



SCU LawTech

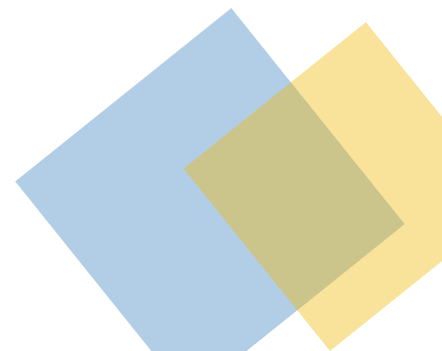
Class #06

Content

- ◆ 作業回顧
- ◆ XPATH 測驗
- ◆ selenium 介紹
- ◆ selenium 的定位方式
- ◆ selenium 三大等待方式
- ◆ selenium 常用語法
- ◆ selenium 實作

作業回顧

作業回顧



A large yellow diamond shape is centered on a white background. Inside the diamond, the text 'XPATH' and '測驗' are displayed in blue.

XPATH

測驗

常用語法

類別	運算式	描述
層級	/	從根結點選取(當前節點的下一級)
	//	從當前節點選取任意子孫節點(跨級)
	.	選取當前節點(如./h3表示當前節點下跨級匹配h3標籤)
屬性	@	屬性訪問(如div[@class="xx"]或//a/@href)
函數	text()	獲取節點文本內容
	contains(A,B)	A是否包含B
	last()	標籤列表中的最後一個(如//a[last()])

測驗

- 請定位標記紅色框框的位置！

司法法院 法學資料檢索系統

網站導覽 司法法院首頁 English 行動版 網頁縮放

首頁 本院主管或審判相關法規 判解函釋 裁判書查詢 簡易案件查詢 除權判決查詢 公示催告裁定查詢 系統說明

① 裁判書查詢 (含部分簡易案件) - 如您為無障礙網頁的使用者，請點擊[這裡](#)進行查詢

題目 1 題目 2

檢索字詞

可輸入法院名稱、裁判案號、案由、全文檢索字詞

送出查詢 檢索字詞說明 ?

※如欲使用可對各項欄位設定條件之複合欄位查詢功能，請按 [更多條件查詢](#)

簡易輸入格式範例：[更多說明](#)

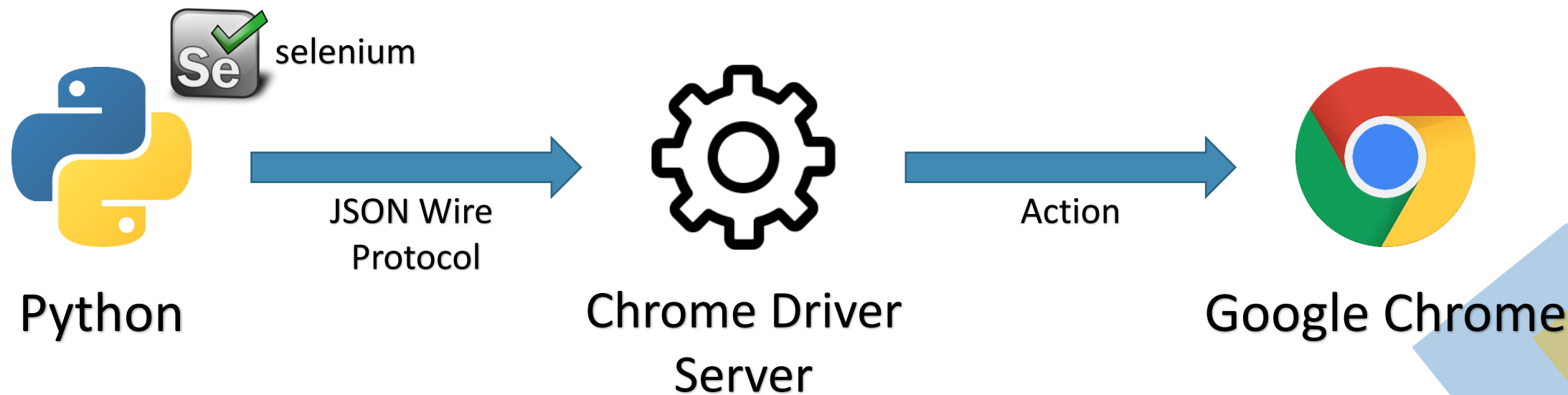
- 輸入【105訴123】：可檢索所有法院中，含有【105年度訴字第123號】之判例裁判。
- 輸入【台北地院105訴123】：可檢索【臺灣臺北地方法院105年度訴字第123號】之資料，或全文含有以下案號格式之判例裁判
 - 臺（台）北地方法院105年度訴字第123號
 - 臺（台）北地方法院一百零五年度訴字第一二三號
 - 臺（台）北地方法院一〇五年度訴字第一二三號
 - 臺（台）北地方法院一〇五年訴字一二三

A large, solid yellow diamond shape is centered on a white background. The word "selenium" is written in a blue, sans-serif font across the middle of the diamond.

selenium

什麼是 selenium ?

- **Selenium** 是為瀏覽器自動化 (Browser Automation) 需求所設計的一套工具集合，讓程式可以直接驅動瀏覽器進行各種網站操作。

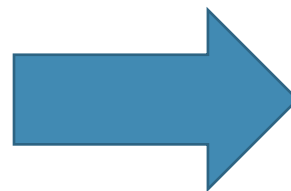


下載 Chromedriver (1)

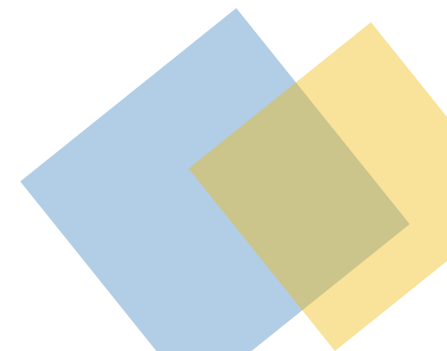
- **STEP 1** : 打開 Google Chrome，左上角點擊 ，並選擇【說明】，點擊【關於 Google Chrome】



- **STEP 2** : 查看當前 Google Chrome 最新版本



版本 : 86.0.4240.183



下載 Chromedriver (2)






- **STEP 3** : 點擊鏈接，選擇對應前兩位版本號，選擇相應的下載選項

Current Releases

- If you are using Chrome version 87, please download [ChromeDriver 87.0.4280.20](#)
- If you are using Chrome version 86, please download [ChromeDriver 86.0.4240.22](#)
- If you are using Chrome version 85, please download [ChromeDriver 85.0.4183.87](#)
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

版本：86.0.4240.183

- **STEP 4** : 下載對應的作業系統安裝包，在完成下載後，解壓縮文件

Name	
	Parent Directory
	chromedriver_linux64.zip
	chromedriver_mac64.zip
	chromedriver_win32.zip
	notes.txt

A large yellow diamond shape is centered on a white background. Inside the diamond, the text 'selenium' and '定位方式' are displayed in blue.

selenium

定位方式

selenium 定位方式 (1)

- selenium 提供 8 種定位方式：

【id】

id 為 HTML 標籤中的屬性，其屬性值必須是唯一的。

【class_name】

class 為 HTML 標籤中的屬性，其可作為操控或編輯該標籤的媒介。

【xpath】

xpath 為一種路徑語言，其用於搜尋 HTML 中的內容。

【name】

Name 為 HTML 標籤中的屬性，可透過自定義其標籤內容。



selenium 定位方式 (2)

- selenium 提供 8 種定位方式：

【css_selector】

css selector 用来匹配 HTML 中標籤元素，css 样式的定义为：
选择器 { 样式 }。

【link_text】

link 為 HTML 標籤中的屬性，其可達到文字超鏈接的效果。

【partial_link_text】

link 為 HTML 標籤中的屬性，其可達到文字超鏈接的效果。

【tag_name】

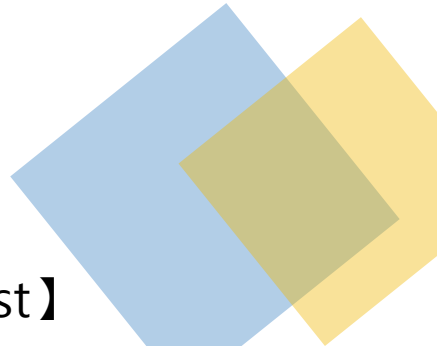
tag 為 HTML 中的標籤。



selenium 定位程式範例

driver.find_element<s>_by_<定位方式> ()

*** 每個定位函式之返回 (Output / Return) 資料類型不同 ! 【element -> str / elements -> list】



A large yellow diamond shape is centered on a white background. Inside the diamond, the text 'selenium' and '等待方式' are displayed in blue.

selenium

等待方式

selenium 等待 (1)

- 強制等待

簡單粗暴，不管瀏覽器是否加載完畢，直接執行下面的程式碼。

```
# -*- coding: utf-8 -*-  
from selenium import webdriver  
from time import sleep  
  
driver = webdriver.Firefox()  
driver.get('https://huilansame.github.io')  
  
sleep(3) # 強制等待3秒再執行下一步  
  
print driver.current_url  
driver.quit()
```

selenium 等待 (2)

- 隱性等待

隐形等待是设置了一个最长等待时间，如果在规定时间内网页加载完成，则执行下一步，否则一直等到时间截止，然后执行下一步。

```
# -*- coding: utf-8 -*-
from selenium import webdriver

driver = webdriver.Firefox()
driver.implicitly_wait(30) # 隐性等待，最长等30秒
driver.get('https://huilansame.github.io')

print driver.current_url
driver.quit()
```

```
quit()
driver.quit()
```

*** 隐性等待对整个 driver 的周期都起作用，所以只要设置一次即可。

<https://huilansame.github.io/huilansame.github.io/archivers/sleep-implicitlywait-wait>

selenium 等待 (3)

- 顯性等待

根据判断条件而进行灵活地等待了。它主要的意思就是：程序每隔 x 秒看一眼，如果条件成立了，则执行下一步，否则继续等待，直到超过设置的最长时间。

```
# -*- coding: utf-8 -*-
from selenium import webdriver
from selenium.webdriver.support.wait import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.webdriver.common.by import By

driver = webdriver.Firefox()
driver.implicitly_wait(10) # 隐性等待和显性等待可以同时用，但要注意：等待的最长时间取两者之中的大者
driver.get('https://huilansame.github.io')
locator = (By.LINK_TEXT, 'CSDN')

try:
    WebDriverWait(driver, 20, 0.5).until(EC.presence_of_element_located(locator))
    print driver.find_element_by_link_text('CSDN').get_attribute('href')
finally:
    driver.close()
```

爬蟲實作

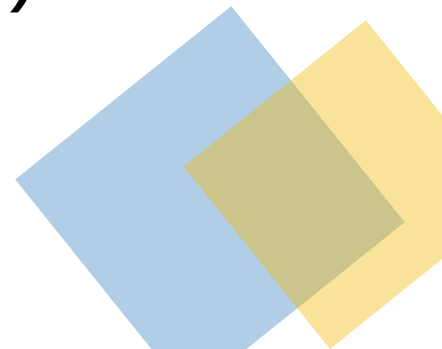
爬蟲：法學資料檢索系統

<https://law.judicial.gov.tw/FJUD/default.aspx>

今日任務：

在輸入指定關鍵字搜尋後，截取所有判決內容（標題，鏈接，日期，判決，內文）。

*** 在尋找目標網址時，通常其內容會出現在開發人員工具中的【XHR】及【HTML】項目中。





THANKS