

LESSON 2

X-PROGRAM

1. 安裝套件

在終端機裡執行以下指令 (**TERMINAL** / 模擬終端機 ... 等等)

- ▶ Capybara

- ▶ `gem install capybara`

- ▶ Selenium-WebDriver

- ▶ `gem install selenium-webdriver`

2. 引用套件

新增一個 **RUBY** 檔案

- ▶ 打開 Sublime Text 編輯器，選擇 File -> New File
- ▶ 輸入
 - ▶ `require 'capybara'`
 - ▶ `require 'capybara/dsl'`
 - ▶ `include Capybara::DSL`

3. 來寫 CODE 囉

基本設定

- ▶ 在 include Capybara::DSL 的下方輸入
 - ▶ `Capybara.javascript_driver = :selenium`
 - ▶ `Capybara.current_driver = :selenium`
 - ▶ `Capybara.register_driver :selenium do |app|`
 - ▶ `Capybara::Selenium::Driver.new(app, :browser => :firefox)`
 - ▶ `end`

3. 來寫 CODE 囉

指定目標網址

- ▶ 輸入

- ▶ `url = "http://24h.pchome.com.tw/"`

- ▶ `visit(url)`

- ▶ `sleep(10)`

- ▶ 存檔，執行看看有沒有成功叫出瀏覽器

- ▶ `ruby 檔名.rb`

3. 來寫 CODE 囉

觀察

- ▶ 我們在設定中指定使用 Firefox，能不能使用 Chrome 呢？
 - ▶ 觀察錯誤訊息

3. 來寫 CODE 囉

找到元素

- ▶ 找到頁面上的搜尋框，右鍵 -> 檢查
 - ▶ `input#keyword`
- ▶ 找到頁面上的搜尋按鈕，右鍵 -> 檢查
 - ▶ `input#doSearch`

3. 來寫 CODE 囉

對元素進行操作

▶ 輸入

- ▶ `find('input#keyword').set('nintendo switch')`

- ▶ `find('input#doSearch').click`

▶ 再執行一次這支程式

3. 來寫 CODE 囉

印出一些東西來看看吧！

▶ 輸入

- ▶ `result_size = find('span#SearchInfo span').text.to_i`
- ▶ `puts "result size #{result_size}"`

3. 來寫 CODE 囉

印出商品資訊吧！

- ▶ 商品格的元素是 `dl.col3f`
- ▶ 輸入
 - ▶ `all('dl.col3f').each do |box|`
 - ▶ `puts box.find('h5 a').text`
 - ▶ `puts box.find('dd.c3f ul:first-child li span span').text`
 - ▶ `end`

3. 來寫 CODE 囉

回到網頁觀察

- ▶ 這次我們的目標網頁，是採用 Infinite Scroll 的換頁方式
- ▶ 沒有像上次的頁碼連結，怎麼辦？
- ▶ 下拉頁面就會讀取更多，怎麼下拉？

3. 來寫 CODE 囉

進行操作

- ▶ Capybara 有一種語法叫 `page.execute_script()`
- ▶ 它可以叫網頁執行 JavaScript
- ▶ JavaScript 有一個方法叫做：`window.scrollTo(x座標, y座標)`
- ▶ 結合起來就成了
 - ▶ `page.execute_script("window.scrollTo(x, y)")`

3. 來寫 CODE 囉

動起來動起來

▶ 輸入

- ▶ `page.execute_script("window.scrollTo(0, 3000)")`
- ▶ `sleep(10)`

3. 來寫 CODE 囉

思考

- ▶ 一次要移動多少？
 - ▶ 每一個商品格的元素長怎樣？

3. 來寫 CODE 囉

解法

- ▶ 記得我們有 `result_size` 這個變數嗎？
- ▶ 我們可以使用迴圈方式，一個一個元素捲動
- ▶ 提示
 - ▶ `i = 0`
 - ▶ `while i < result_size do`
 - ▶ `blahblahblah here`
 - ▶ `end`