## Automation Script

## For HOMES BPU Speed Measure Testing

## Using & Maintenance Guide

Program Language:

[*Python 3.10.11*](https://www.python.org/)

3rd Party Python Packages Used:

[*Selenium 4.0*](https://www.selenium.dev/documentation/)*, [openpyxl](https://openpyxl.readthedocs.io/en/stable/)*

IDE suggestion:

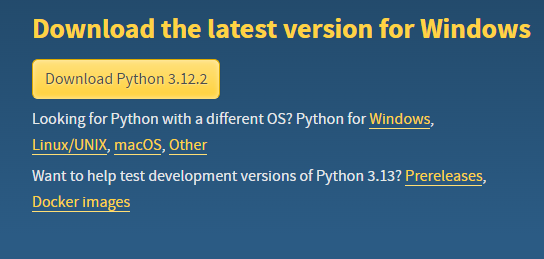
[*PyCharm Community version*](https://www.jetbrains.com/pycharm/?var=1) (free, easy to use)

[*VS Code*](https://code.visualstudio.com/) (open source, free, high expandability, need to set up developing environment by yourself)

### Install the python 3.0 and packages

**Install Python3**

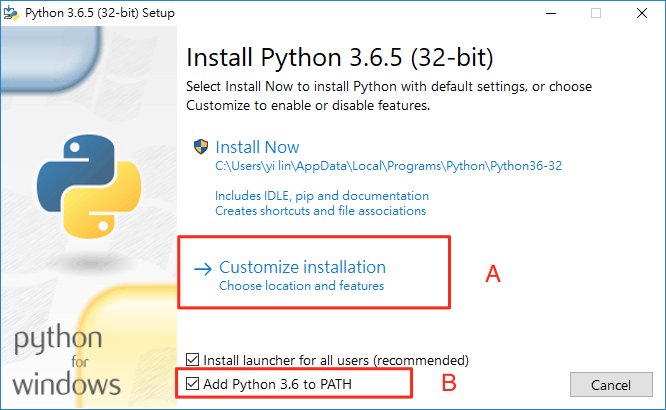
Step1: Go to the python official website [Download Python | Python.org](https://www.python.org/downloads/)

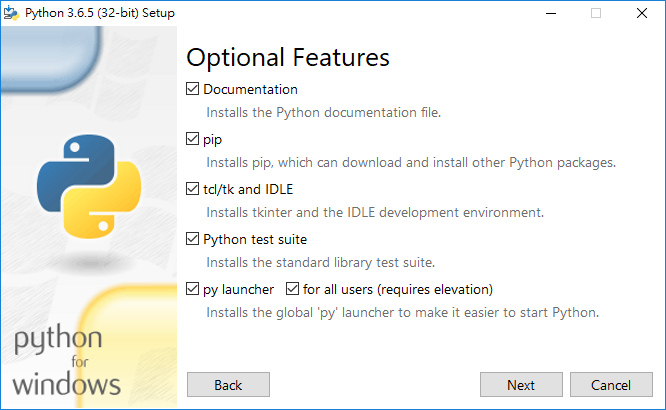


Or you could select the suitable version (recommend the version 3.10) for your PC.

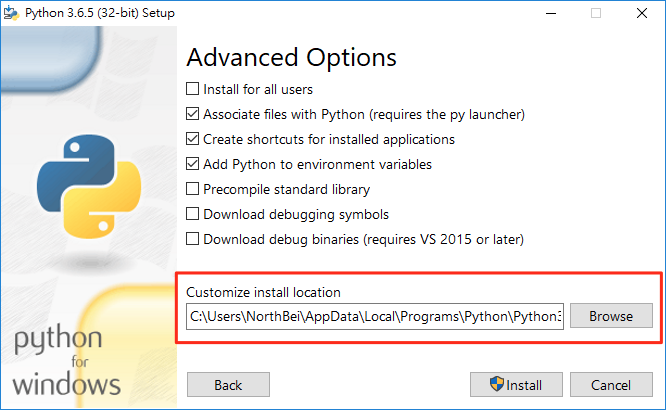


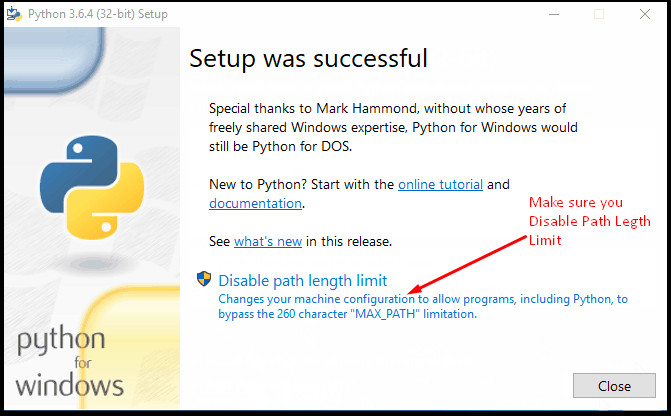
Step2: Follow below Python Set up





Please remember the location you install it



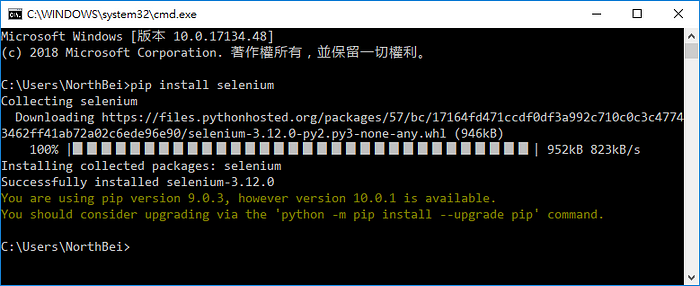


**Install Selenium and openpyxl:**

Open cmd and type below command:

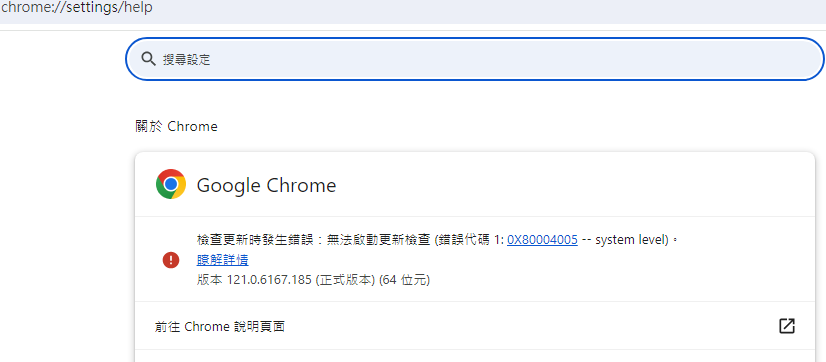
$pip install selenium

$pip install openpyxl

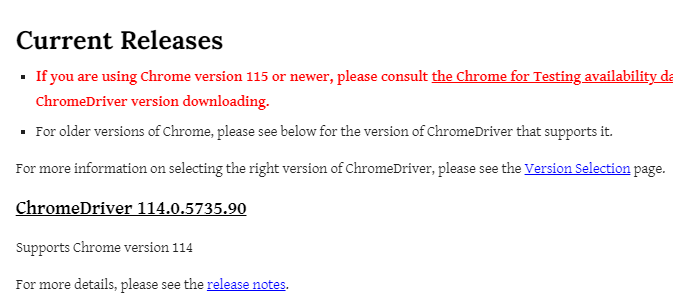


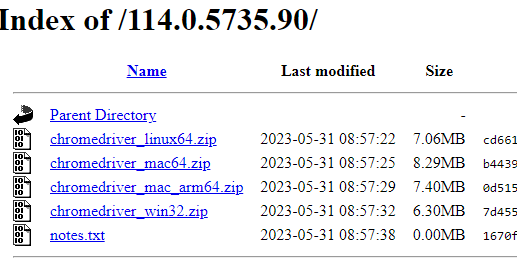
**Install WebDriver:**

Step1: Go to <chrome://settings/help> to check your Chrome version

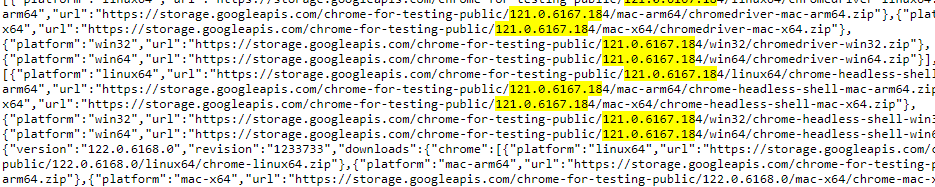


Step2: Go to [ChromeDriver - WebDriver for Chrome - Downloads (chromium.org)](https://chromedriver.chromium.org/downloads) to download the corresponding WebDriver version

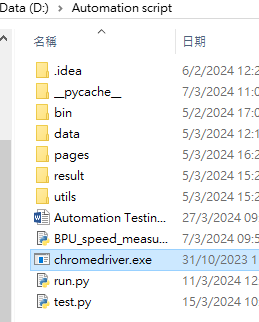




Step3: If didn’t find corresponding version, go to <https://googlechromelabs.github.io/chrome-for-testing/known-good-versions-with-downloads.json> find the corresponding version (no need to be exactly same) and download it



Step4: Put the *chromedriver.exe* in the folder where the main scripts located



### Setup configuration file for selenium

Step1: Go to the C:\Users\ *your PC user name*\.cache\selenium

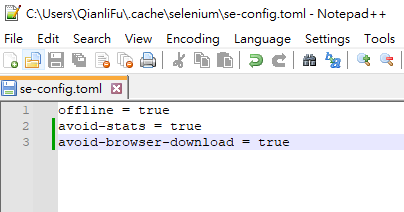


Step2: Input below CLI arguments and save as *se-config.toml*

*offline = true*

*avoid-stats = true*

*avoid-browser-download = true*



Explanation:

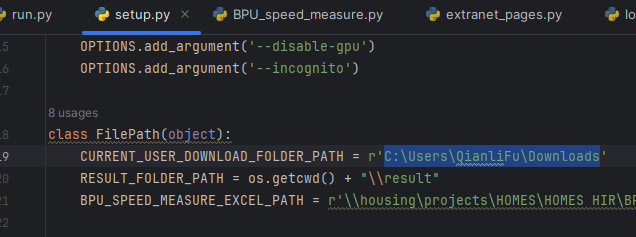
*offline = true* means offline mode (i.e., disabling network requests and downloads)

*avoid-browser-download = true* means avoid to download browser, e.g., when a browser is supposed to be downloaded by Selenium Manager, but you prefer to avoid it

*avoid-browser-download = true* means avoid sends usage statistics to plausible.io. Default: false

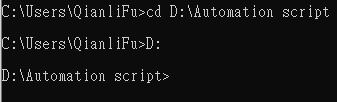
You could see more details on the official website: [Selenium Manager (Beta) | Selenium](https://www.selenium.dev/documentation/selenium_manager/)

Step 3: Go to *setup.py* -> *FilePath* class -> find and modify *CURRENT\_USER\_DOWNLOAD\_FOLDER\_PATH* to your PC downloading folder path

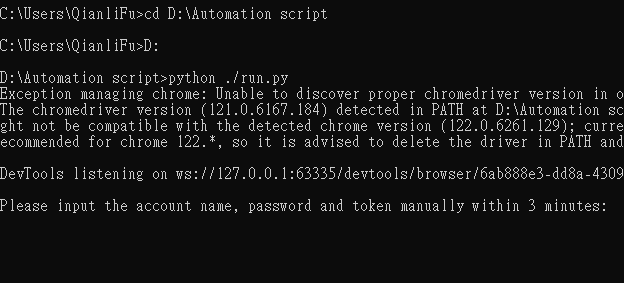


### Run the automation script for testing

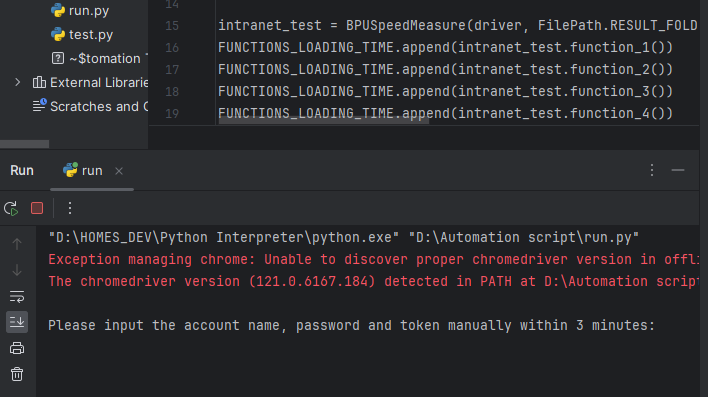
Step1: Go to the directory where *run.py* is located



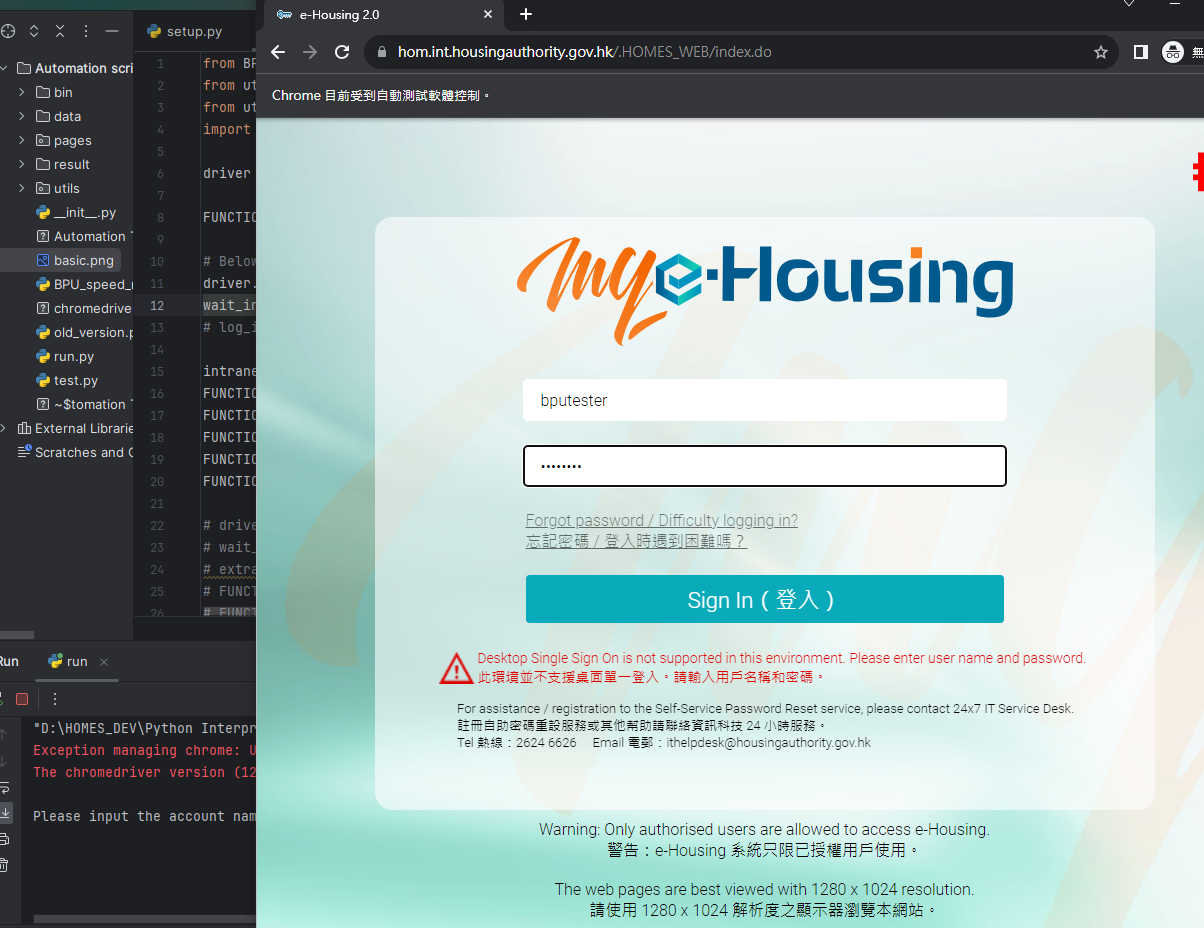
Step2: Run the *run.py* file



Or you could use an IDE to run it:



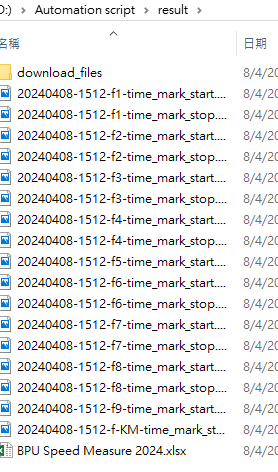
Step3: Manually input the user name, password and token on the pop out window. Then click login and the script will automatically run. (same as login the extranet HOMES for testing function6-10)



Step 4:

Go to the *result* folder and check whether there exist any abnormal things.

If no, update the result.



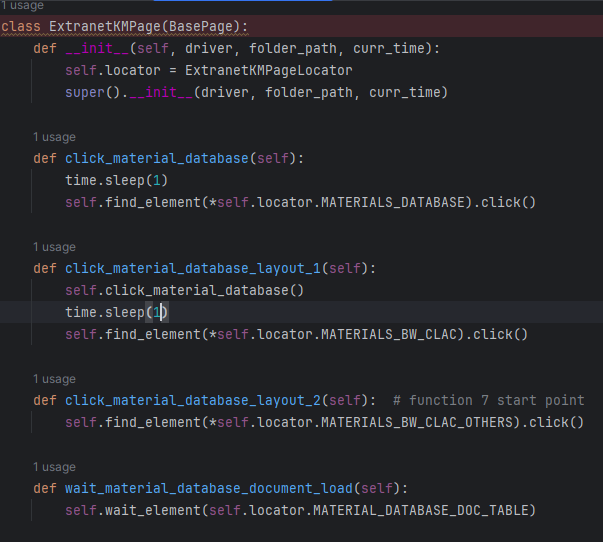
### Maintenance guideline

The *callgraph.png* contains the relations of calling functions in this script.

This automation script using pattern named page-object-model (POM) to minimize the maintenance cost.

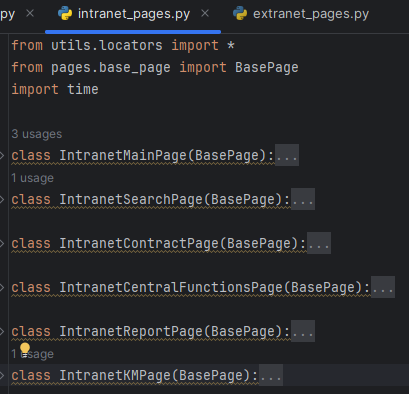
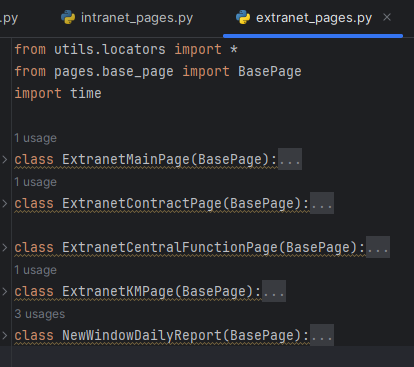
**The *pages* folder:**

This pattern tread each page as an object, and for each specific page it has a corresponding class (inherited from *BasePage* class) to store the operations that you wish to do on this page. For example the *ExtranetKMPage* contains 3 click operations and 1 wait operation and the *locators(will be instruct later)* for all above operations.

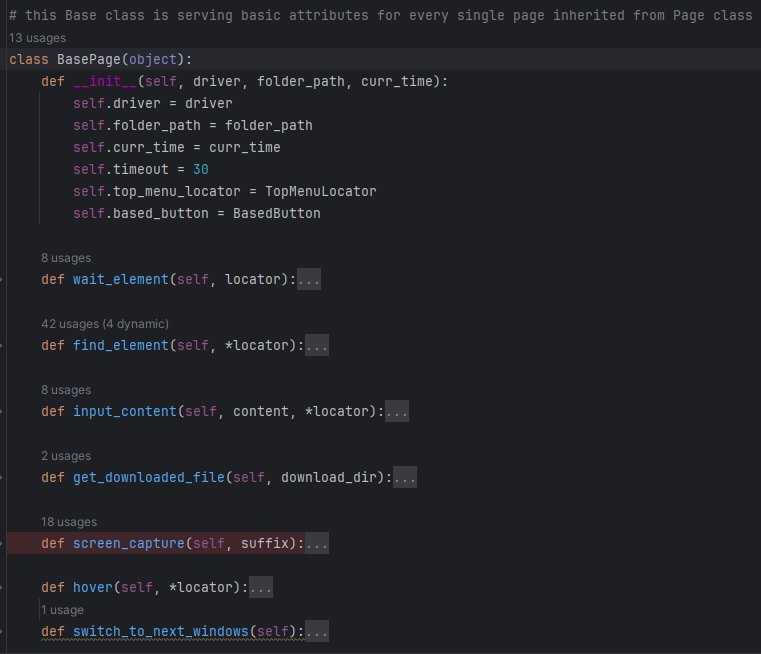


The *extranet\_pages.py* contains 4 specific extranet page classes and

The *intranet\_pages.py* contains 6 specific intranet page classes.

For *base\_page.py*, it include all the basic functions which will be used in every pages



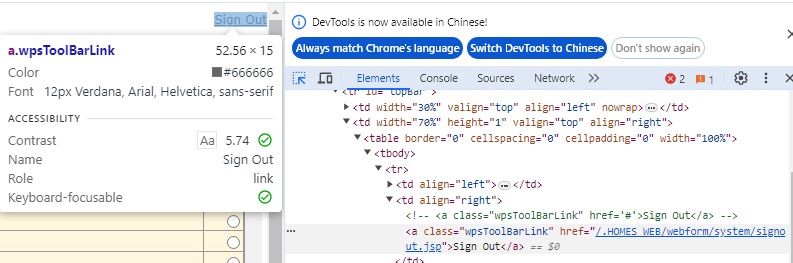
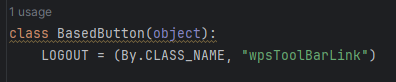
**The *utils* folder:**

Include *excel\_operations.py, locators.py, setup.py.* Itis the main maintenance part, if no operations need to be added.

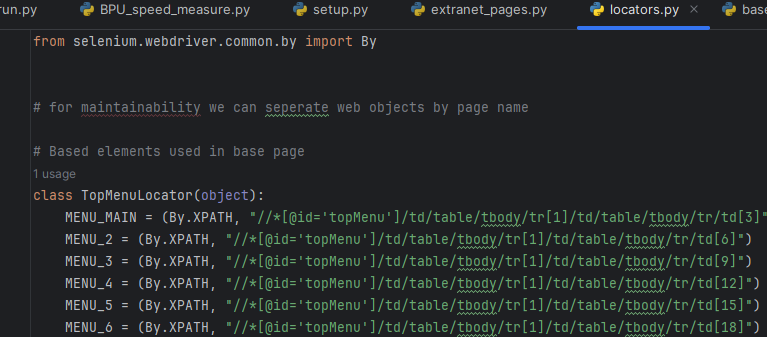
For *locators.py*, it contains all the locator to each corresponding HTML elements.

Notice: if the HTML file changed the *locator.py* needs to be modified correspondingly

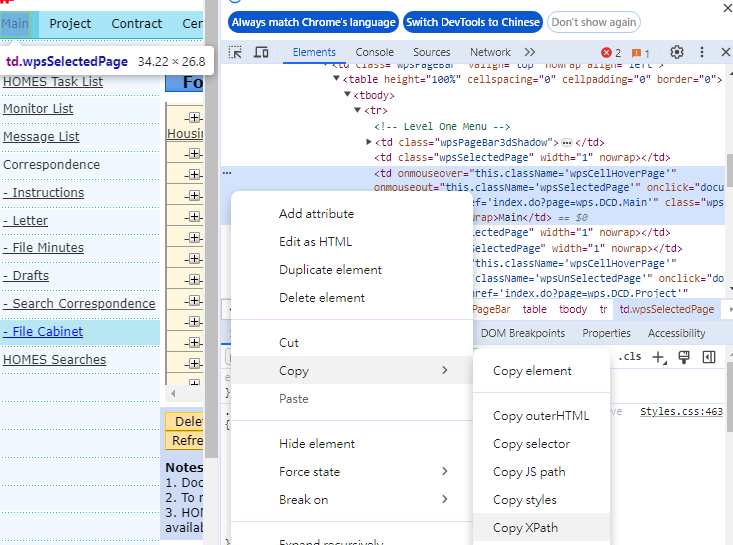
For example, the *LOGOUT* in *BasedButton* class represents the HTML elements of the logout button:



Another example, the *MENU\_MAIN* represents the HTML elements of the top menu navigator bar (main)



The way to find the XPath is right click->copy->copy XPath

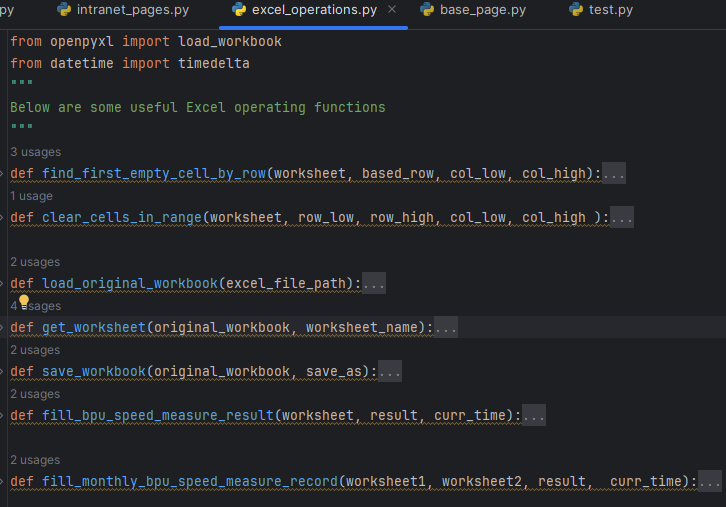


For *setup.py* it contains all the setup setting, including the browser options setting (more information please click [Browser Options | Selenium](https://www.selenium.dev/documentation/webdriver/drivers/options/)), URL address, PC file path, and some input contents.



For *excel\_operations.py* it contains all the necessary functions for operate excel files.

Need import *openpyxl*

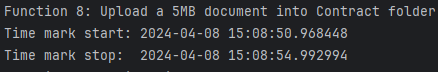
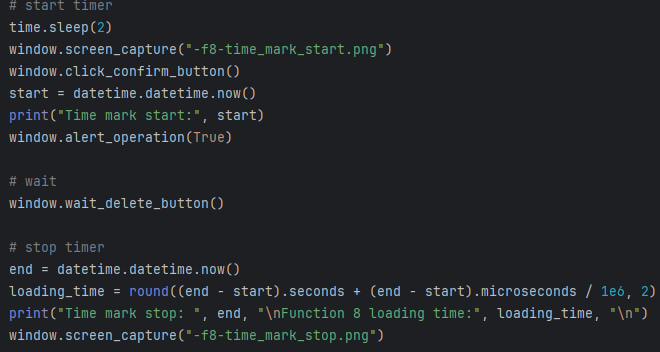


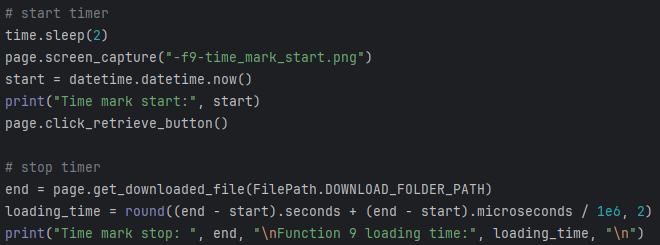
**For *BPU\_speed\_measure.py*:**

Apply the functions(operations) in *pages* folder to complete the BPU speed measure.

For each speed loading test it will calculate the loading time by timer stop time moment (often a click shows in blue) minus the timer start time moment(often wait some elements appear shows in blue).

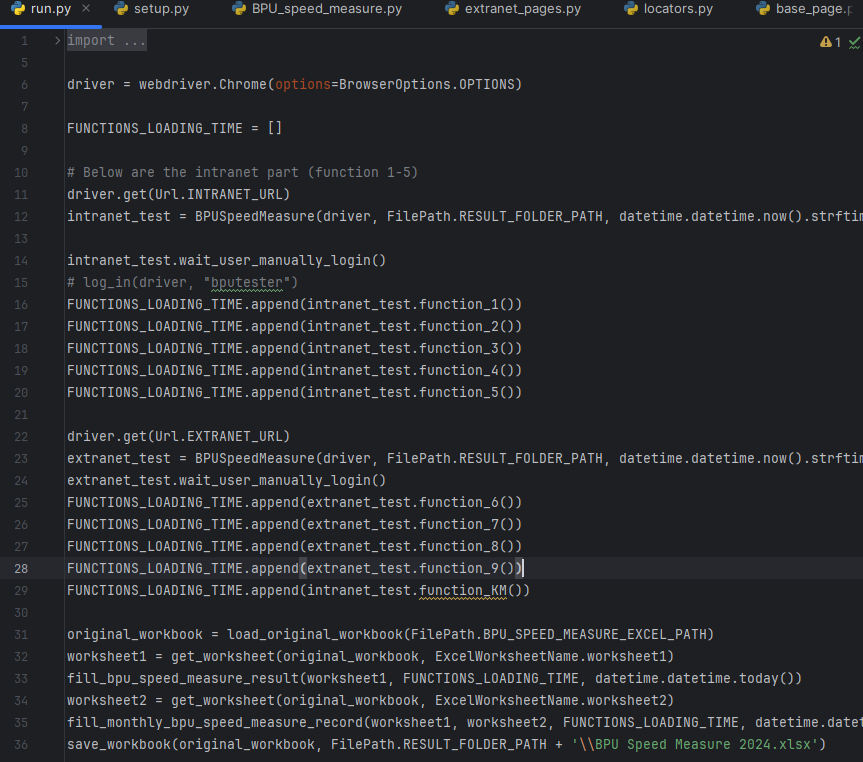
Normally one function will take two screen captures(shows in yellow) to give the evidence of timer setting correctly, one is timer start and one is timer stop.



But for f5, f9 and function KM, the timer stop time moment is a file been download(shows in blue) the calling function *get\_downloaded\_file* return the last modify time moment of the news file in the *result/download\_file* folder so there is only one screen capture here(shows in yellow)****

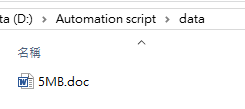
**For run.py:**

Apply all the BPU speed measure functions and store the data to the result excel file. Common cases no need to modify it.



**For *data* folder:**

Include the non-programing files for BPU speed measure, like the testing 5MB file



**For *result* folder:**

Include the generated result, mainly are screen captures, excel file and download files(download in f5, f9 and function KM).

