

# Lab 4: Web Applications Testing

*Software Testing 2023*

*2023/03/16*

# whoami

- Software Quality Lab @ EC547
- TA
  - a. 蔡惠喬
    - [hctsai.cs10@nycu.edu.tw](mailto:hctsai.cs10@nycu.edu.tw)
  - b. 陳舜寧
    - [xdev11.cs11@nycu.edu.tw](mailto:xdev11.cs11@nycu.edu.tw)
  - c. 萬尚宏
    - [sun.cs11@nycu.edu.tw](mailto:sun.cs11@nycu.edu.tw)

# GitHub Repo

- <student\_id>-ST-2023
- Add collaborators
  - XDEv11, chameleon10712, skhuang

# Selenium

## About selenium

Selenium automates browsers.

Primarily used for automating web applications for testing purpose, but is certainly not limited to just that.

Boring web-based administration tasks can also be automated as well.



# Selenium webdriver

*WebDriver* uses browser automation APIs provided by browser vendors to control browser and run tests, as if a real user is operating the browser.

Support multiple browsers and languages.



# Install selenium library with pip

```
pip3 install selenium
```

```
pip3 install webdriver_manager
```

# Auto install and use webdriver

```
1  from selenium import webdriver
2  from webdriver_manager.chrome import ChromeDriverManager
3  from selenium.webdriver.chrome.service import Service
4
5  driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()))
6  |
7  driver.get("https://google.com")
8
9
```

<https://pypi.org/project/webdriver-manager/>



# Example

# Browser navigation

Navigate to:

```
driver.get("https://google.com")
```

Refresh the current page:

```
driver.refresh()
```

Pressing the browser's forward, back button:

```
# Pressing the browser's back button  
driver.back()  
# Pressing the browser's forward button  
driver.forward()
```

Quitting the browser at the end of a session:

```
driver.quit()
```

# Windows and Tabs

Get window handle:

```
In [50]: driver.window_handles
Out[50]:
['CDwindow-40DB5FF2B7E673E82B03E1C52E09A5FB',
 'CDwindow-C5BABC52E81C69C12E3EA3B3872CF102',
 'CDwindow-E45204BB0C2F54FA4D87A260E2F00AFA',
 'CDwindow-F44134887F43D2C742D967CAB54B60BE',
 'CDwindow-8C945448D03658C90163E0907934B3DB']
```

Switching windows or tabs:

```
# Store the ID of the original window
original_window = driver.current_window_handle

# Check we don't have other windows open already
assert len(driver.window_handles) == 1

# Click the link which opens in a new window
driver.find_element(By.LINK_TEXT, "new window").click()

# Wait for the new window or tab
wait.until(EC.number_of_windows_to_be(2))

# Loop through until we find a new window handle
for window_handle in driver.window_handles:
    if window_handle != original_window:
        driver.switch_to.window(window_handle)
        break

# Wait for the new tab to finish loading content
wait.until(EC.title_is("SeleniumHQ Browser Automation"))
```

Create new window (or) new tab and switch:

```
# Opens new tab and switch to new tab
driver.switch_to.new_window('tab')

# Opens new window and switch to new window
driver.switch_to.new_window(['window'])
```

Closing a window or tab:

```
#Close the tab or window
driver.close()

#Switch back to the old tab or window
driver.switch_to.window(original_window)
```

# Element selection strategies

Locator	Description
class name	Locates elements whose class name contains the search value (compound class names are not permitted)
css selector	Locates elements matching a CSS selector
id	Locates elements whose ID attribute matches the search value
name	Locates elements whose NAME attribute matches the search value
link text	Locates anchor elements whose visible text matches the search value
partial link text	Locates anchor elements whose visible text contains the search value. If multiple elements are matching, only the first one will be selected.
tag name	Locates elements whose tag name matches the search value
xpath	Locates elements matching an XPath expression

<https://www.selenium.dev/selenium/docs/api/py/webdriver/selenium.webdriver.common.by.html>

# Waits

```
<!doctype html>
<meta charset=utf-8>
<title>Race Condition Example</title>

<script>
    var initialised = false;
    window.addEventListener("load", function () {
        setTimeout(function () {
            var newElement = document.createElement("p");
            newElement.textContent = "Hello from JavaScript!";
            document.body.appendChild(newElement);
            initialised = true;
        }, 3000);
    });
</script>
```

# Waits

```
driver.navigate("file:///race_condition.html")  
el = driver.find_element(By.TAG_NAME, "p")  
assert el.text == "Hello from JavaScript!"
```

```
selenium.common.exceptions.NoSuchElementException: Message: no such element:  
Unable to locate element: {"method":"css selector","selector":"p"}
```

# Waits

## Explicit wait: (with condition)

```
driver.get("file:///home/a13579and2468/software_testing_2022/race_condition.html")
el = WebDriverWait(driver,10).until(lambda d: d.find_element(By.TAG_NAME,"p"))
assert el.text == "Hello from JavaScript!"
```

## Expected condition:

```
from selenium.webdriver.support import expected_conditions as EC

driver.get("file:///home/a13579and2468/software_testing_2022/race_condition.html")
el = WebDriverWait(driver,10).until(EC.element_to_be_clickable((By.TAG_NAME,'p')))
assert el.text == "Hello from JavaScript!"
```

## Implicit wait: (without condition)

```
driver.get("file:///home/a13579and2468/software_testing_2022/race_condition.html")
driver.implicitly_wait(5)
el = driver.find_element(By.TAG_NAME,"p")
assert el.text == "Hello from JavaScript!"
```

# Others

- .text
- .click()
- .send\_keys()
- Other information can be found in document
  - <https://www.selenium.dev/documentation/en/webdriver/>





Lab

# Lab 4

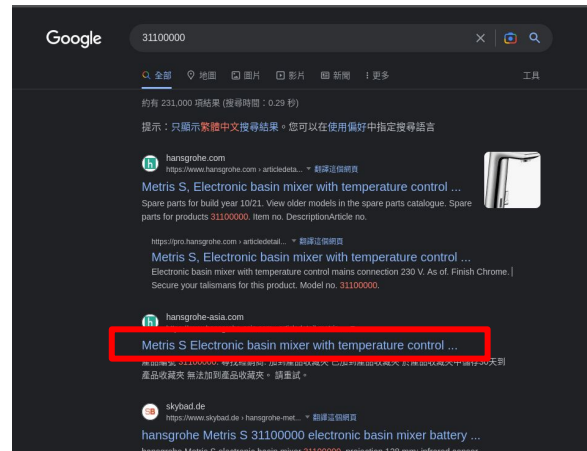
## Write scripts to emulate a user's operations below:

- launch browser →
- navigate to NYCU home page (<https://www.nycu.edu.tw/>) →
- maximize the window →
- click 新聞 →
- click first news →
- print the title and content →
- open a new tab and switch to it →
- navigate to google (<https://www.google.com>) →
- input your student number and submit →
- print the title of second result →
- close the browser

2023-03-15 用人類智慧聊人工智慧 生成式AI對學習的挑戰與反思

生成式AI工具（如：ChatGPT）的興起，越來越多學生將其應用於撰寫作業、論文，各大學陸續發出相關教學指引來因應生成式AI所帶來的教學衝擊。對此，陽明交通大學於分別於3/10及3/14舉辦教師與學生的校內座談會，透過開放的各方討論蒐集師生...

最新消息	科學新知	校園活動	藝術人文	人物報導	國際事務	產學合作
2023-03-15	用人類智慧聊人工智慧 生成式AI對學習的挑戰與反思					
2023-03-10	開啟商教新頁，培育未來人才— 陽明交大模式					
2023-03-08	校歌徵選					
2023-03-06	台灣免疫學之父、陽明大學首任校長韓詔華辭世，醫界齊聚分享台灣免疫史					



## Lab 4

By default, GitHub Actions cannot run Selenium. To complete the lab, you can use the following code to initialize the webdriver. However, note that this option will not open the browser. Therefore, we recommend that you do not add this option until you have finished all jobs.

```
from selenium.webdriver.chrome.options import Options
```

```
options = Options()
```

```
options.add_argument('--headless')
```

```
options.add_argument('--window-size=1920,1080')
```

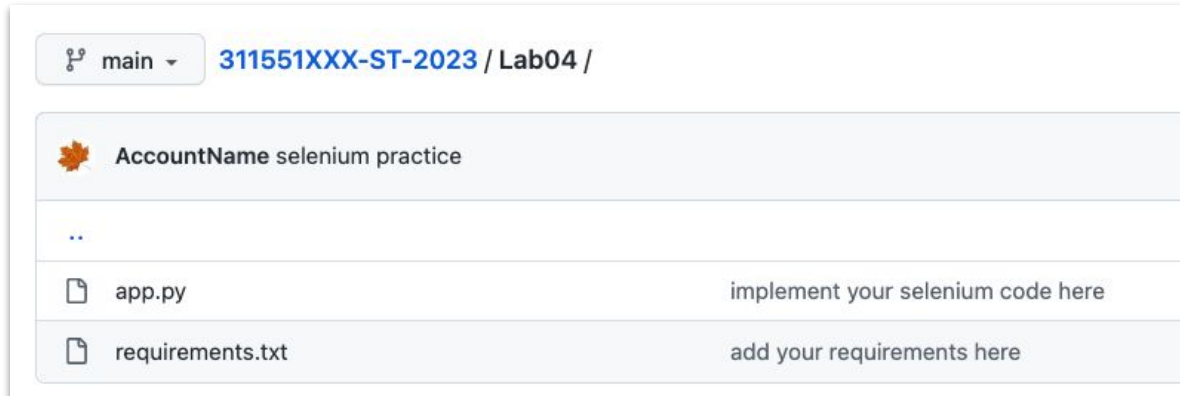
```
options.add_argument('--disable-gpu')
```

```
driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()),
```

```
options=options)
```

# Lab 4

- command
  - `python3 app.py`
- put your requirements.txt under Lab04 directory
  - `pip freeze > requirements.txt`



# Lab 4

1 ▶ Run cd Lab04

12

13 [WDM] - Downloading: 0%| | 0.00/7.05M [00:00<?, ?B/s]

14 [WDM] - Downloading: 100%| | 7.05M/7.05M [00:00<00:00, 163MB/s]

15 用人類智慧聊聊人工智慧 生成式AI對學習的挑戰與反思

16 生成式AI工具（如：ChatGPT）的興起，越來越多學生將其應用於撰寫作業、論文，各大學陸續發出相關教學指引來因應生成式AI所帶來的教學衝擊。對此，陽明交通大學於分別於3/10及3/14舉辦教師與學生的校內座談會，透過開放的多方討論蒐集師生對於ChatGPT的使用觀點及疑慮，形塑出對於議題的校內共識，藉此協助校方訂立相關規範。

17 3/10召開生成式人工智慧教與學的教師論壇，與會教師一致認為面對跨世代的科技應用，應朝正面積極的態度面對，結合本校電機、資訊、生醫等優質研發領域，多方面發展人工智慧技術及應用，發揮本校作為頂尖研究型大學之優勢。昨日（14日）則邀請學生一同討論「用人類智慧聊聊人工智慧」的活動，實體場次有130多位學生參與，線上同步直播也達175人次加入討論。

18 讓我們用人類智慧來聊聊人工智慧於學習

19 學生討論：人機協力輔助學習

20 針對ChatGPT對於學習的幫助，陽明交大學生以個人經驗指出，對於上課遇到不會的知識，可以給予詳細且通用的答案；也有學生表示其適用於跨領域合作，能夠了解不同領域的基礎知識和術語。

21 然而，並非所有的使用經驗皆為正面，有部分學生反應對於ChatGPT所給予的內容抱持存疑態度，認為在使用過程中需要具備足夠的背景知識、搭配資料檢證的能力，才能判斷內容真假；更有學生指出，ChatGPT無法理解較複雜的學科（如高階統計），因此在使用後反而需要花更多時間扭轉ChatGPT的錯誤。也有學生提出科技應用的資訊安全及隱私的關係，例如：對於上傳的內容之後會被收錄到哪邊，會再做哪些利用，表達憂慮。

22 周倩副校長表示，目前對於生成式人工智慧有許多的討論，今天這場活動以學生為主體分享生成式人工智慧的應用與看法，有別於其他邀請專家學者提供建議的方式。現場活動可以看出陽明交大學生面對新的科技應用並非完全擁抱科技，而是提出疑慮及找出可能的解決方案，顯示學生在面對新的學習科技時，具有更高層次的反思能力，這也是在人才培育中重要的任務。目前陽明交大也成立了生成式人工智慧應用於大學之教與學研議小組，近期也將透過教師及學生的回饋，擬定生成式人工智慧的使用政策及參考指引，作為師生使用的依據。

23 達成共識：態度開放，ChatGPT應部份協助課堂

Demo

# Submission

# Submission

- Add Lab04 status badge in your README file
- Please submit your Github repo `<student_id>-ST-2023`, and upload these to

E3:

- commit URL
  - refer to Lab 1 submission
- github action job URL
  - refer to Lab 3 submission



# Reference

- <https://www.selenium.dev/>
- <https://pypi.org/project/webdriver-manager/>
- <https://www.selenium.dev/selenium/docs/api/py/webdriver/selenium.webdriver.common.by.html>
- <https://selenium-python.readthedocs.io/>

# DOM

Document Object Model

