷與流程控制

» 甚至還可以結合 Range...

```
age = 10

case age
when 0..3
  puts "寶寶"
when 4..10
  puts "小孩"
when 11..17
  puts "青少年"
else
  puts "成人"
end
```

RUBY FOR RAILS

符號 SYMBOL

- » 冒號開頭，看起來有點像變數的東西
- » 這東西有點玄，但又常常在程式裡看到...
- » 符號 = 有名字的物件

```
class User < ApplicationRecord
  has_many :posts
end
```


RUBY FOR RAILS

註解 COMMENT

» 註解的目的是什麼？

這是單行註解

這是單行註解

這是單行註解

=begin

這是多行註解

要放幾行都可以

=end

RUBY FOR RAILS

集合

陣列 Array

```
friends = ["eddie", "jonh", "mary", "ema"]
```

雜湊 Hash

```
book = {title: "Ruby", price: 350 }
```

```
book = { :title => "Ruby", :price => 350 }
```

RUBY FOR RAILS

集合

Range

`(1..10)` `#=>` 1, 2, 3 ... 10

`(1...10)` `#=>` 1, 2, 3 ... 9

Range 轉 Array

`(1..5).to_a` `#=>` [1, 2, 3, 4, 5]

`[*1..5]` `#=>` [1, 2, 3, 4, 5]

RUBY FOR RAILS

迴圈與迭代 LOOP AND ITERATION

```
5.times do  
  puts "hello, ruby"  
end
```

```
1.upto(5) do |i|  
  puts "hi, ruby #{i}"  
end
```

RUBY FOR RAILS

迴圈與迭代 LOOP AND ITERATION

```
friends = ["eddie", "joanne", "john", "mark"]
```

```
for friend in friends  
  puts friend  
end
```

» friend / friends 單、複數？

RUBY FOR RAILS

迴圈與迭代 LOOP AND ITERATION

```
friends = ["eddie", "joanne", "john", "mark"]
```

```
friends.each do |friend|  
  puts friend  
end
```

RUBY FOR RAILS

迴圈與迭代 LOOP AND ITERATION

» 在 view 裡使用迭代的樣子

```
<% @posts.each do |post| %>
  <tr>
    <td><%= post.title %></td>
    <td><%= post.content %></td>
    <td><%= link_to 'Show', post %></td>
    <td><%= link_to 'Edit', edit_post_path(post) %></td>
    ... [略] ...
  </tr>
<% end %>
```

RUBY FOR RAILS

迴圈與迭代 LOOP AND ITERATION

» 練習題

» 請把陣列 [1, 2, 3, 4, 5] 轉換成 [2, 4, 6, 8, 10]。

» 提示：

1. 準備一個新的陣列，並且把 [1, 2, 3, 4, 5] 拿來轉圈圈

2. 轉的過程把每個元素 $\times 2$

3. $\times 2$ 之後的結果再丟進剛剛建立的陣列

RUBY FOR RAILS

迴圈與迭代 LOOP AND ITERATION

» 你是這樣寫嗎？

```
result = []
```

```
[1, 2, 3, 4, 5].each do |n|
```

```
  result << n * 2
```

```
end
```

```
p result
```

RUBY FOR RAILS

迴圈與迭代 LOOP AND ITERATION

» 但其實你可以這樣寫

```
p [1, 2, 3, 4, 5].map { |n| n * 2 }
```

RUBY FOR RAILS

迴圈與迭代 LOOP AND ITERATION

» 練習題

» 請計算從 1 加到 100 的總和。

RUBY FOR RAILS

迴圈與迭代 LOOP AND ITERATION

» 你是這樣寫嗎？

```
total = 0
(1..100).to_a.each do |i|
  total += i
end

puts total
```

RUBY FOR RAILS

迴圈與迭代 LOOP AND ITERATION

» 但其實你可以這樣寫

```
p [*1..100].reduce { |sum, n| sum + n }
```

或更簡短一點：

```
p [*1..100].reduce(&:+)
```

RUBY FOR RAILS

定義方法

- » 為什麼要定義「方法」？
- » 「方法(Method)」跟「函式(Function)」有什麼不一樣？

RUBY FOR RAILS

定義方法

使用 def 定義方法

```
def is_adult?(age)
  if age >= 18
    return true
  else
    return false
  end
end
```

```
if is_adult?(20)
  puts "成年人"
end
```

RUBY FOR RAILS

定義方法

- » "?" 跟 "!" 可以是方法名字的一部份，但只能放在最後一個字
- » "?" 方法通常會回傳 `true` 或 `false`，沒有其它
- » "!" 方法通常會有一些副作用〔使用前請詳閱說明書〕

RUBY FOR RAILS

定義方法

```
list = [1, 2, 3, 4, 5]
```

```
p list.reverse      # [5, 4, 3, 2, 1]
p list              # [1, 2, 3, 4, 5] 沒有改變

p list.reverse!     # [5, 4, 3, 2, 1]
p list              # [5, 4, 3, 2, 1] 已改變
```

RUBY FOR RAILS

定義方法

» 想一想，請問 `link_to` 這個方法，共有幾個參數？

```
<%= link_to "首頁", root_path, class:"btn btn-default", method: "post", confirm: true %>
```

RUBY FOR RAILS

定義方法

» 想一想...

```
age = 18
```

```
def age  
  20  
end
```

```
puts age    # 會得到什麼？
```

RUBY FOR RAILS

回傳值

» "return" 可以被省略

```
def is_adult?(age)
  if age >= 18
    true
  else
    false
  end
end
```

RUBY FOR RAILS

區塊 (BLOCK)

樣式 1

```
5.times do |i|  
  puts i  
end
```

樣式 2

```
5.times { |i|  
  puts i  
}
```

RUBY FOR RAILS

區塊 (BLOCK)

```
def greeting(name)
  yield name
end
```

```
greeting("eddie") { |n|
  puts "Hello, #{n}"
}
```

RUBY FOR RAILS

區塊 (BLOCK)

» 請試著寫出下的程式碼：

```
# 土砲 select 方法(只印出單數)  
my_select([1, 2, 3, 4, 5]) { |i| i.odd? }
```

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 物件(object) = 狀態(state) + 行為(behavior)

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 類別的命名必須是常數

```
class Animal
  def eat(food)
    puts "#{food} is Yammy!!"
  end
end
```

```
class Cat < Animal
end
```

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 建立一個 Cat 實體(instance)

```
c = Cat.new
```

```
c.eat "fish"    #=> fish is Yammy!
```

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 初始化 `initialize`

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 實體變數 `instance variable`

» 開頭有 "@"

» 存活在每個實體內

```
@name = "eddie"
```

```
@age = 18
```

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 實體變數 instance variable

```
class PostsController < ApplicationController
  def index
    @posts = Post.all
  end
end
```

RUBY FOR RAILS

物件導向程式設計 (OOP)

» Rails 裡的 View 可以使用 Controller 建立的實體變數是怎麼回事？

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 實體方法及類別方法

```
class Cat
  def sleep
    # ...
  end
end
```

```
kitty = Cat.new      # kitty 是一個 Cat 類別的實體
kitty.sleep          # "sleep" 作用在 kitty 上，它是個實體方法
```

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 實體方法及類別方法

```
class Cat
  def self.all
    # ...
  end
end
```

Cat.all # 在 Cat 類別上呼叫方法

RUBY FOR RAILS

物件導向程式設計 (OOP)

» Public, Protected, Private 方法

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 存取屬性

» `attr_reader`、`attr_writer` 以及 `attr_accessor`

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 開放類別 (Open Class)

```
class String
  def say_hello
    "hi, I am #{self}"
  end
end
```

```
puts "eddie".say_hello    #=> hi, I am eddie
```

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 問題： 我有一隻貓，我希望這隻貓會飛

» 答： 建立一個鳥類別，然後叫貓去繼承它？

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 模組 (Module)

```
module Flyable
  def fly
    puts "I can fly"
  end
end
```

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 引入模組

```
class Cat  
  include Flyable  
end
```

```
kitty = Cat.new  
kitty.fly
```

RUBY FOR RAILS

物件導向程式設計 (OOP)

» 2 個冒號 (::) 的意思？

```
class User < ActiveRecord::Base  
end
```

RUBY FOR RAILS

物件導向程式設計 (OOP)

» Ruby 的物件模型 (Ruby Object Model)

課後練習

1. 作業練習 <https://goo.gl/u6uzqj>
2. 看一下第一堂課使用 Scaffold 產生的 Model、View 以及 Controller 檔案，試著解釋每一行是在做什麼的。

聯絡 高見龍 (EDDIE)

- » Blog: <https://kaochenlong.com>
- » Facebook: <http://www.facebook.com/eddiekao>
- » Twitter: <https://twitter.com/eddiekao>
- » Email: eddie@5xruby.tw
- » Mobile: +886-928-617-687