

## 2.5 Selenium

# 你將學習到

1. 打開/關閉瀏覽器
2. 瀏覽器設定
3. 瀏覽器控制
4. 瀏覽器資料
5. 視窗控制
6. 元素取得
7. 元素控制

# Selenium 是什麼？

Selenium 是瀏覽器自動控制套件。它會控制你的瀏覽器自動完成一系列動作，例如打開網頁、滑鼠點擊、鍵盤輸入等，並取得目標元素內的資料。

Selenium 不只是限於 Python 語言，其他編程語言例如 Java、R 等也有相應的 Selenium 套件，用法相似。

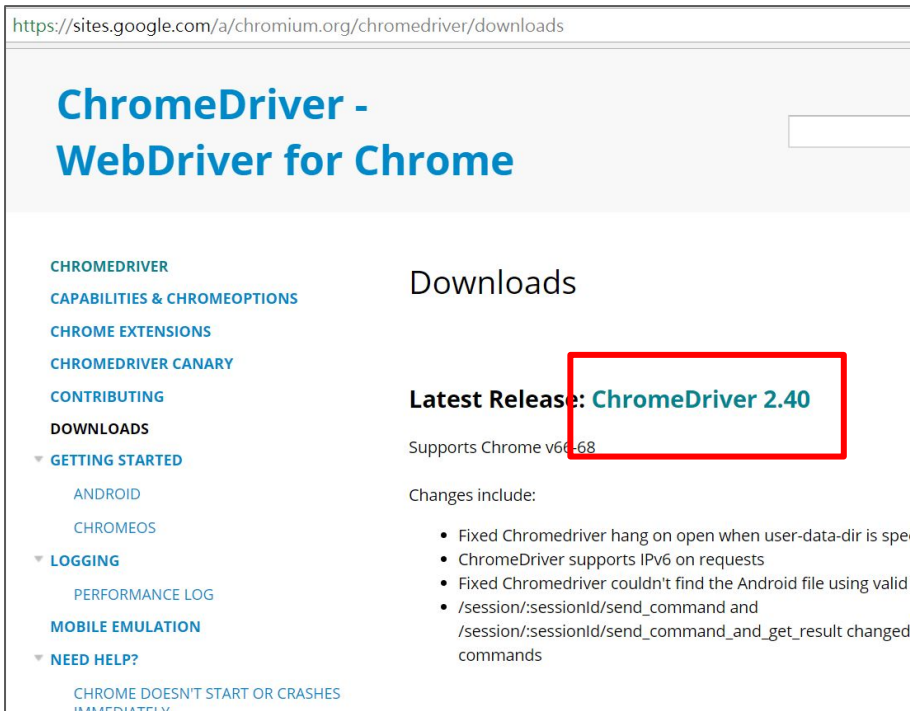
# Selenium、Chrome Driver、Chrome

Selenium 函式庫是經過一個名為 Chrome Driver 的程式來控制瀏覽器。Chrome Driver 才是控制瀏覽器的真身。但由於 Chrome 瀏覽器會經常自動更新為最新版本，因此 Selenium、Chrome Driver 版本也有可能需要相應更新。



這三個組件的版本需要配對

# 更新 Chrome Driver



The screenshot shows the ChromeDriver download page. The URL in the browser bar is <https://sites.google.com/a/chromium.org/chromedriver/downloads>. The page title is "ChromeDriver - WebDriver for Chrome". On the left is a navigation menu with links: CHROMEDRIVER, CAPABILITIES & CHROME OPTIONS, CHROME EXTENSIONS, CHROMEDRIVER CANARY, CONTRIBUTING, DOWNLOADS, GETTING STARTED (expanded), ANDROID, CHROME OS, LOGGING (expanded), PERFORMANCE LOG, MOBILE EMULATION, and NEED HELP? (expanded). The main content area is titled "Downloads" and features a "Latest Release: ChromeDriver 2.40" which is highlighted with a red rectangle. Below this, it states "Supports Chrome v66-68" and lists "Changes include:" followed by a bulleted list of fixes.

<https://sites.google.com/a/chromium.org/chromedriver/downloads>

## ChromeDriver - WebDriver for Chrome

- CHROMEDRIVER
- CAPABILITIES & CHROME OPTIONS
- CHROME EXTENSIONS
- CHROMEDRIVER CANARY
- CONTRIBUTING
- DOWNLOADS
- ▼ GETTING STARTED
  - ANDROID
  - CHROME OS
- ▼ LOGGING
  - PERFORMANCE LOG
- MOBILE EMULATION
- ▼ NEED HELP?
- CHROME DOESN'T START OR CRASHES IMMEDIATELY

### Downloads

**Latest Release: ChromeDriver 2.40**

Supports Chrome v66-68

Changes include:

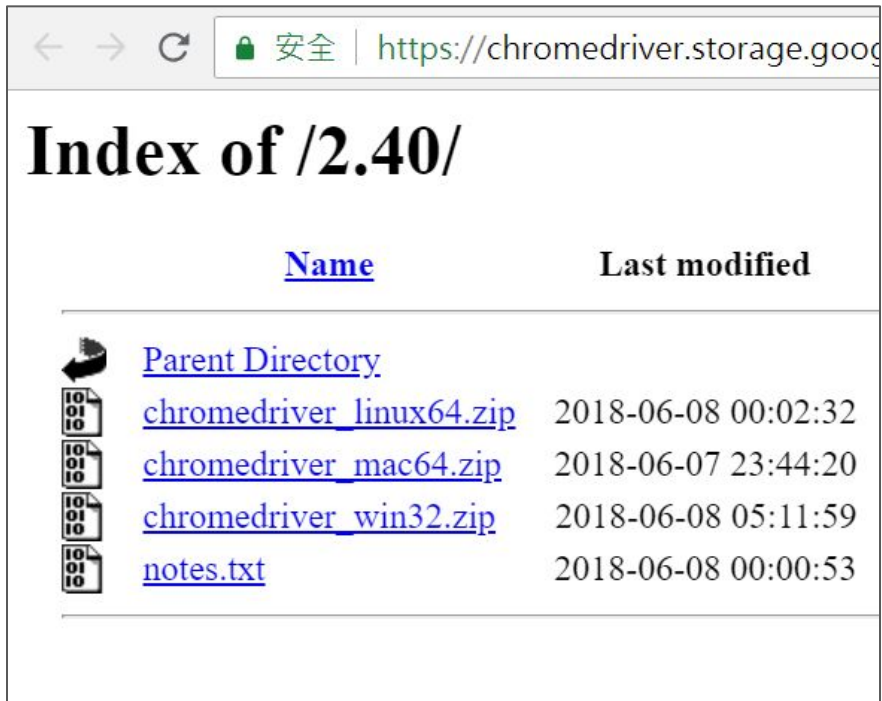
- Fixed Chromedriver hang on open when user-data-dir is specified
- ChromeDriver supports IPv6 on requests
- Fixed Chromedriver couldn't find the Android file using valid /session/:sessionId/send\_command and /session/:sessionId/send\_command\_and\_get\_result changed commands






## 步驟 1:

如果某一天你發現程式無故出錯，你可嘗試更新 Chrome Driver。你可以 Google “Chrome Driver” 找到以下這個網址。

<https://sites.google.com/a/chromium.org/chromedriver/downloads>

# 更新 Chrome Driver



<u>Name</u>	Last modified
 <a href="#">Parent Directory</a>	
 <a href="#">chromedriver_linux64.zip</a>	2018-06-08 00:02:32
 <a href="#">chromedriver_mac64.zip</a>	2018-06-07 23:44:20
 <a href="#">chromedriver_win32.zip</a>	2018-06-08 05:11:59
 <a href="#">notes.txt</a>	2018-06-08 00:00:53




## 步驟 2:

根據您的作業系統下載相應版本。  
注意 windows 64-bit 也是用 win32 版本。

# 更新 Chrome Driver

## 步驟 3:

解壓縮下載的 zip 檔, 把裡面的 chromedriver.exe (Windows) 或 chromedriver (Mac) 放到程式資料夾內。

 2.4 selenium.py	17/8/2018 22:16	PY 檔案
 2.5 file.py	17/8/2018 0:40	PY 檔案
 chromedriver	20/9/2018 17:02	檔案
 chromedriver.exe	20/9/2018 17:03	應用程式

# Selenium 函數庫組成

Selenium 函數庫分成以下 7 個部份：

1. 打開/關閉瀏覽器
2. 瀏覽器設定
3. 瀏覽器控制
4. 瀏覽器資料
5. 視窗控制
6. 元素取得
7. 元素控制



# 匯入 selenium 模組

```
from selenium import webdriver
```

# 打開瀏覽器

# Windows

```
driver = webdriver.Chrome( './chromedriver.exe' )
```

# Mac

```
driver = webdriver.Chrome( './chromedriver' )
```

# 前往網址 driver.get()

前往 aastocks.com 作為示範。

```
driver.get( 'http://www.aastocks.com/tc/ltp/rtquote.aspx?symbol=00001' )
```

# 取得股票輸入框元素

## `driver.find_element_by_css_selector()`



運用 chrome developer tools  
檢查輸入框元素的 id、class、attributes, 找出相應的 CSS selectors。

## 取得股票輸入框元素

# `driver.find_element_by_css_selector()`

```
inputElem = driver.find_element_by_css_selector( '[id="sb2-txtSymbol-aa"]' )
```

# 清除輸入框文字 `element.clear()`

```
element.clear()
```

## 輸入框輸入文字 `element.send_keys()`

```
element.send_keys( '00001' )
```

# 點擊前往按鈕 `element.click()`

```
driver.find_element_by_css_selector( '[id="sb2-btnSubmit"]' ).click()
```

取得按鈕元素

點擊滑鼠左鍵



## 切換至 iframe 頁框

### `driver.switch_to.frame()`

如果目標元素位於 iframe/frame 頁框內，我們需要先切換到頁框內才可用 `find_element_by_*`() 取得元素。

```
driver.switch_to.frame( 'frameQuickQuote' )
```

# 取得股票現價 element.text

# 取得元素文字內容

```
driver.find_element_by_css_selector( 'css-selectors' ).text
```

# 取得元素文字內容, 移除文字前後多餘的空格

```
driver.find_element_by_css_selector( 'css-selectors' ).text.strip()
```

# 取得元素文字內容, 移除文字前後多餘的空格

```
driver.find_element_by_css_selector( 'css-selectors' ).text.strip().split("\n")[-1]
```

# 將 iframe 切換至預設狀態

## `driver.switch_to.default_content()`

若不想留在 iframe 內，可輸入以下指令，變成預設狀態。

```
driver.switch_to.default_content()
```

# 設定頁面等待超時

## `driver.set_page_load_timeout()`

瀏覽器最多等待頁面載入20秒，若網頁沒有全部載入完成，則拋出例外。

```
driver.set_page_load_timeout( 20 )
```

# 設定元素等待超時

## `driver.implicitly_wait()`

如果 `find_element_by_*`() 找不到指定的元素, 則最多等待10秒, 超時則拋出例外。

```
driver.implicitly_wait( 10 )
```

# 進階部份 (選讀)

接下來會介紹更多 Selenium 進階用法，  
課堂上的實作不會用到。

# 取得頁面元素函數

函數	解釋	效果
<code>driver.find_element_by_id( "student" )</code>	取得指定id元素	<code>&lt;div id="student"&gt; &lt;/div&gt;</code>
<code>driver.find_element_by_name( "username" )</code>	取得指定name 屬性元素	<code>&lt;input type="text" name="username" /&gt;</code>
<code>driver.find_element_by_tag_name( "h1" )</code>	取得指定tag元素	<code>&lt;h1&gt;Heading 1&lt;/h1&gt;</code>
<code>driver.find_element_by_class_name( "dog" )</code>	取得含有class屬 性元素	<code>&lt;div class="animal dog"&gt;&lt;/div&gt;</code>

## 取得頁面元素函數

函數	解釋	效果
<code>driver.find_element_by_link_text("Continue" )</code>	取得連結文字等於"Continue"的元素	<code>&lt;a href="abc.com"&gt; Continue &lt;/a&gt;</code>
<code>driver.find_element_by_partial_link_text("Cont" )</code>	取得連結文字含有"Cont"的元素	<code>&lt;a href="abc.com"&gt; Continue &lt;/a&gt;</code>



# 取得頁面元素函數

函數	解釋	效果
<code>driver.find_element_by_xpath( "//h1" )</code> <a href="https://www.w3schools.com/xml/xpath_syntax.asp">https://www.w3schools.com/xml/xpath_syntax.asp</a>	取得指定 xpath 元素	<code>&lt;h1&gt;Header 1&lt;/h1&gt;</code>
<code>driver.find_element_by_css_selector( "h1" )</code> <a href="https://www.w3schools.com/cssref/css_selectors.asp">https://www.w3schools.com/cssref/css_selectors.asp</a>	取得指定 css selector 元素	<code>&lt;h1&gt;Header 1&lt;/h1&gt;</code>

# 取得多個元素版本

除了 `find_element_by_id()` 外，其他函數都有取得單個及多個元素版本。

單個元素	多個元素
<code>driver.find_element_by_id()</code>	無
<code>driver.find_element_by_name()</code>	<code>driver.find_elements_by_name()</code>
<code>driver.find_element_by_tag_name()</code>	<code>driver.find_elements_by_tag_name()</code>
<code>driver.find_element_by_class_name()</code>	<code>driver.find_elements_by_class_name()</code>
<code>driver.find_element_by_link_text()</code>	<code>driver.find_elements_by_link_text()</code>
<code>driver.find_element_by_partial_link_text()</code>	<code>driver.find_elements_by_partial_link_text()</code>
<code>driver.find_element_by_xpath()</code>	<code>driver.find_elements_by_xpath()</code>
<code>driver.find_element_by_css_selector()</code>	<code>driver.find_elements_by_css_selector()</code>

# 取得多個元素版本

阿思達克財經網

市場動態

報價

主頁

香港指數

快速即時報價 ▾

中國站

國際指數

即時詳細報價

產品資訊

中國主要指數

即時基本圖表

即時期貨

即時圖表分析

活躍股票

即時互動圖表

客戶服務

行業分類表現

即時期貨

會員登記

香港指數成份股

相關認股證報價

更改資料

所有國企股

相關牛熊證報價

更改密碼

所有紅籌股

股票代號搜尋

忘記密碼

所有創業板股份

```
selectElem =  
driver.find_elements_by_css_selector('.s1')
```

```
selectElem[0].text      #主頁
```

```
selectElem[1].text      #中國站
```

# 取得元素屬性

## `element.get_attribute()`

可以取得特定元素的內容，以下例子取出 href 的內容。

```
driver.find_element_by_css_selector('css-selector').get_attribute('href')
```

取得超連結元素網址 href

# 取得元素代碼

## innerHTML, outerHTML

innerHTML 取得元素內包含的代碼。而 outerHTML 則取得元素內和元素本身的代碼。

```
driver.find_element_by_css_selector('css-selector').get_attribute('outerHTML')
```

取得元素源代碼, 包括元素本身

```
driver.find_element_by_css_selector('css-selector').get_attribute('innerHTML')
```

取得元素源代碼, 不包括元素本身

## 多瀏覽器控制

```
# 打開兩個瀏覽器視窗, 分開兩個變數儲存 (driver1, driver2)
```

```
driver1 = webdriver.Chrome( './chromedriver.exe' )
```

```
driver2 = webdriver.Chrome( './chromedriver.exe' )
```

```
# 各自前往不同網站同時抓取資料
```

```
driver1.get( 'https://www.google.com' )
```

```
driver2.get( 'https://hk.yahoo.com' )
```

# 重新整理頁面

## `driver.refresh()`

```
driver.refresh()
```

# 拍攝屏幕

## `driver.save_screenshot()`

```
driver.save_screenshot( 'result.png' )
```

暫時只支援 .png !



# 運行 Javascript 代碼

## `driver.execute_script()`

```
driver.execute_script( "window.open();" ; )
```

運行 Javascript 代碼

開啟新分頁

# 切換分頁

## `driver.switch_to.window()`

`driver.switch_to.window()` 需提供目標分頁ID參數，可利用  
`driver.window_handles` 取得。

`driver.switch_to.window()` ( `driver.window_handles[-1]` )

切換分頁

取得最新開啟的分頁 ID

# 關閉分頁

如果關閉最後一個分頁，瀏覽器也會被一併關閉。

```
driver.close()
```

# 關閉瀏覽器

一次關閉所有分頁和瀏覽器。

```
driver.quit()
```

# 輸入特殊按鍵

例子: `from selenium.webdriver.common.keys import Keys`  
`element.send_keys(Keys.ENTER)`

`ADD = u'\ue025'`

`ALT = u'\ue00a'`

`ARROW_DOWN = u'\ue015'`

`ARROW_LEFT = u'\ue012'`

`ARROW_RIGHT = u'\ue014'`

`ARROW_UP = u'\ue013'`

`BACKSPACE = u'\ue003'`

`BACK_SPACE = u'\ue003'`

`CANCEL = u'\ue001'`

`CLEAR = u'\ue005'`

`COMMAND = u'\ue03d'`

`CONTROL = u'\ue009'`

`DECIMAL = u'\ue028'`

`DELETE = u'\ue017'`

`DIVIDE = u'\ue029'`

`DOWN = u'\ue015'`

`END = u'\ue010'`

`ENTER = u'\ue007'`

`EQUALS = u'\ue019'`

`ESCAPE = u'\ue00c'`

`F1 = u'\ue031'`

`F10 = u'\ue03a'`

`F11 = u'\ue03b'`

`F12 = u'\ue03c'`

`F2 = u'\ue032'`

`F3 = u'\ue033'␣`

`F4 = u'\ue034'`

`F5 = u'\ue035'`

`F6 = u'\ue036'`

`F7 = u'\ue037'`

`F8 = u'\ue038'`

`F9 = u'\ue039'`

`HELP = u'\ue002'`

`HOME = u'\ue011'`

`INSERT = u'\ue016'`

`LEFT = u'\ue012'`

`LEFT_ALT = u'\ue00a'`

`LEFT_CONTROL = u'\ue009'`

`LEFT_SHIFT = u'\ue008'`

`META = u'\ue03d'`

`MULTIPLY = u'\ue024'`

`NULL = u'\ue000'`

`NUMPAD0 = u'\ue01a'`

`NUMPAD1 = u'\ue01b'`

`NUMPAD2 = u'\ue01c'`

`NUMPAD3 = u'\ue01d'`

`NUMPAD4 = u'\ue01e'␣`

`NUMPAD5 = u'\ue01f'`

`NUMPAD6 = u'\ue020'`

`NUMPAD7 = u'\ue021'`

`NUMPAD8 = u'\ue022'`

`NUMPAD9 = u'\ue023'`

`PAGE_DOWN = u'\ue00f'`

`PAGE_UP = u'\ue00e'`

`PAUSE = u'\ue00b'`

`RETURN = u'\ue006'`

`RIGHT = u'\ue014'␣`

`SEMICOLON = u'\ue018'`

`SEPARATOR = u'\ue026'`

`SHIFT = u'\ue008'`

`SPACE = u'\ue00d'`

`SUBTRACT = u'\ue027'`

`TAB = u'\ue004'`

`UP = u'\ue013'`

# 輸入特殊按鍵例子

在輸入框輸入 00002 後, 再輸入 ENTER, 頁面會連結到另一頁。

```
from selenium.webdriver.common.keys import Keys  
inputElem = driver.find_element_by_css_selector('#PY_txt')  
inputElem.send_keys('00002')  
inputElem.send_keys(Keys.ENTER)
```

# Selenium 函數參考

## 打開瀏覽器

```
driver = webdriver.Chrome(executable_path)
```

打開瀏覽器

## 關閉瀏覽器

```
driver.close()
```

關閉分頁

```
driver.quit()
```

關閉瀏覽器

## 瀏覽器設定

```
driver.implicitly_wait(seconds)
```

如果瀏覽器頁面載入後, `find_element_by_*`() 指定的元素未立刻出現, 最多等待seconds 秒, 超時則拋出例外。

```
driver.set_page_load_timeout(seconds)
```

瀏覽器最多等待頁面載入seconds 秒, 超時則拋出例外。

# Selenium 函數參考

瀏覽器控制	
driver.get(url)	前往網址 url
driver.refresh()	重新整理頁面
driver.save_screenshot(filename)	拍攝屏幕
driver.execute_script(javascript)	運行 Javascript 代碼
瀏覽器資料	
driver.current_url	取得現在頁面網址
driver.title	取得現在頁面標題



# Selenium 函數參考

視窗控制	
<code>driver.window_handles</code>	取得所有已打開的分頁
<code>driver.switch_to.frame(element)</code>	切換到頁框 iframe/frame
<code>driver.switch_to.default_content()</code>	切換到最外層頁框
<code>driver.switch_to.window(id)</code>	切換到分頁, 需指定分頁ID, 例如: <code>driver.switch_to.window(driver.window_handles[-1] )</code>
元素取得	
<code>element = driver.find_element_by_css_selector(css_selector)</code>	用 css selector 取得元素

# Selenium 函數參考

元素控制	
<code>element.text</code>	取得元素文字
<code>element.get_attribute(attriute_name)</code>	取得元素屬性
<code>element.get_attribute('innerHTML')</code>	取得元素內 HTML 代碼, 不包括自己
<code>element.get_attribute('outerHTML')</code>	取得元素內 HTML 代碼, 包括自己
<code>element.click()</code>	點擊滑鼠左鍵
<code>element.send_keys(text)</code>	文字框輸入文字
<code>element.clear()</code>	清除文字框

# Selenium 函數參考

<http://selenium-python.readthedocs.io/index.html>

或

Google “Python Selenium”

# 學習回顧

1. 打開/關閉瀏覽器
2. 瀏覽器設定
3. 瀏覽器控制
4. 瀏覽器資料
5. 視窗控制
6. 元素取得
7. 元素控制