**Installation :**

* Install Python on Windows

<https://www.python.org/downloads/>

1. Download the latest version of python and install python.

C🡪 Program files(x 86) 🡪 Python X.XX

Set System properties :

Environment Variable 🡪 Path🡪 Edit – Add

( C:Program files(x 86)/ Python X.XX)

( C:Program files(x 86)/ Python X.XX/Scripts)

1. Validate if python installed correctly.

Go to cmd prompt🡪 python (enter)

Version which is installed should be displayed.

* Install PyCharm IDE

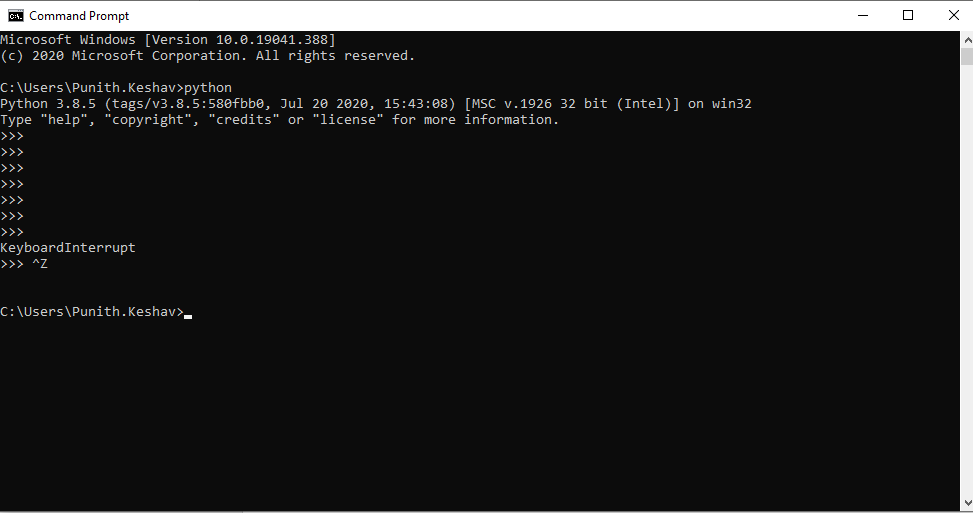
<https://www.jetbrains.com/pycharm/>

Install community Edition of PyCharm

* Install Selenium with Python

Install Selenium packages in Python

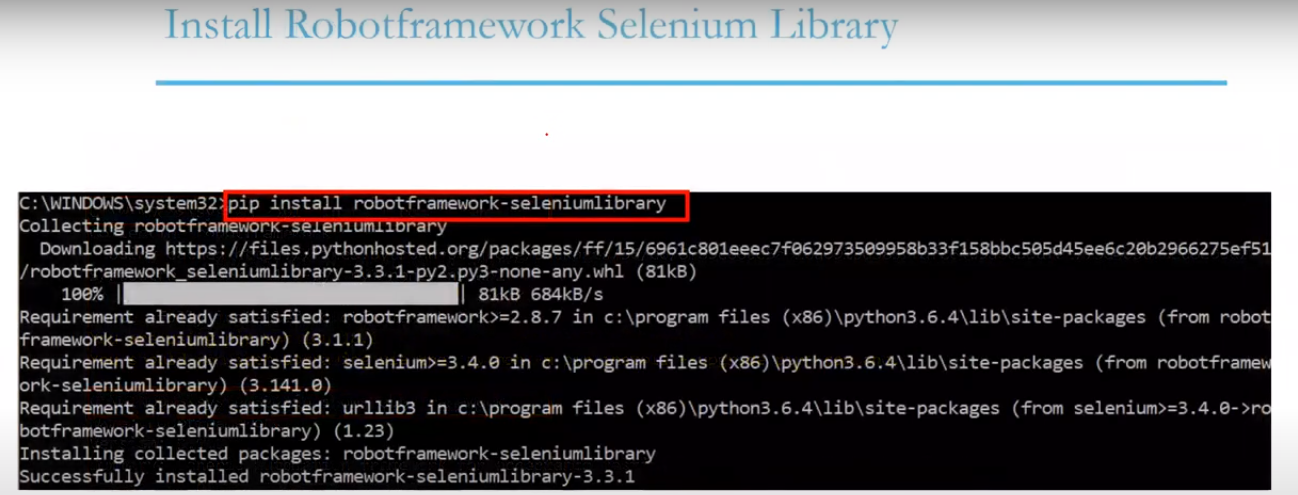
cmd prompt 🡪 control c enter and control z to come out of python prompt



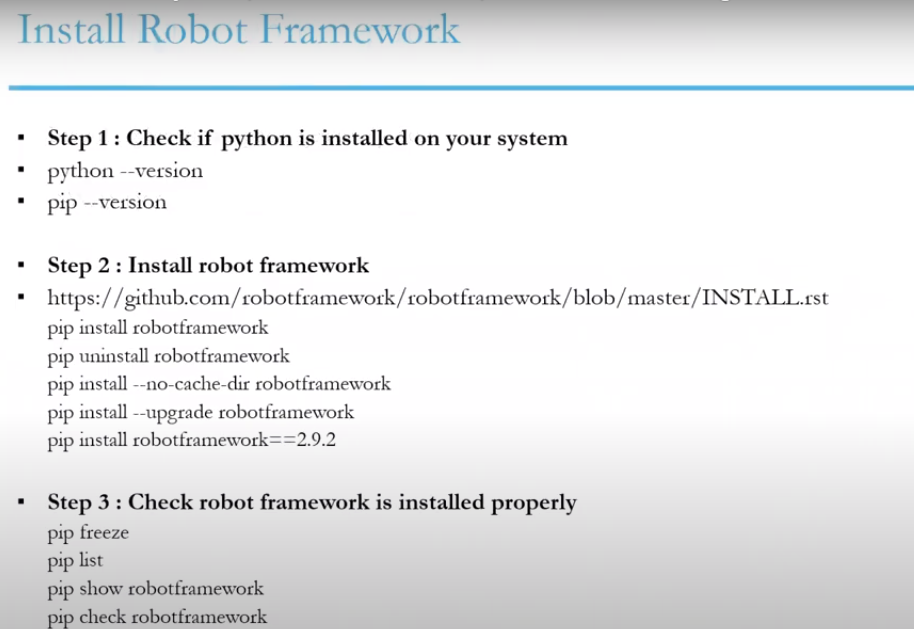
Pip list 🡪 List of installed items on python

Pip install -U selenium 🡪 Installs selenium on python

* Install Robot Framework Selenium Library



* Configure Pycharm with Selenium
* Robot Framework
  + Install Robot Framework
  + Install Robot Framework Selenium Library
  + Install Intellibot plugin



Pip List :

(venv) C:\Users\Punith.Keshav\PycharmProjects\pythonProject>pip list

Package Version

---------------------------------- ---------

allure-pytest 2.8.18

allure-pytest-bdd 2.8.18

allure-python-commons 2.8.18

allure-robotframework 2.8.18

atomicwrites 1.4.0

attrs 20.1.0

Babel 2.8.0

BDSSeleniumLibrary 0.0.2

certifi 2020.6.20

chardet 3.0.4

colorama 0.4.3

docutils 0.16

enum34 1.1.6

et-xmlfile 1.0.1

Faker 4.14.0

glob2 0.7

idna 2.10

iniconfig 1.0.1

jdcal 1.4.1

lxml 4.5.2

Mako 1.1.3

MarkupSafe 1.1.1

more-itertools 8.4.0

namedlist 1.8

natsort 5.2.0

numpy 1.19.2

openpyxl 2.5.1

packaging 20.4

panda 0.3.1

parse 1.16.0

parse-type 0.5.2

pip 20.2.4

pluggy 0.13.1

py 1.9.0

Pygments 2.6.1

pyparsing 2.4.7

pytest 6.0.1

pytest-allure-adaptor 1.7.10

pytest-bdd 3.4.0

python-dateutil 2.8.1

pytz 2020.1

requests 2.24.0

robotframework 3.2.2

robotframework-allurereport 1.1.1

robotframework-datadriver 0.3.6

robotframework-datetime-tz 1.0.6

robotframework-docker 0.1.0

robotframework-excel 1.0.0b4

robotframework-faker 5.0.0

robotframework-flexseleniumlibrary 0.3.4

robotframework-pabot 1.10.0

robotframework-pythonlibcore 2.1.0

robotframework-seleniumlibrary 4.5.0

selenium 3.141.0

SeleniumLibraryExtension 0.0.7

setuptools 49.6.0

six 1.11.0

text-unidecode 1.3

toml 0.10.1

urllib3 1.25.10

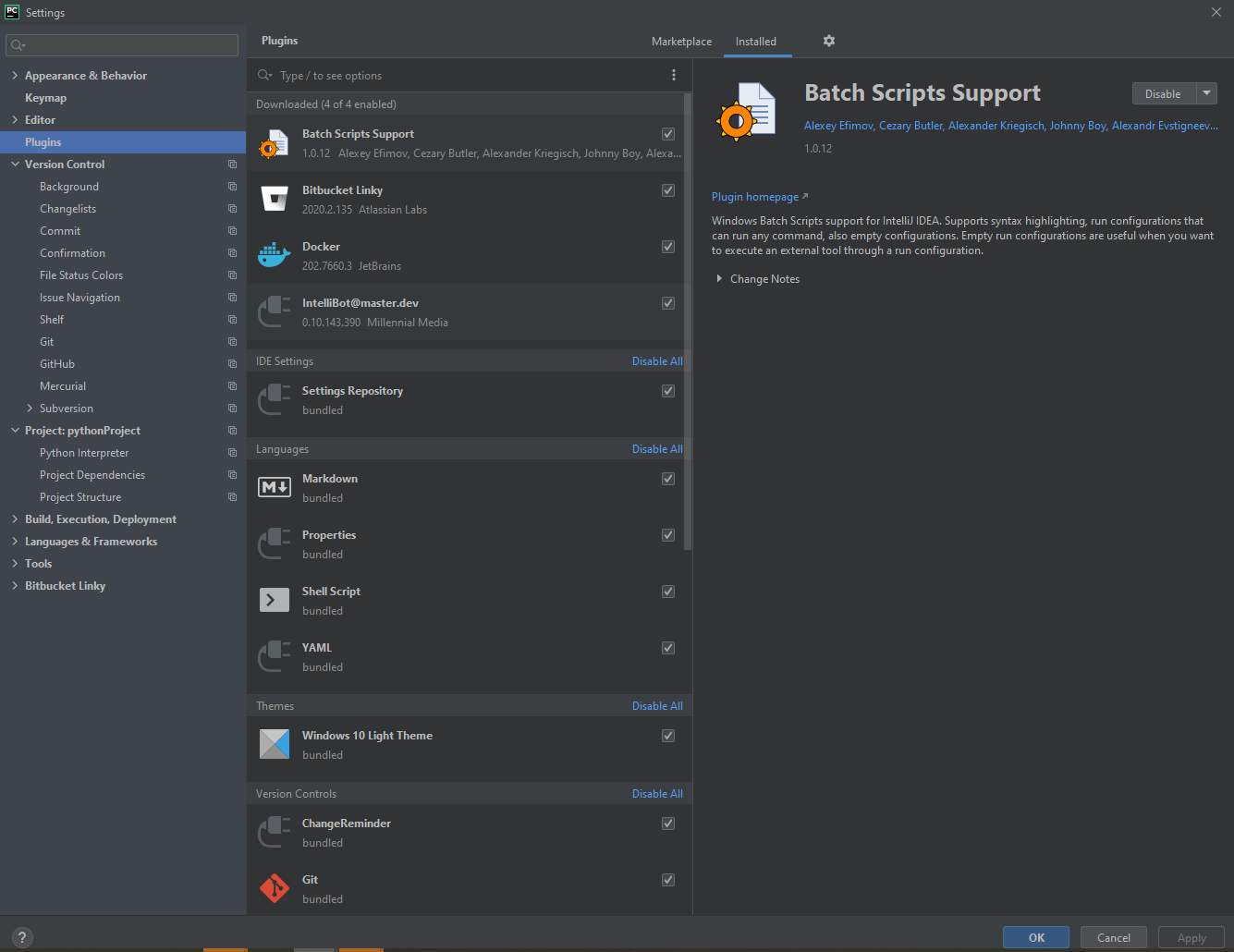
wrapt 1.12.1

xlrd 1.2.0

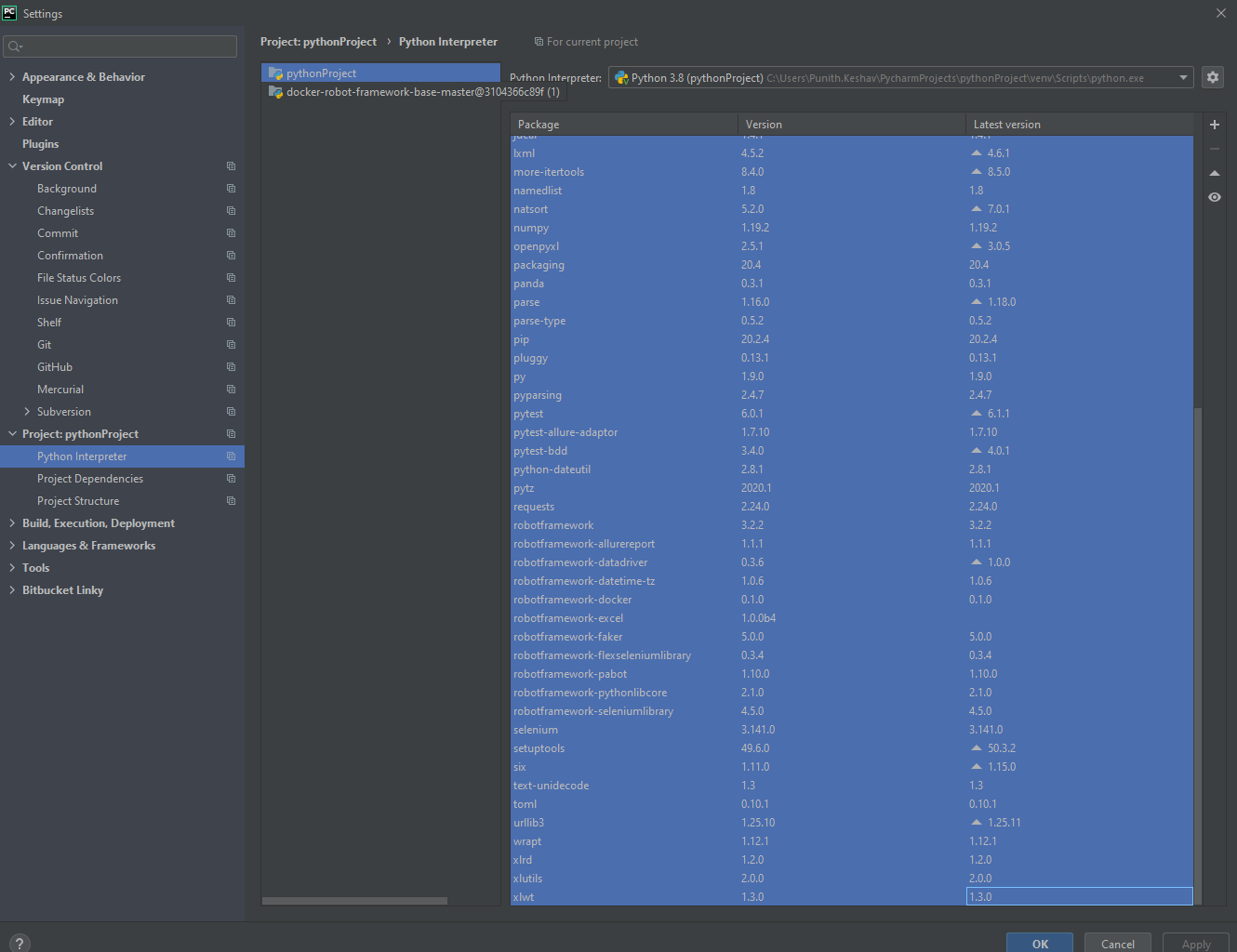
xlutils 2.0.0

xlwt 1.3.0

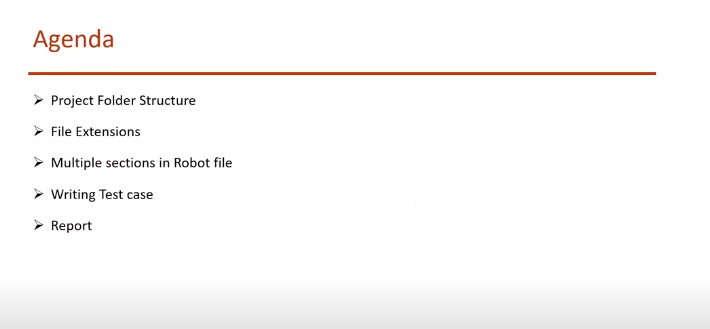
PyCharm Plugins:



|  |  |  |
| --- | --- | --- |
| BDSSeleniumLibrary | 0.0.2 | 0.0.2 |
| Babel | 2.8.0 | 2.8.0 |
| Faker | 4.14.0 | 4.14.0 |
| Mako | 1.1.3 | 1.1.3 |
| MarkupSafe | 1.1.1 | 1.1.1 |
| Pygments | 2.6.1 | 2.7.2 |
| SeleniumLibraryExtension | 0.0.7 | 0.0.7 |
| allure-pytest | 2.8.18 | 2.8.18 |
| allure-pytest-bdd | 2.8.18 | 2.8.18 |
| allure-python-commons | 2.8.18 | 2.8.18 |
| allure-robotframework | 2.8.18 | 2.8.18 |
| atomicwrites | 1.4.0 | 1.4.0 |
| attrs | 20.1.0 | 20.2.0 |
| certifi | 2020.6.20 | 2020.6.20 |
| chardet | 3.0.4 | 3.0.4 |
| colorama | 0.4.3 | 0.4.4 |
| docutils | 0.16 | 0.16 |
| enum34 | 1.1.6 | 1.1.10 |
| et-xmlfile | 1.0.1 |  |
| glob2 | 0.7 | 0.7 |
| idna | 2.10 | 2.10 |
| iniconfig | 1.0.1 | 1.1.1 |
| jdcal | 1.4.1 | 1.4.1 |
| lxml | 4.5.2 | 4.6.1 |
| more-itertools | 8.4.0 | 8.5.0 |
| namedlist | 1.8 | 1.8 |
| natsort | 5.2.0 | 7.0.1 |
| numpy | 1.19.2 | 1.19.2 |
| openpyxl | 2.5.1 | 3.0.5 |
| packaging | 20.4 | 20.4 |
| panda | 0.3.1 | 0.3.1 |
| parse | 1.16.0 | 1.18.0 |
| parse-type | 0.5.2 | 0.5.2 |
| pip | 20.2.4 | 20.2.4 |
| pluggy | 0.13.1 | 0.13.1 |
| py | 1.9.0 | 1.9.0 |
| pyparsing | 2.4.7 | 2.4.7 |
| pytest | 6.0.1 | 6.1.1 |
| pytest-allure-adaptor | 1.7.10 | 1.7.10 |
| pytest-bdd | 3.4.0 | 4.0.1 |
| python-dateutil | 2.8.1 | 2.8.1 |
| pytz | 2020.1 | 2020.1 |
| requests | 2.24.0 | 2.24.0 |
| robotframework | 3.2.2 | 3.2.2 |
| robotframework-allurereport | 1.1.1 | 1.1.1 |
| robotframework-datadriver | 0.3.6 | 1.0.0 |
| robotframework-datetime-tz | 1.0.6 | 1.0.6 |
| robotframework-docker | 0.1.0 | 0.1.0 |
| robotframework-excel | 1.0.0b4 |  |
| robotframework-faker | 5.0.0 | 5.0.0 |
| robotframework-flexseleniumlibrary | 0.3.4 | 0.3.4 |
| robotframework-pabot | 1.10.0 | 1.10.0 |
| robotframework-pythonlibcore | 2.1.0 | 2.1.0 |
| robotframework-seleniumlibrary | 4.5.0 | 4.5.0 |
| selenium | 3.141.0 | 3.141.0 |
| setuptools | 49.6.0 | 50.3.2 |
| six | 1.11.0 | 1.15.0 |
| text-unidecode | 1.3 | 1.3 |
| toml | 0.10.1 | 0.10.1 |
| urllib3 | 1.25.10 | 1.25.11 |
| wrapt | 1.12.1 | 1.12.1 |
| xlrd | 1.2.0 | 1.2.0 |
| xlutils | 2.0.0 | 2.0.0 |
| xlwt | 1.3.0 | 1.3.0 |

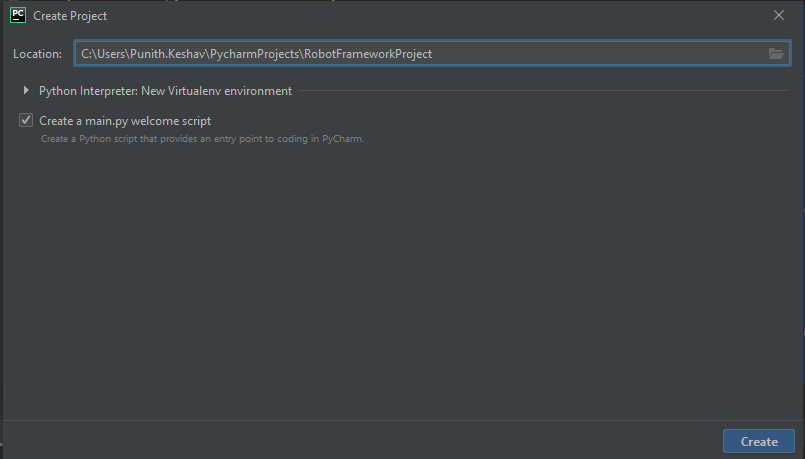


**Test Case in Robot Framework:**



Create New Project:

File 🡪New Project 🡪 Name of the Project 🡪 Create

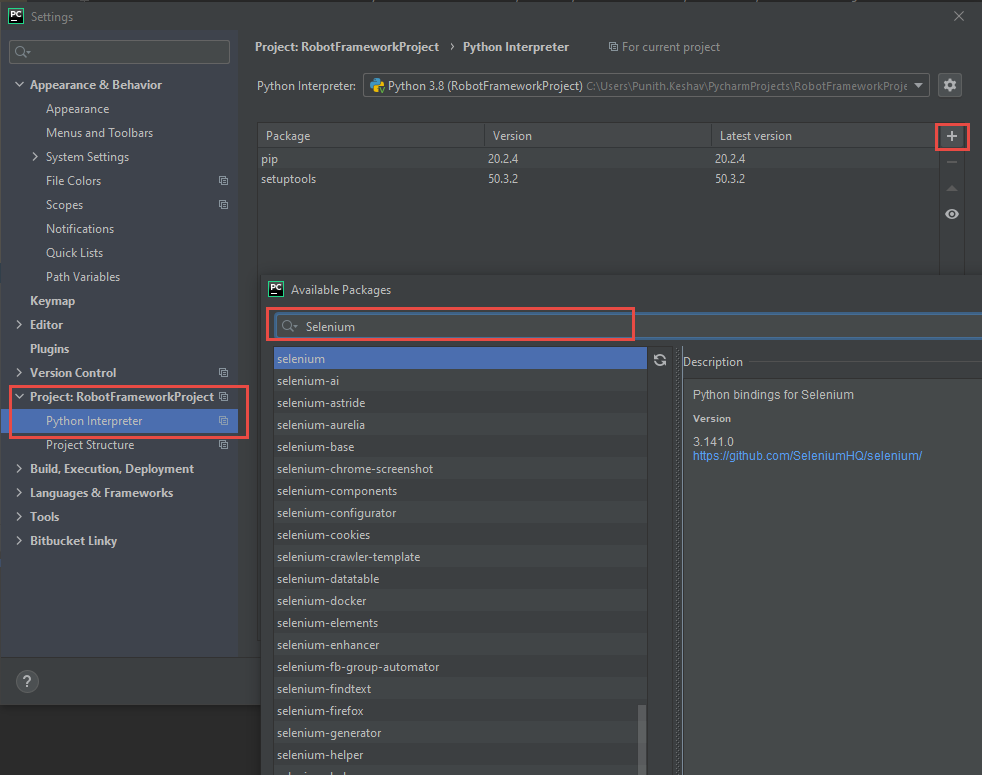


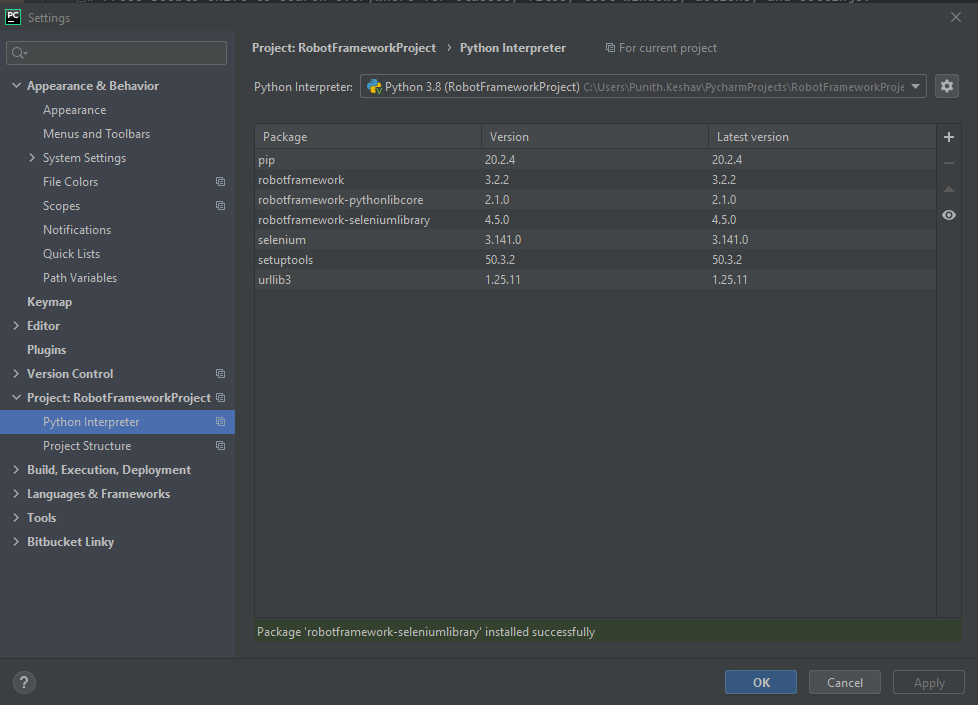
We need to add 3 different packages to this project to start with test cases

File 🡪 Settings 🡪 Project 🡪Project Interpreter 🡪 Add button 🡪 Search for packages to add 🡪Select the package and click Install Packages

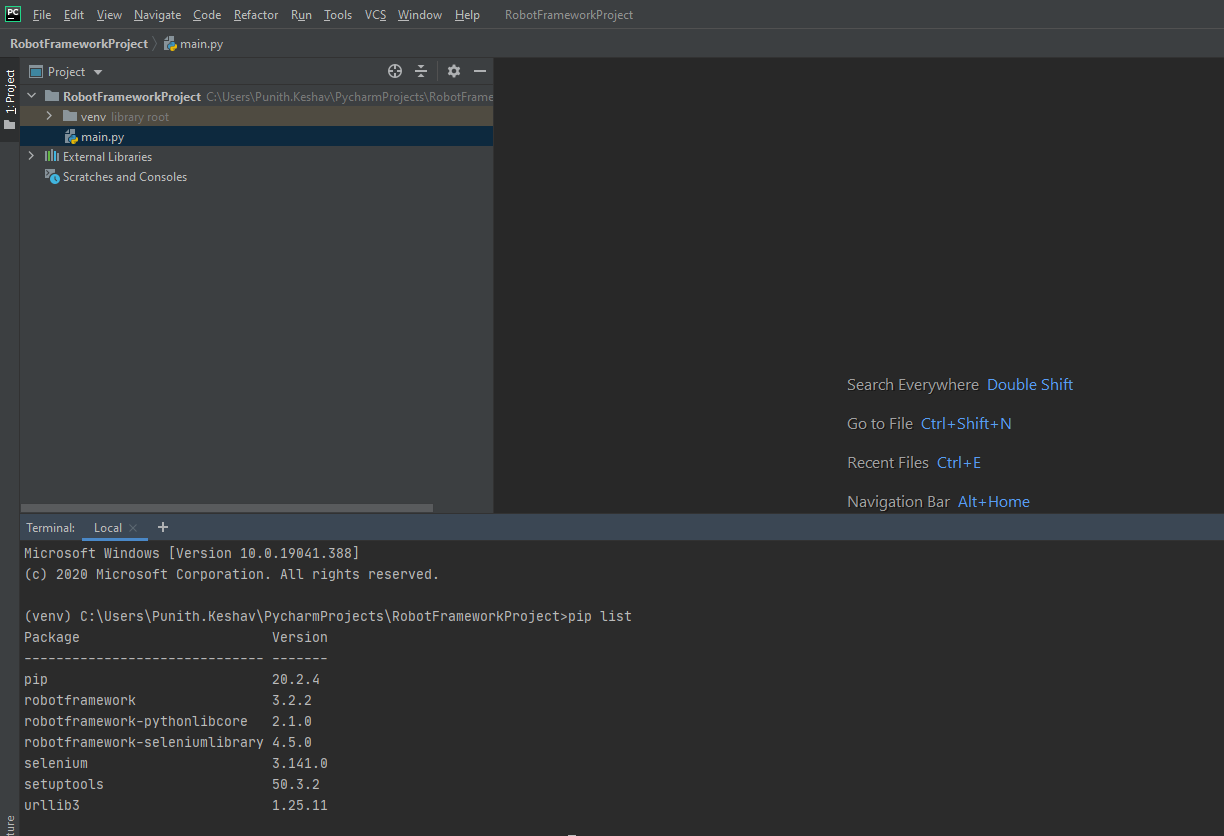
* Selenium Package -- selenium
* Robot Framework Package-- RobotFramework
* Robot Framework Selenium Library 🡪 robotframework-seleniumlibrary

<https://pypi.org/search/?q=RobotFramework> 🡪 Link to see all the packages





To validate installed packages in the terminal 🡪 pip list



Files and extension used in project :

Test Cases 🡪 TestCaseName**.robot**

Excel 🡪 .xls

Notepad 🡪 .txt

Python 🡪 .py

Test Directory:

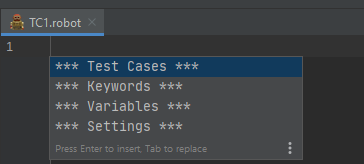
Right click on project 🡪 New 🡪Directory 🡪 Name

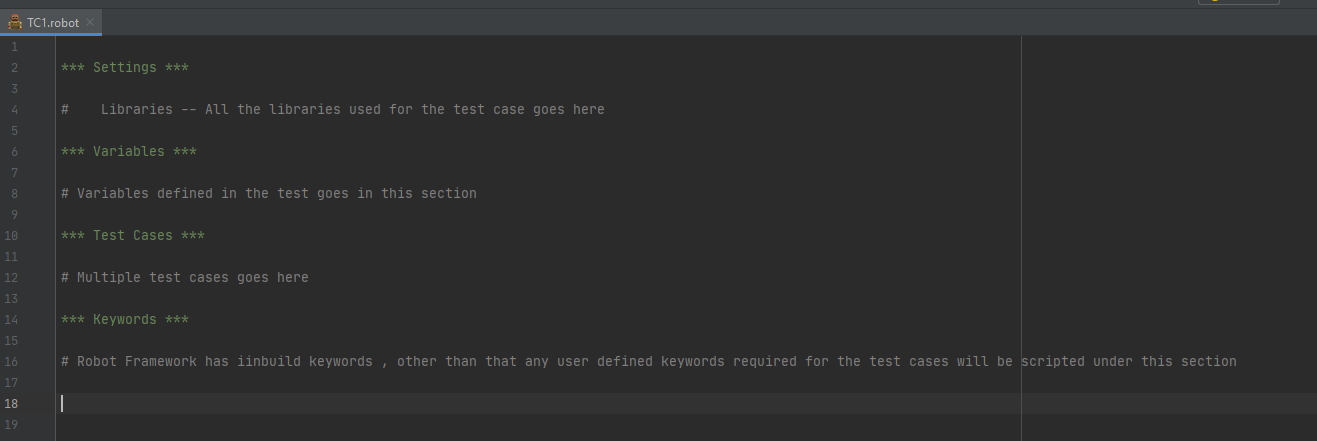
Test Case:

Right click on project 🡪 New 🡪New File 🡪 TC1.robot

Sections under Robot Test case file:

Control + Space Bar 🡪 to see the sections in Test Cases File

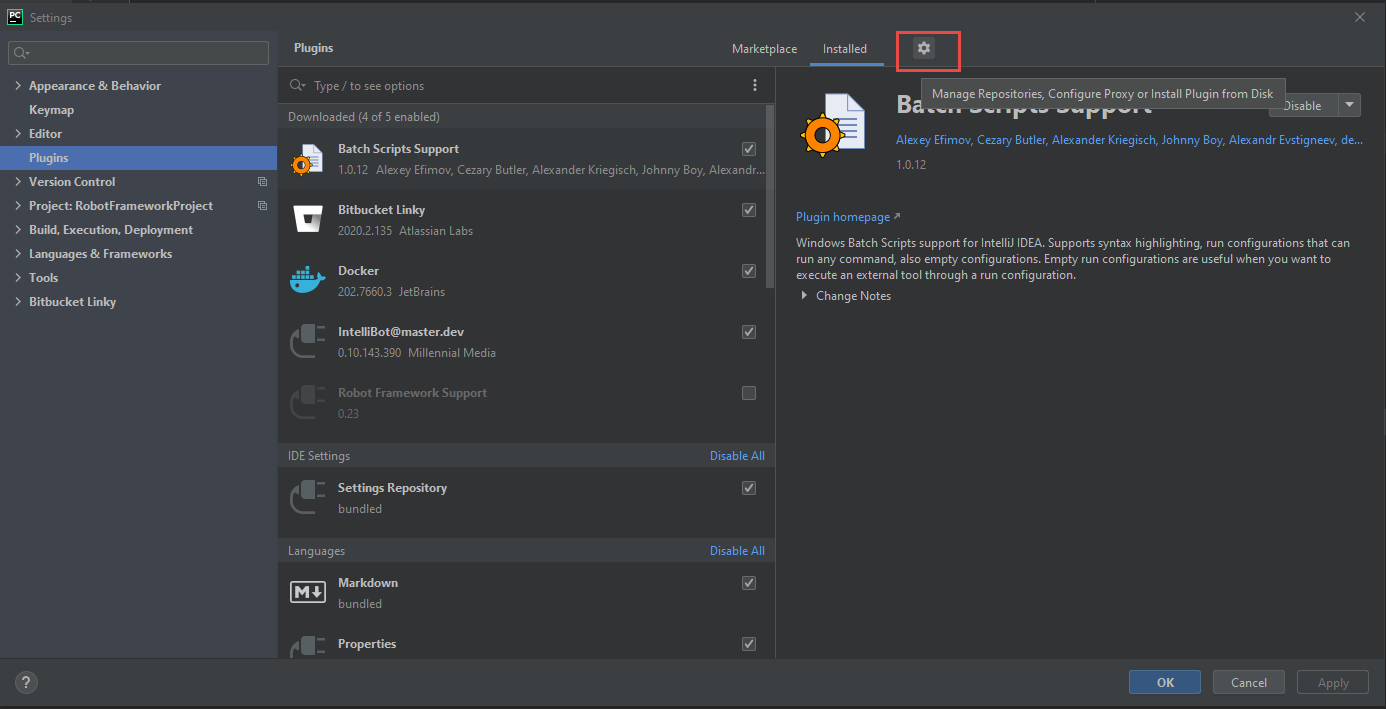


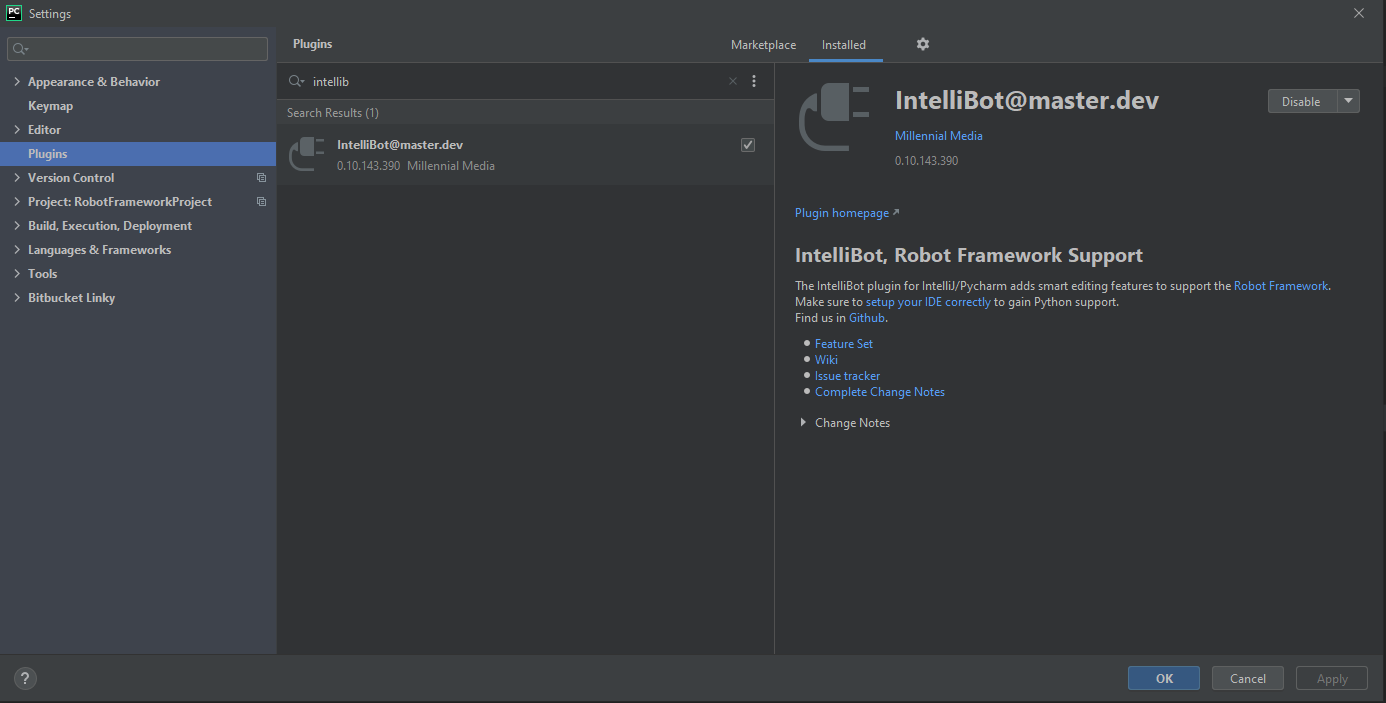


Autosuggestions : Intellibot Plugin

Download Intellibot : <https://github.com/lte2000/intellibot>

File 🡪 Settings 🡪Plugins 🡪 Manage Repositories, configure proxy or Install Plugin from disk 🡪Install Plugin from Disk 🡪( Select Intellibot download )

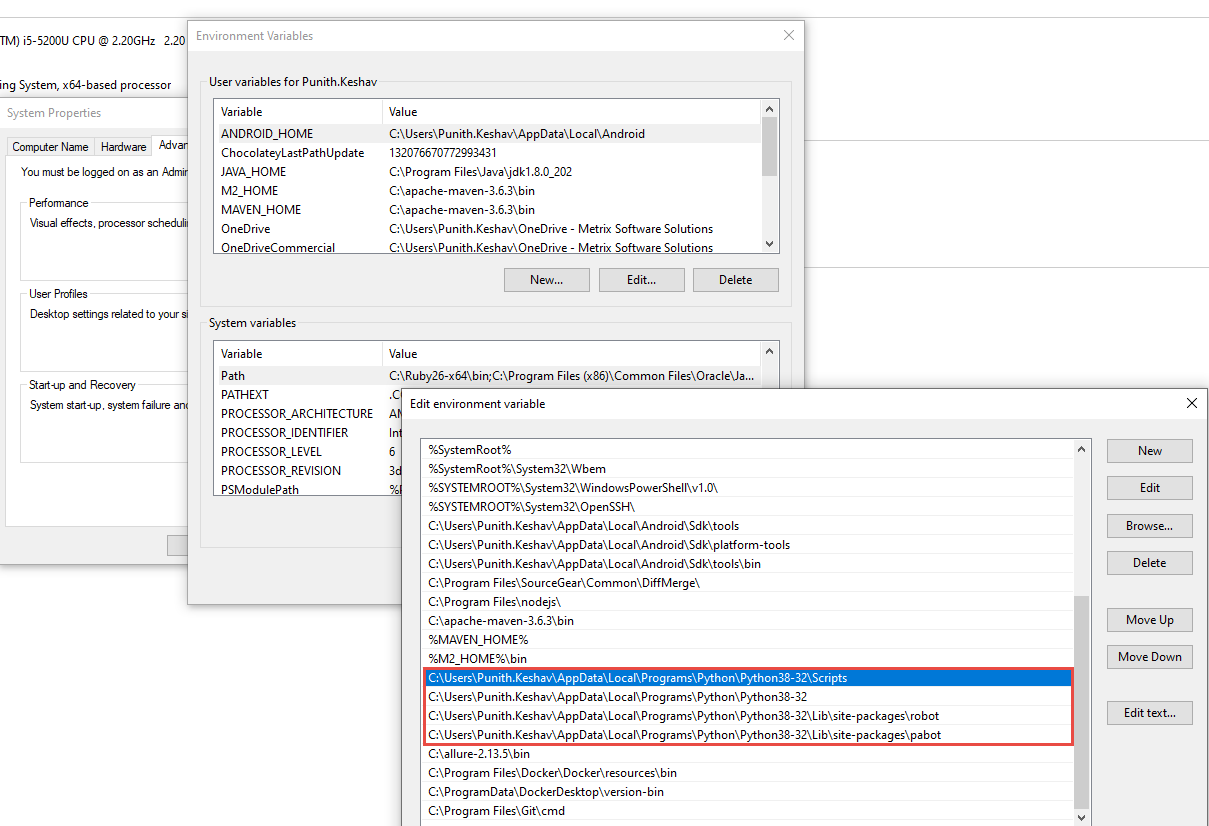




**Webdriver:**

All the web drivers to be placed in python folder 🡪 scrips folder.

And this path should be set in system path variables.



WebDriver can be downloaded from below link,

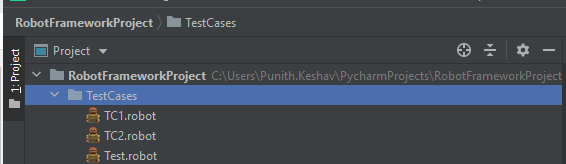
<https://www.selenium.dev/documentation/en/webdriver/driver_requirements/>

Chrome Driver :

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

**Test Cases in RobotFramework:**

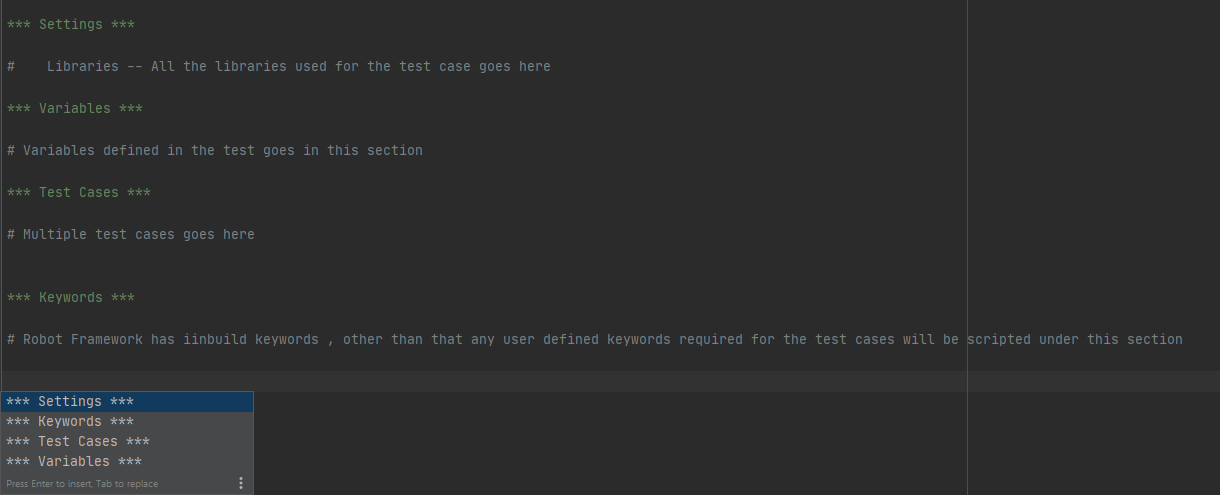
Test cases created should always end with .robot



Syntax inside Test File.

To see all the sections under robot file

**Control + Spacebar**



Not every sections in this robot file are mandatory

**Settings :**

All the libraries used for particular test case will be added under settings

Import libraries, Add Resources, Test SetUp and Teardown

\*\*\*Settings\*\*\*

***Library*** *SeleniumLibrary****Library*** *OperatingSystem*

**Variables:**

# Variables defined in the test goes in this section

\*\*\*Variables\*\*\*

**${BROWSER}**= ie**${URL}**= https://wordpress.com/**${GET\_STARTED}**= //a[@title='Get Started']

**Test Cases** :

This represents a flow of specific test cases

**\*\*\* Test Cases \*\*\***

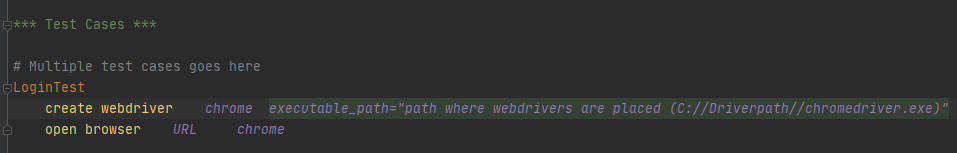
**Navigate to Login Page** Open WordPress

In built Keywords are found in the below link which are widely used in test cases.

<https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>

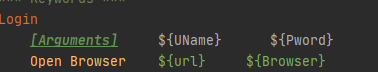
**Browser Initiation:**

**Step 1:**



**Alternative Step**

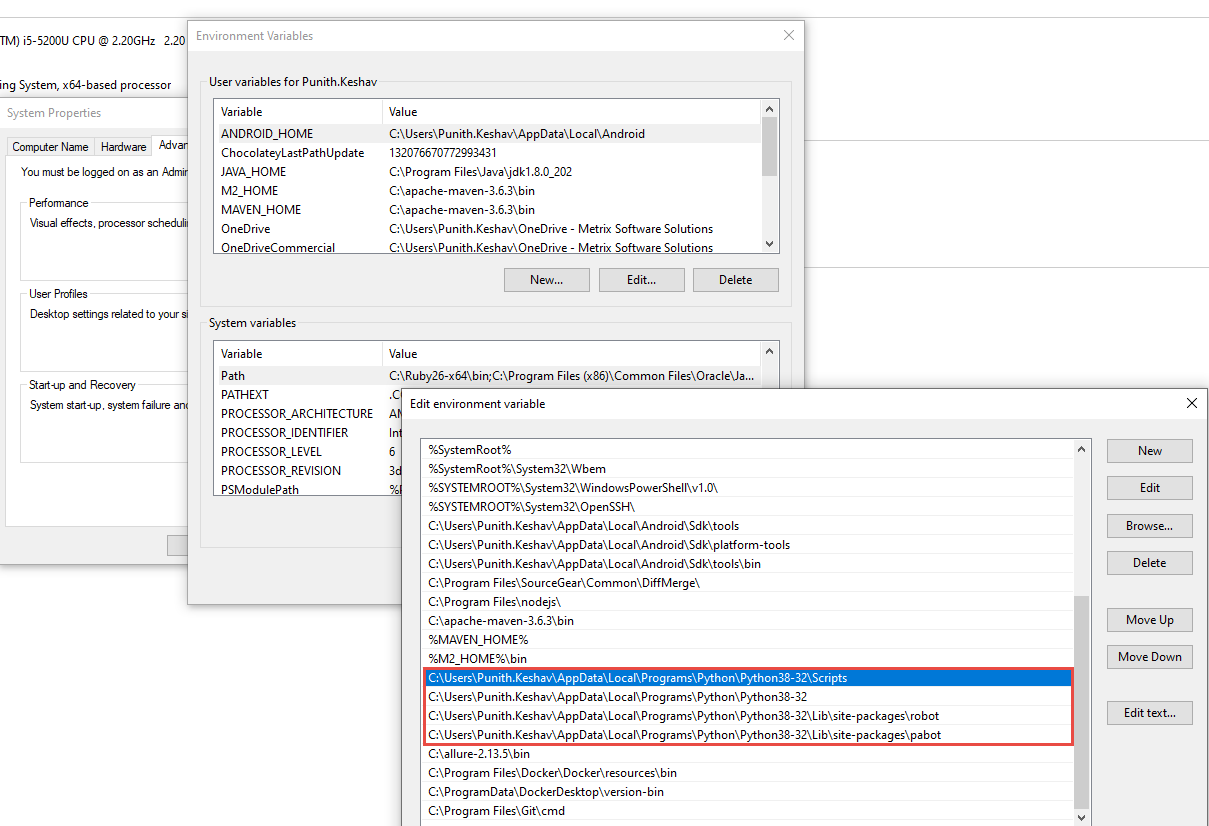
In order to do this we should copy drivers in script folder as shown below



**Webdriver:**

All the web drivers to be placed in python folder 🡪 scrips folder.

And this path should be set in system path variables.



WebDriver can be downloaded from below link,

<https://www.selenium.dev/documentation/en/webdriver/driver_requirements/>

Chrome Driver :

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

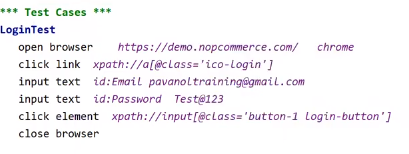
Keywords:

*Keywords are used to write a logic to perform specific task*

*For example, Enter UID, PWD and Click on Login button*

**\*\*\* Keywords \*\*\***

**Open WordPress** Open Browser ${URL} ${BROWSER}  
 sleep *5s  
 #Maximize Browser Window* Wait Until Page Contains Element ${GET\_STARTED} *5s* Close Browser  
 sleep *2s*



**Command to execute this script in Terminal:**

Execute this script through Pycharm terminal (**To open Terminal** : View –> Tools window –> Terminal)

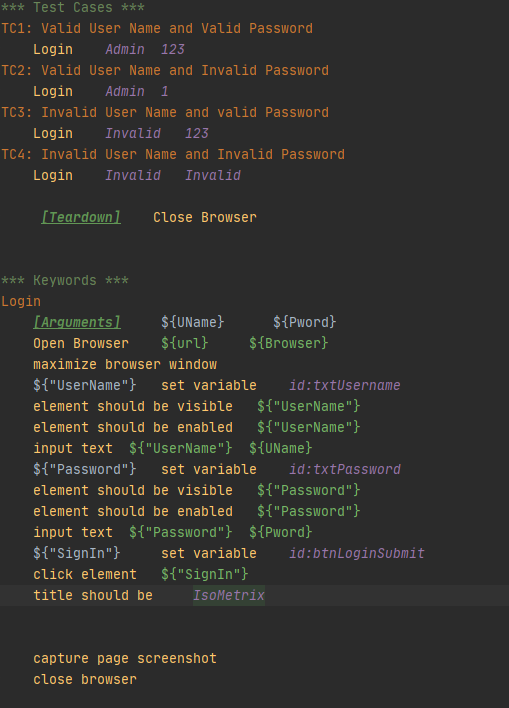
**Command:***robot -d Results TestScripts\FirstScript.robot*

**Note:**

* **-d results** – will save the test results in Results directory
* ***TestScripts\FirstScript.robot****– path of the test script*

**Keywords:**

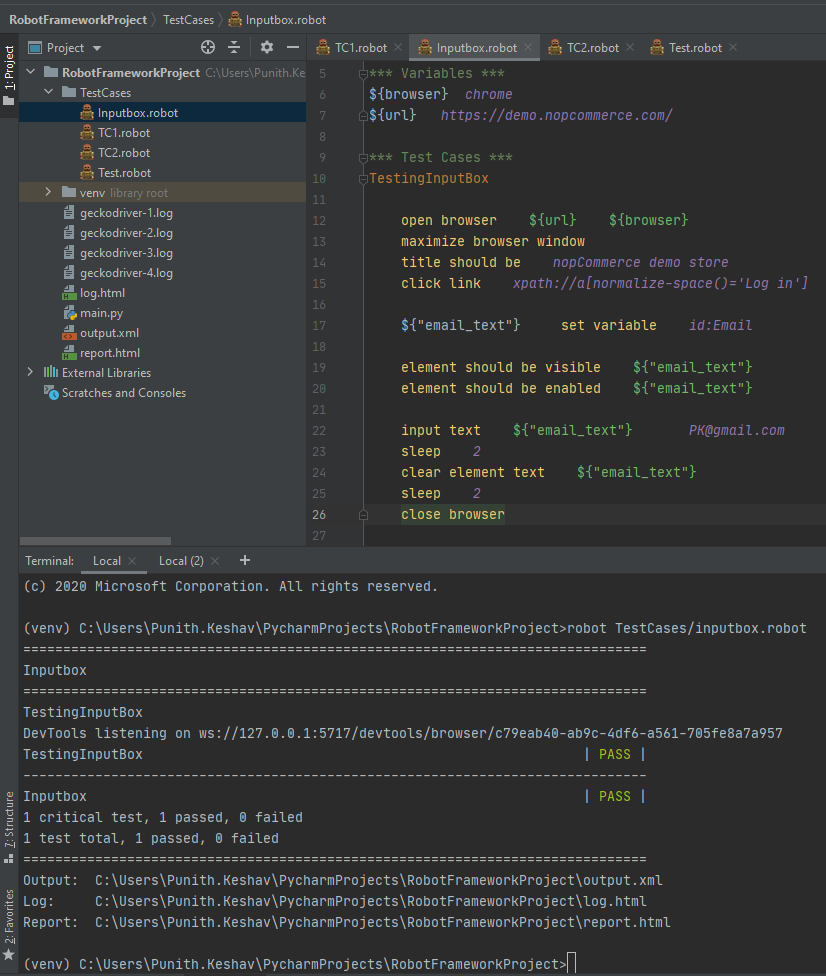
Robot Framework has inbuild keywords, other than that any user defined keywords required for the test cases will be scripted under this section



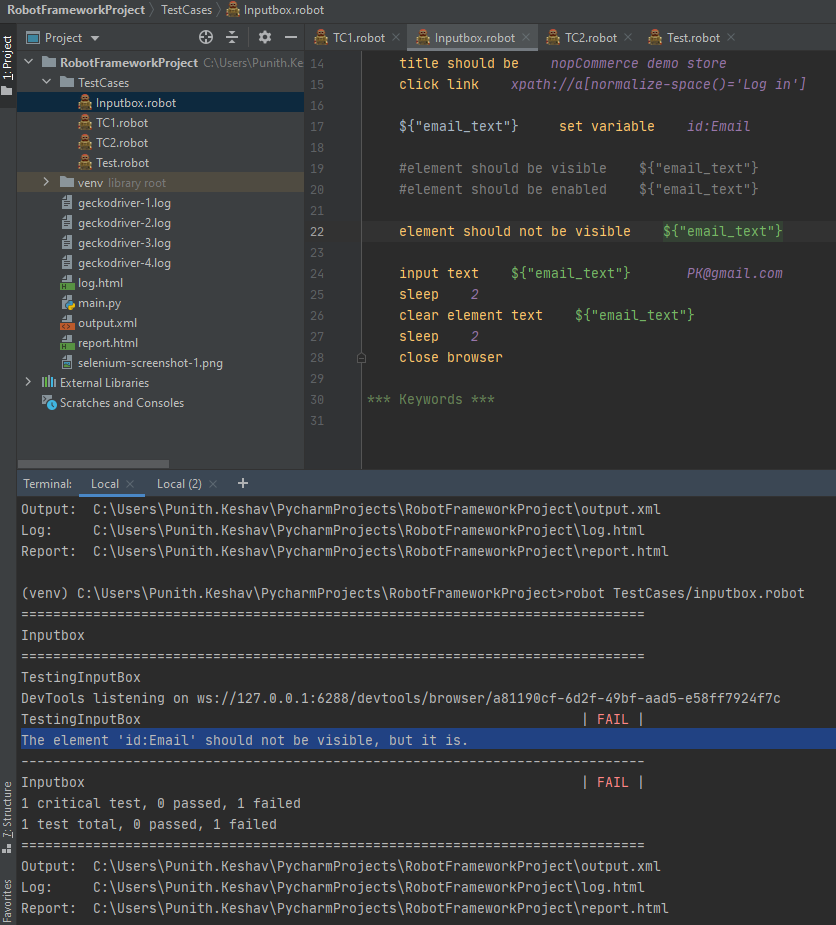
User Defined keywords are scripted here in this section and will be called in Test cases

**Handling INPUT BOX in Robot Framework:**

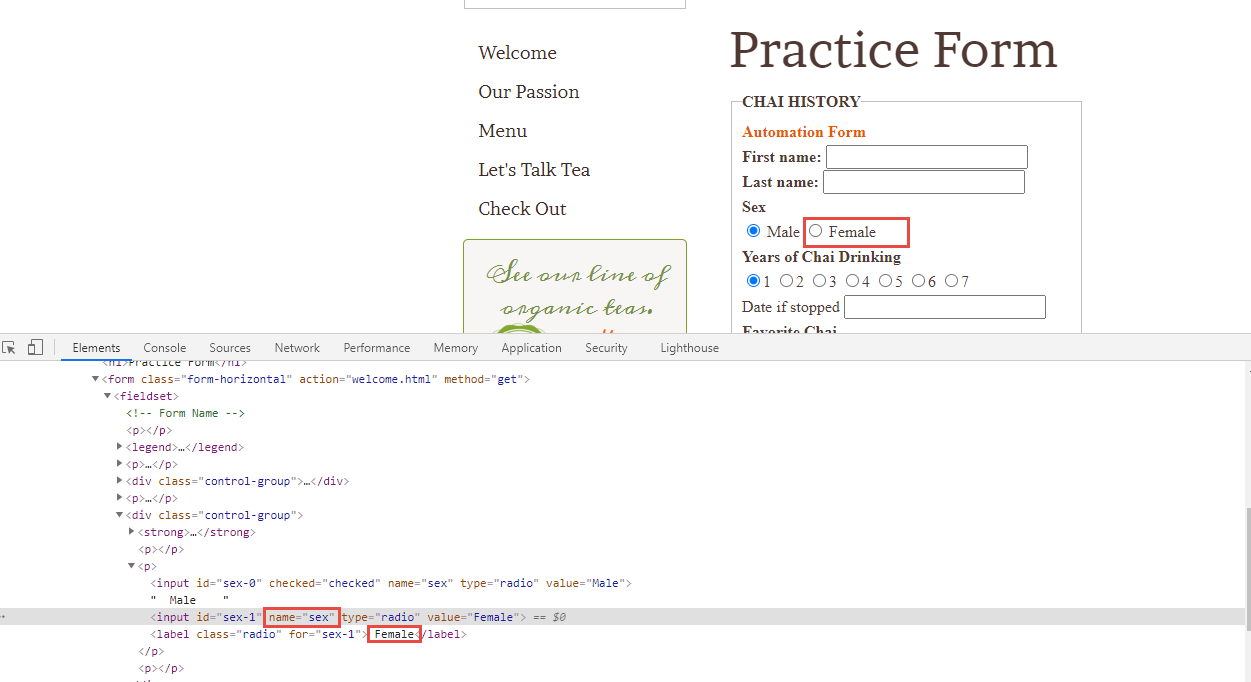


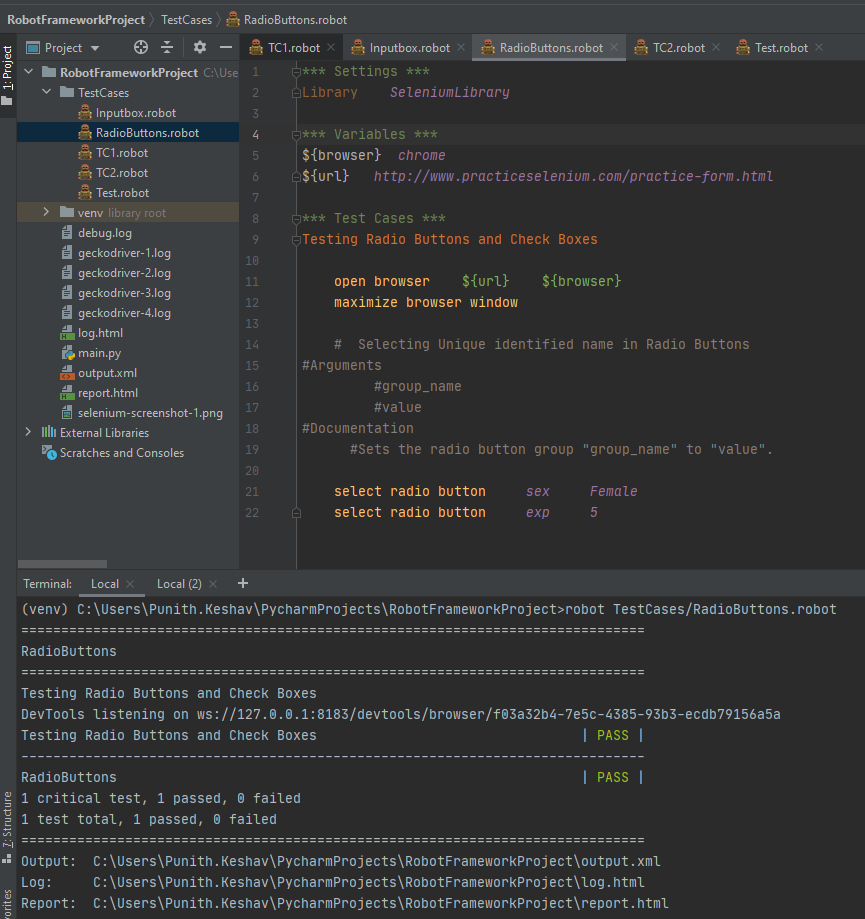


**Negative case:**



How To Select Radio Buttons & Check Boxes in Robot Framework | Selenium with Python





Set Selenium Speed:

## [Set Selenium Speed](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Set%20Selenium%20Speed)

#### Arguments

value

#### Documentation

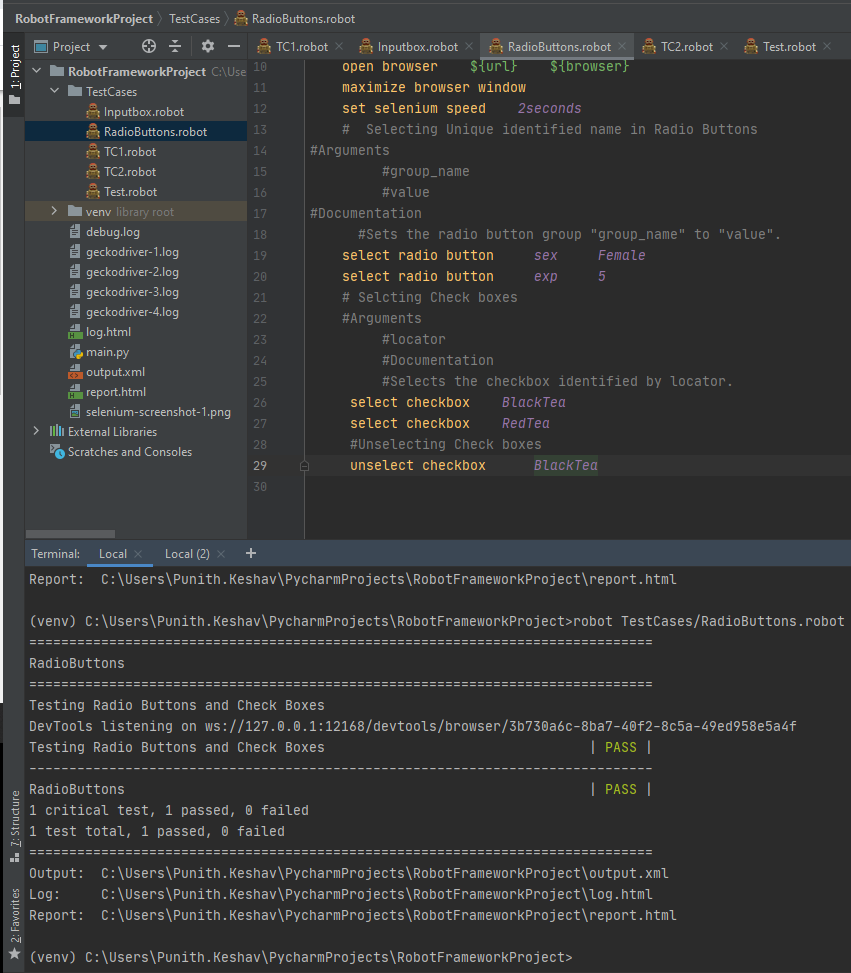
Sets the delay that is waited after each Selenium command.

The value can be given as a number that is considered to be seconds or as a human-readable string like 1 second. The previous value is returned and can be used to restore the original value later if needed.

See the [*Selenium Speed*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Selenium%20speed) section above for more information.

Example:

|  |
| --- |
| [*Set Selenium Speed*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Set%20Selenium%20Speed) |



# How To Select Options from Drop-Down & List Boxes in Robot Framework

## [Get Selected List Label](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Get%20Selected%20List%20Label)

#### Arguments

locator

#### Documentation

Returns the label of selected option from selection list locator.

If there are multiple selected options, the label of the first option is returned.

See the [*Locating elements*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Locating%20elements) section for details about the locator syntax.



select from list by label *continents Australia*

## [Select From List By Index](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Select%20From%20List%20By%20Index)

#### Arguments

locator  
\*indexes

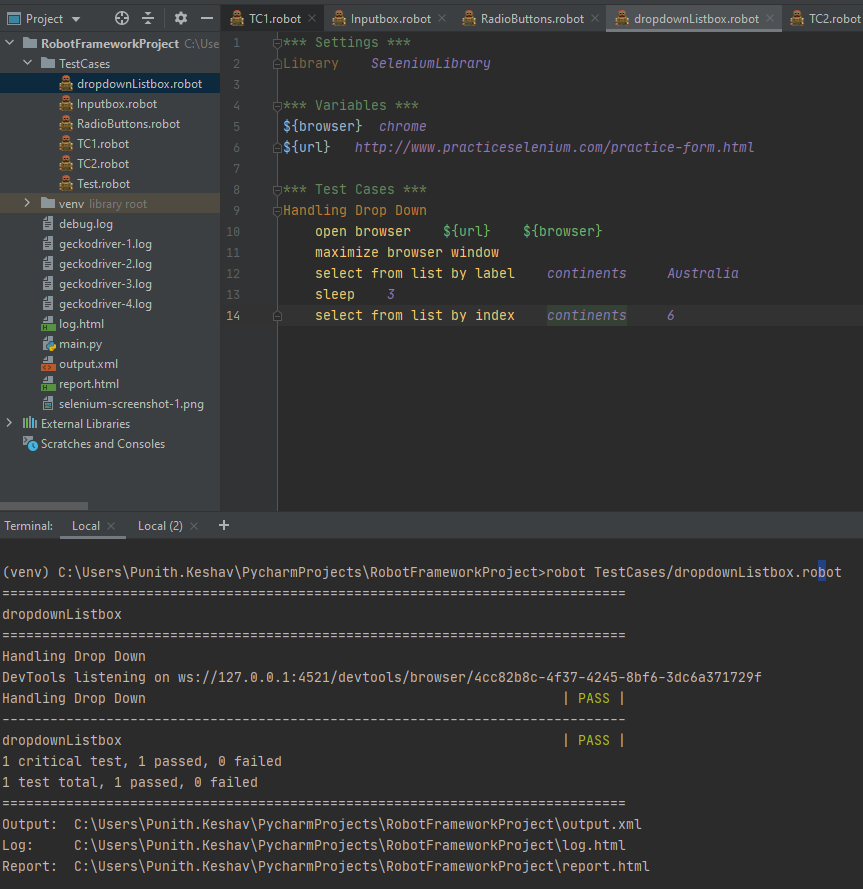
#### Documentation

Selects options from selection list locator by indexes.

Indexes of list options start from 0.

If more than one option is given for a single-selection list, the last value will be selected. With multi-selection lists all specified options are selected, but possible old selections are not cleared.

select from list by index *continents 6*



## [Select From List By Value](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Select%20From%20List%20By%20Value)

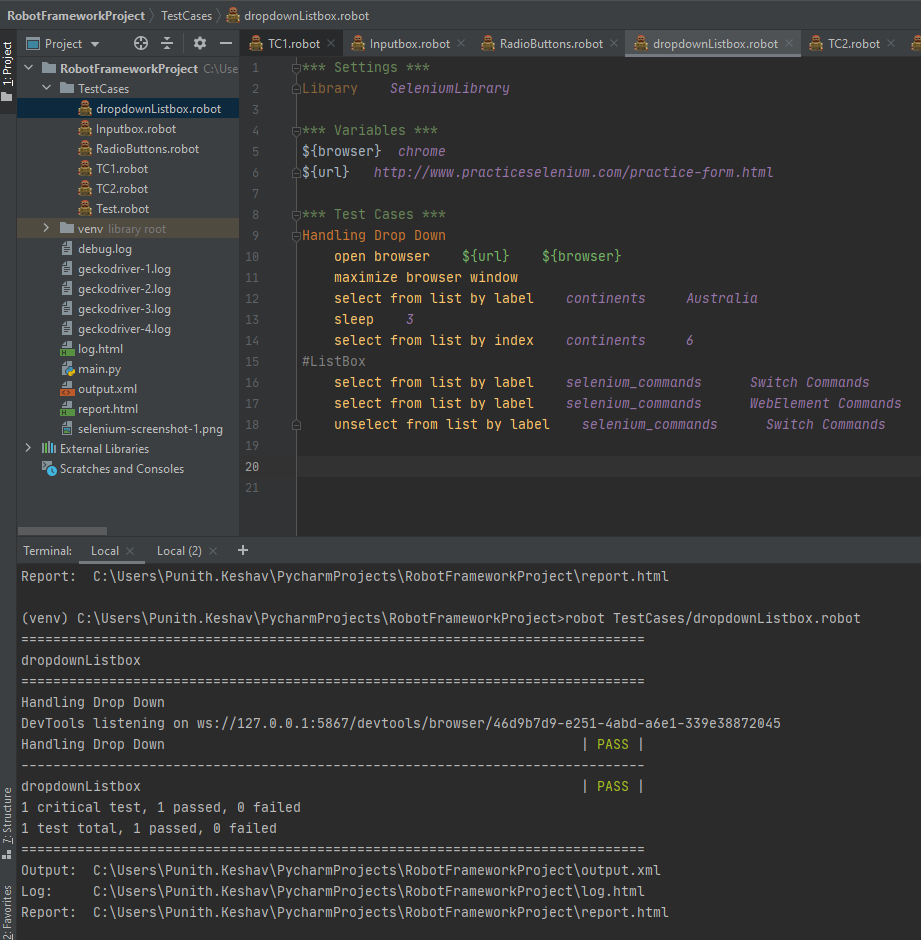
#### Arguments

locator  
\*values

#### Documentation

Selects options from selection list locator by values.

If more than one option is given for a single-selection list, the last value will be selected. With multi-selection lists all specified options are selected, but possible old selections are not cleared.



# Waits & TimeOuts in Robot Framework | Selenium Speed | Selenium Timeout | Implicit wait

# 

## [Select Radio Button](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Select%20Radio%20Button)

#### Arguments

group\_name  
value

#### Documentation

Sets the radio button group group\_name to value.

The radio button to be selected is located by two arguments:

* group\_name is the name of the radio button group.
* value is the id or value attribute of the actual radio button.

**Sleep :**

**This is used to give hard wait on particular step only**

sleep *2*

## [Set Selenium Speed](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Set%20Selenium%20Speed)

#### Arguments

value

#### Documentation

Sets the delay that is waited after each Selenium command.

The value can be given as a number that is considered to be seconds or as a human-readable string like 1 second. The previous value is returned and can be used to restore the original value later if needed.

See the [*Selenium Speed*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Selenium%20speed) section above for more information.

set selenium speed *3 seconds*

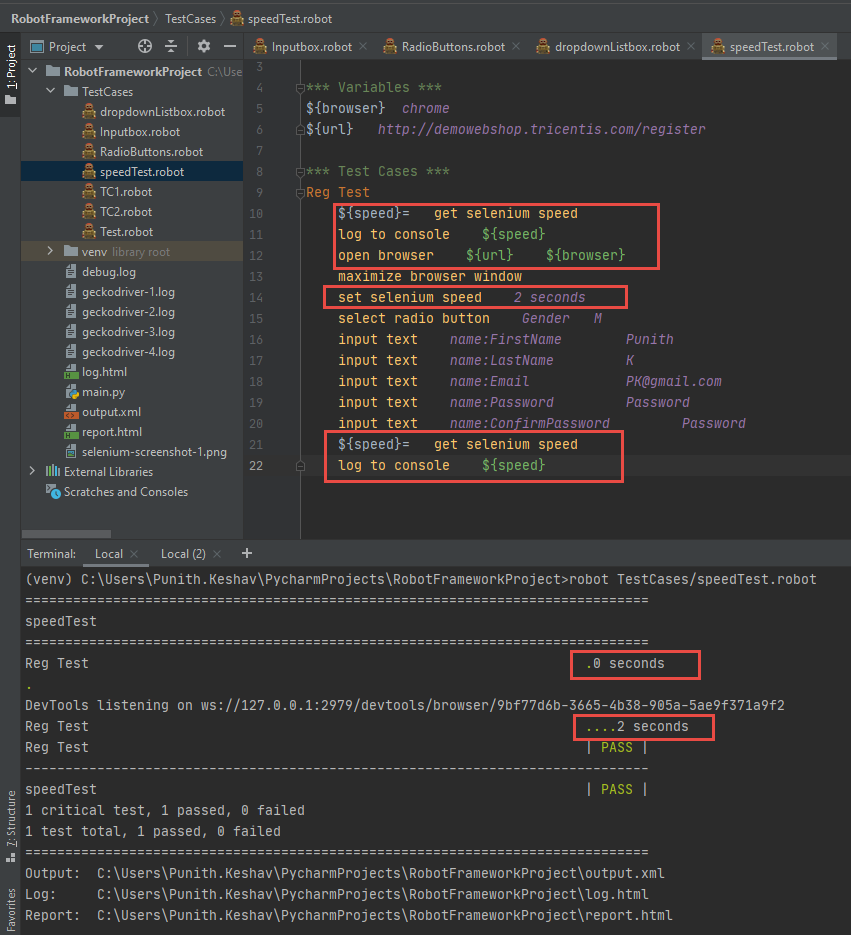
**This keyword gives 3 seconds delay for all the steps scripted under this keyword**

## [Get Selenium Speed](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Get%20Selenium%20Speed)

#### Documentation

Gets the delay that is waited after each Selenium command.

The value is returned as a human-readable string like 1 second.



## [Wait Until Page Contains](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Wait%20Until%20Page%20Contains)

#### Arguments

text  
timeout=None  
error=None

#### Documentation

Waits until text appears on the current page.

Fails if timeout expires before the text appears. See the *Timeouts* section for more information about using timeouts and their default value.error can be used to override the default error message.

wait until page contains *Register* # Default wait time is 5 seconds

## [Set Selenium Timeout](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Set%20Selenium%20Timeout)

#### Arguments

value

#### Documentation

Sets the timeout that is used by various keywords.

The value can be given as a number that is considered to be seconds or as a human-readable string like 1 second. The previous value is returned and can be used to restore the original value later if needed.

See the [*Timeout*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Timeout) section above for more information.

set selenium timeout *10 seconds*

## [Set Selenium Implicit Wait](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Set%20Selenium%20Implicit%20Wait)

#### Arguments

value

#### Documentation

Sets the implicit wait value used by Selenium.

The value can be given as a number that is considered to be seconds or as a human-readable string like 1 second. The previous value is returned and can be used to restore the original value later if needed.

This keyword sets the implicit wait for all opened browsers. Use [*Set Browser Implicit Wait*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Set%20Browser%20Implicit%20Wait) to set it only to the current browser.

# How to Close Single & Multiple Browsers in Robot Framework

## [Close Browser](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Close%20Browser)--Closes the current browser.

## [Close All Browsers](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Close%20All%20Browsers)

Closes all open browsers and resets the browser cache.

After this keyword, new indexes returned from [*Open Browser*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Open%20Browser) keyword are reset to 1.

This keyword should be used in test or suite teardown to make sure all browsers are closed.

# How to Handle Alerts & Frames Robot Framework | Selenium with Python

## [Handle Alert](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Handle%20Alert)

#### Arguments

action=ACCEPT  
timeout=None

#### Documentation

Handles the current alert and returns its message.

By default, the alert is accepted, but this can be controlled with the action argument that supports the following case-insensitive values:

* ACCEPT: Accept the alert i.e. press Ok. Default.
* DISMISS: Dismiss the alert i.e. press Cancel.
* LEAVE: Leave the alert open.

# Ex

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *http://testautomationpractice.blogspot.com/*\*\*\* Test Cases \*\*\*  
Alert  
 open browser ${url} ${browser}  
 maximize browser window  
 click element *xpath://button[normalize-space()='Click Me']* sleep *3* handle alert *accept*

## [Alert Should Be Present](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Alert%20Should%20Be%20Present)

#### Arguments

text=  
action=ACCEPT  
timeout=None

#### Documentation

Verifies that an alert is present and by default, accepts it.

Fails if no alert is present. If text is a non-empty string, then it is used to verify alert's message. The alert is accepted by default, but that behavior can be controlled by using the action argument same way as with [*Handle Alert*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Handle%20Alert).

timeout specifies how long to wait for the alert to appear. If it is not given, the global default [*timeout*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Timeout) is used instead.

action and timeout arguments are new in SeleniumLibrary 3.0. In earlier versions, the alert was always accepted and a timeout was hardcoded to one second.

## [Select Frame](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Select%20Frame)

#### Arguments

locator

#### Documentation

Sets frame identified by locator as the current frame.

See the [*Locating elements*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Locating%20elements) section for details about the locator syntax.

Works both with frames and iframes. Use [*Unselect Frame*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Unselect%20Frame) to cancel the frame selection and return to the main frame.

# How to Handle Tabbed Windows & Browser Windows | RobotFramework | Selenium with Python

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *http://demo.automationtesting.in/Windows.html*\*\*\* Test Cases \*\*\*  
Tabbed Windows  
 open browser ${url} ${browser}  
 maximize browser window  
 click element *xpath://a[@href='http://www.selenium.dev']//button[@class='btn btn-info'][normalize-space()='click']* select window *title=SeleniumHQ Browser Automation* click element *xpath://a[contains(text(),'Support')]* sleep *3* close all browsers

Multiple Windows:

## [Switch Browser](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Switch%20Browser)

#### Arguments

index\_or\_alias

#### Documentation

Switches between active browsers using index\_or\_alias.

Indices are returned by the [*Open Browser*](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Open%20Browser) keyword and aliases can be given to it explicitly. Indices start from 1.

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *https://www.google.com/*${url1} *https://www.bing.com/*\*\*\* Test Cases \*\*\*  
Multiple Windows  
 open browser ${url} ${browser}  
 maximize browser window  
 sleep *3* open browser ${url1} ${browser}  
 maximize browser window  
 sleep *3* switch browser *1* ${title}= get title  
 log to console ${title}  
  
 switch browser *2* ${title2}= get title  
 log to console ${title2}  
  
 sleep *3* close all browsers

# Browser related Keywords | Robot Framework | Go To | Go Back | Get Location

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *https://www.google.com/*${url1} *https://www.bing.com/*\*\*\* Test Cases \*\*\*  
Navigation  
 open browser ${url} ${browser}  
 maximize browser window  
 ${loc}= get location  
 log to console ${loc}  
  
 go to ${url1}  
 ${loc}= get location  
 log to console ${loc}  
  
 go back  
 ${loc}= get location  
 log to console ${loc}

# How to Capture Element & Full Page Screenshot | RobotFramework | Selenium with Python



# How to perform Mouse Operations in Robot Framework

Right Click

## [Open Context Menu](https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html#Open%20Context%20Menu)

#### Arguments

locator

#### Documentation

Opens the context menu on the element identified by locator.

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *https://swisnl.github.io/jQuery-contextMenu/demo.html*${url2} *http://testautomationpractice.blogspot.com/*${url3} *http://www.dhtmlgoodies.com/scripts/drag-drop-custom/demo-drag-drop-3.html*\*\*\* Test Cases \*\*\*  
MouseActions  
 open browser ${url} ${browser}  
 maximize browser window  
 # Right Click action  
 open context menu *xpath://span[@class='context-menu-one btn btn-neutral']* sleep *3* #Double Click action  
 go to ${url2}  
 maximize browser window  
 double click element *xpath://button[normalize-space()='Copy Text']* sleep *3* #Drag and Drop  
 go to ${url3}  
 maximize browser window  
 drag and drop *id:box6 id:box106* sleep *4* close all browsers

# User Defined Keywords & Resource Files in Robot Framework



User Defined Keyword without Arguments:

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *http://demo.guru99.com/test/newtours/index.php*\*\*\* Test Cases \*\*\*  
TC1  
 BrowserLogin  
 input text *name:userName Mercury* input text *name:password Mercury*\*\*\* Keywords \*\*\*  
BrowserLogin  
 open browser ${url} ${browser}  
 maximize browser window

User Defined Keyword with Arguments:

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *http://demo.guru99.com/test/newtours/index.php*\*\*\* Test Cases \*\*\*  
TC1  
 LaunchBrowser ${url} ${browser}  
 input text *name:userName Mercury* input text *name:password Mercury*\*\*\* Keywords \*\*\*  
LaunchBrowser  
 ***[Arguments]*** ${appurl} ${appbrowser}  
 open browser ${appurl} ${appbrowser}  
 maximize browser window

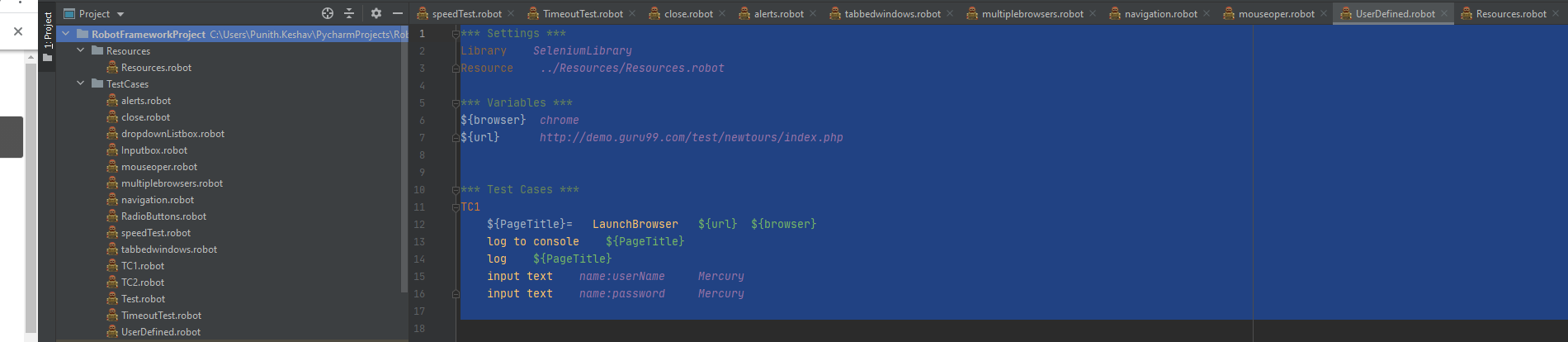
User Defined Keyword with Arguments and return Value:

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *http://demo.guru99.com/test/newtours/index.php*\*\*\* Test Cases \*\*\*  
TC1  
 ${PageTitle}= LaunchBrowser ${url} ${browser}  
 log to console ${PageTitle}  
 log ${PageTitle}  
 input text *name:userName Mercury* input text *name:password Mercury*\*\*\* Keywords \*\*\*  
LaunchBrowser  
 ***[Arguments]*** ${appurl} ${appbrowser}  
 open browser ${appurl} ${appbrowser}  
 maximize browser window  
 ${title}= get title  
 ***[Return]*** ${title}

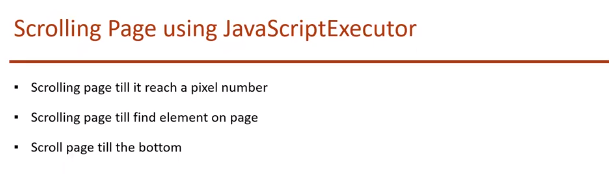
Resources File:

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Keywords \*\*\*  
LaunchBrowser  
 ***[Arguments]*** ${appurl} ${appbrowser}  
 open browser ${appurl} ${appbrowser}  
 maximize browser window  
 ${title}= get title  
 ***[Return]*** ${title}

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*Resource *../Resources/Resources.robot*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *http://demo.guru99.com/test/newtours/index.php*\*\*\* Test Cases \*\*\*  
TC1  
 ${PageTitle}= LaunchBrowser ${url} ${browser}  
 log to console ${PageTitle}  
 log ${PageTitle}  
 input text *name:userName Mercury* input text *name:password Mercury*



# Scrolling Page using JavaScript executor in Robot Framework



\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *https://www.countries-ofthe-world.com/flags-of-the-world.html*\*\*\* Test Cases \*\*\*  
Scroll  
#Scroll page till it reach a pixel number  
 open browser ${url} ${browser}  
 maximize browser window  
 execute javascript *window.scrollTo(0,1500)*#Scroll page till find element on the page  
 scroll element into view *xpath://img[@alt='Flag of India']*#Scroll page till bottom  
 execute javascript *window.scrollTo(0,document.body.scrollHeight)* sleep *5*#Scroll page back to top  
 execute javascript *window.scrollTo(0,-document.body.scrollHeight)*

# How to work with FOR loop in Robot Framework

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Test Cases \*\*\*  
ForLoop1  
 # FOR ${i} IN RANGE 1 4  
 # click element xpath:(//div[@class='tabpanel\_tab\_content']/div)[6]  
 #END  
 FOR ${i} IN RANGE *1 2* LOG TO CONSOLE ${i}  
 END  
ForLoop2 # with single spacing  
 FOR ${i} IN *1 2* LOG TO CONSOLE ${i}  
 END  
ForLoop3 # with 2 space spacing  
 FOR ${i} IN *1 2* LOG TO CONSOLE ${i}  
 END

# 

# How to Count & Extract Link Texts in Robot Framework

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *http://demo.guru99.com/test/newtours/index.php*\*\*\* Test Cases \*\*\*  
Extract Links  
 open browser ${url} ${browser}  
 maximize browser window  
 ${AllLinksCount}= get element count *xpath://a* log to console ${AllLinksCount}  
  
 @{LinkItems} create list  
 FOR ${i} IN RANGE *1* ${AllLinksCount}  
 ${linktext}= get text *xpath:(//a)[*${i}*]* LOG TO CONSOLE ${linktext}  
 END

# How to Handle Web/HTML Table in Robot Framework

# 

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${url} *http://testautomationpractice.blogspot.com/*\*\*\* Test Cases \*\*\*  
Table Validations  
 open browser ${url} ${browser}  
 maximize browser window  
 #Gives rows and colums count  
 ${rows}= get element count *xpath://table[@name='BookTable']/tbody/tr* ${columns}= get element count *xpath://table[@name='BookTable']/tbody/tr[1]/th* log to console ${rows}  
 log to console ${columns}  
 #Data from Table  
 ${data}= get text *xpath://table[@name='BookTable']/tbody/tr[5]/td[1]* log to console ${data}  
 #Table Column Data Validations  
 table column should contain *xpath://table[@name='BookTable'] 2 Author* table header should contain *xpath://table[@name='BookTable'] Author* #Table Row Data Validations  
 table row should contain *xpath://table[@name='BookTable'] 4 Learn JS* # Table cell validation  
 table cell should contain *xpath://table[@name='BookTable'] 2 Mukesh*

Robotmetrics:

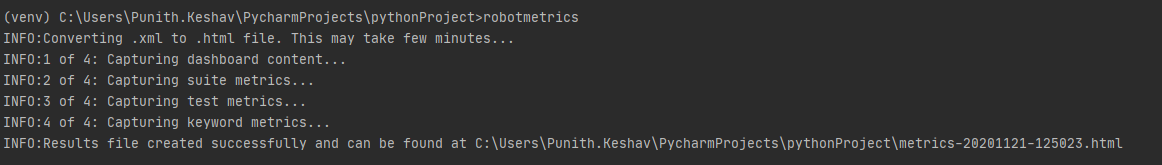
<https://github.com/adiralashiva8/robotframework-metrics>

Install plug in below

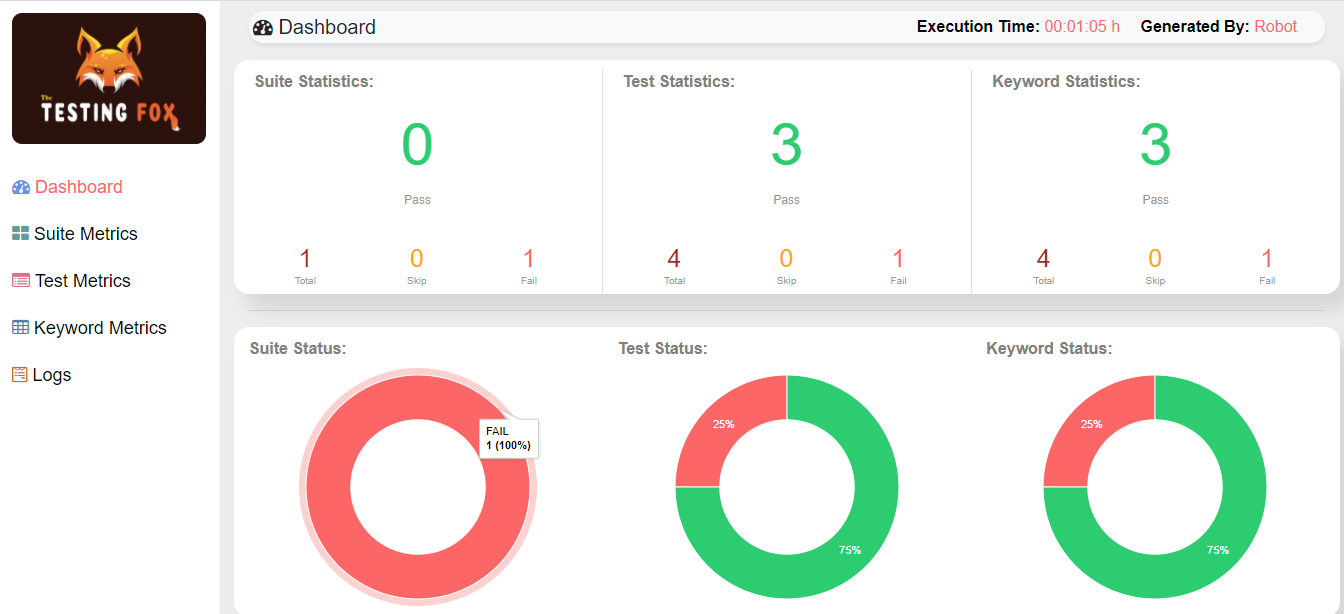
pip install robotframework-metrics

Execute Particular Test

After execution on command line run : robotmetrics ( to generate test metrices)



**Test Metrices will be displayed as below**



# Data Driven Testing Using Script in Robot Framework



Data in Script:

Resource file :

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*\*\*\* Variables \*\*\*  
${browser} *chrome*${Login url} *https://admin-demo.nopcommerce.com/*\*\*\* Keywords \*\*\*  
Open My Browser  
 open browser ${Login url} ${browser}  
 maximize browser window  
  
close Browsers  
 close all browsers  
  
Open Login Page  
 go to ${Login url}  
  
Input UserName  
 ***[Arguments]*** ${username}  
 input text *id:Email* ${username}  
  
Input Password  
 ***[Arguments]*** ${password}  
 input text *id:Password* ${password}  
  
Click on Login Button  
 click element *xpath://input[@value='Log in']*Click on Logout Link  
 click link *Logout*Error Message should be visible  
 page should contain *Login was unsuccessful*Dashboard Page should be visible  
 page should contain *Dashboard*

Test File :

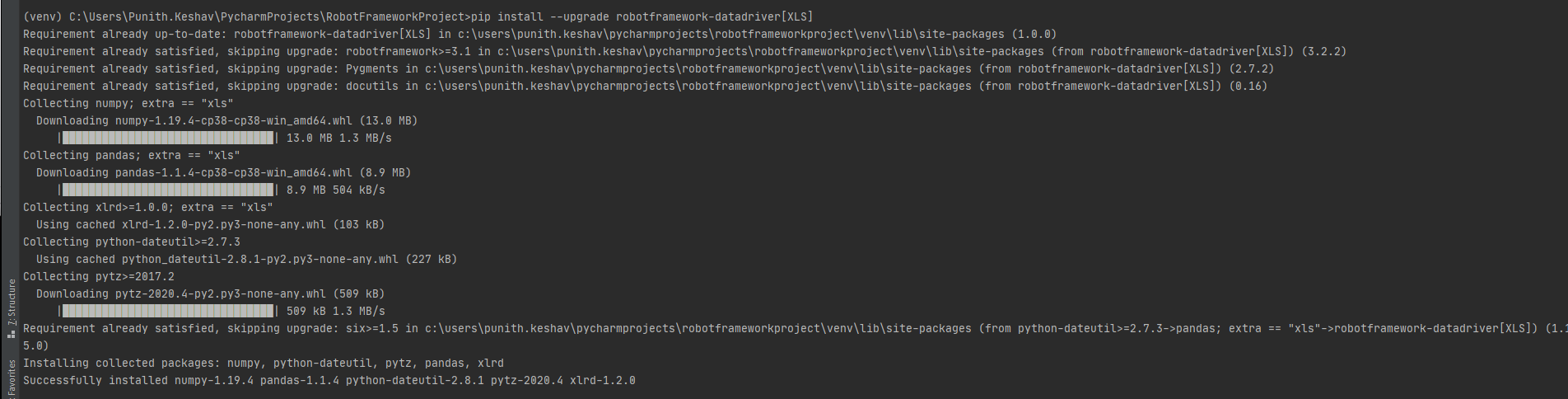
\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*Resource *../Resources/login\_Resources.robot*Suite Setup Open My Browser  
Suite Teardown close Browsers  
Test Template Invalid Login  
  
\*\*\* Test Cases \*\*\* username password  
Right User empty Password admin@yourstore.com ${EMPTY}  
Right User wrong Password admin@yourstore.com xyz  
Wrong User right Password admin1@yourstore.com admin  
Wrong User empty Password admin1@yourstore.com ${EMPTY}  
Wrong User wrong Password admin1@yourstore.com xyz  
  
  
\*\*\* Keywords \*\*\*  
Invalid Login  
 ***[Arguments]*** ${username} ${password}  
 Input UserName ${username}  
 Input Password ${password}  
 Click on Login Button  
 Error Message should be visible

Using excel file:

Install below plugin

pip install robotframework-datadriver

pip install --upgrade robotframework-datadriver[XLS]



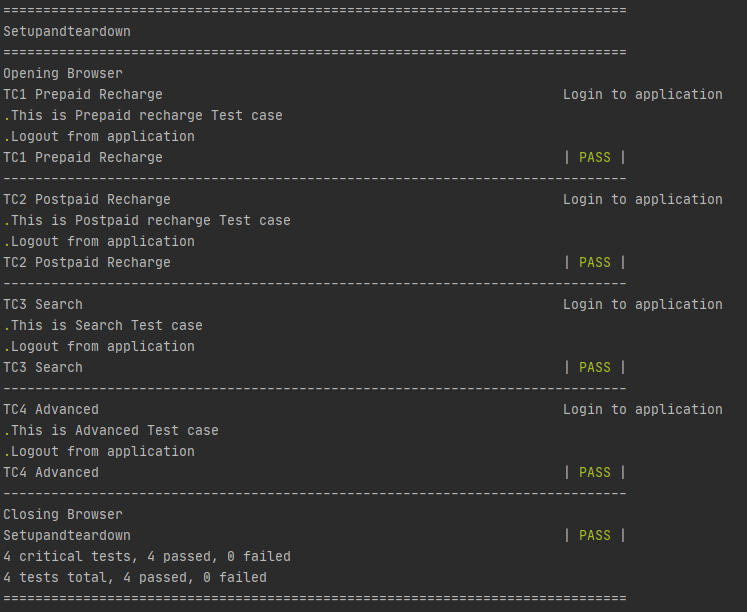
**CSV and Excel has to be relooked:**

# Setup & TearDown in Robot FrameworkRobot

# 

During Test execution we need to run some test cases before and some test case after .Hence Test Set up and Test tear down is used

\*\*\* Settings \*\*\*  
Suite Setup log to console *Opening Browser*Suite Teardown log to console *Closing Browser*Test Setup log to console *Login to application*Test Teardown log to console *Logout from application*\*\*\* Test Cases \*\*\*  
TC1 Prepaid Recharge  
 log to console *This is Prepaid recharge Test case*TC2 Postpaid Recharge  
 log to console *This is Postpaid recharge Test case*TC3 Search  
 log to console *This is Search Test case*TC4 Advanced  
 log to console *This is Advanced Test case*



# Tags | Grouping Tests in Robot FrameworkRobot

\*\*\* Settings \*\*\*  
  
  
\*\*\* Test Cases \*\*\*  
TC1 User Registertration Test  
 ***[Tags]*** *sanity* log to console *This is User Registertration Test* log to console *User Registertration Test is over*TC2 Login Test  
 ***[Tags]*** *Regression* log to console *This is Login Test* log to console *User Login Test is over*TC3 Change User Settings Test  
 ***[Tags]*** *Regression* log to console *This is Change User Settings Test* log to console *User Change User Settings Test is over*TC3 Logout Test  
 ***[Tags]*** *sanity* log to console *This is Logout Test* log to console *User Logout Test is over*

To Run only Sanity Test Cases

**robot --include=sanity TestCases/tags.robot**

To Run only Regression Test Cases

**robot --include=Regression TestCases/tags.robot**

To include Multiple Tags in single Test run:

**robot -i sanity -i Regression TestCases/Tags.robot**

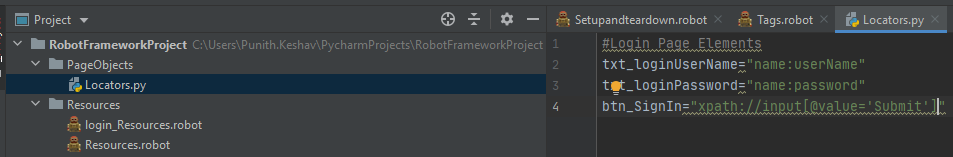
To exclude Regression Tags in single Test run:

**robot -e Regression TestCases/Tags.robot**

# Page Object Model (POM) Pattern in Robot Framework

# 

Create **PageObjects** Folder and create **Locator**.**py** file to capture all the locators



In Resources Folder Create **LoginKeyword.robot** to capture user defined keywords

Need to specify Selenium Library and also have t specify the locator reference elements from Locator.py file created

\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*Variables *../PageObjects/Locators.py*

Keywords are created as below

\*\*\* Keywords \*\*\*  
Open My Browser  
 ***[Arguments]*** ${SiteUrl} ${Browser}  
 open browser ${SiteUrl} ${Browser}  
 maximize browser window  
Enter User Name  
 ***[Arguments]*** ${username}  
 input text ${txt\_loginUserName} ${username}  
Enter Password  
 ***[Arguments]*** ${password}  
 input text ${txt\_loginPassword} ${password}  
Click SignIn  
 click button ${btn\_SignIn}  
Veriify Login  
 title should be *Login: Mercury Tours*Close my Browsers  
 close all browsers

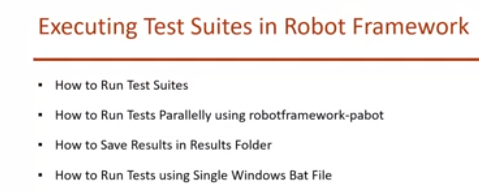
Create a Test Case now : LoginTest.robot

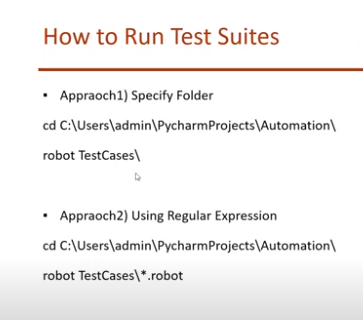
\*\*\* Settings \*\*\*  
Library *SeleniumLibrary*Resource *../Resources/LoginKeyword.robot*\*\*\* Variables \*\*\*  
${Browser} *chrome*${SiteUrl} *http://demo.guru99.com/test/newtours/index.php*${user} *tutorial*${password} *tutorial*\*\*\* Test Cases \*\*\*  
Login Test  
 Open My Browser ${SiteUrl} ${Browser}  
 Enter User Name ${user}  
 Enter Password ${password}  
 Click SignIn  
 sleep *3* Verify Successfull Login  
 Close my Browsers

**Adding another Test Case:**



# Parallel Test Execution in Robot Framework





If we have to run all the Test cases in the folder

**robot Foldername 🡪 This will run the tests under the folder one by one**

**Parallel Test Execution:**

**Install Pabot Plugin :** [**https://pypi.org/project/robotframework-pabot/**](https://pypi.org/project/robotframework-pabot/)

pip install robotframework-pabot

In Pycharm setting install Pabot for the project

Open Cmd prompt and run as administrator to install pabot

Pip install -U robotframework-pabot

To run Tests: Run two Test cases Paralelly

pabot --processes 2 TestCases\\*.robot

Save is particular directory :

pabot --processes 2 --outputdir Results Testcases\\*.robot

**Run Using Batch file :**

We can execute batch file either on pycharm or from any location in the system even without having pycarm

Project 🡪 Create New File 🡪 run.bat

cd C:\Users\Punith.Keshav\PycharmProjects\RobotFrameworkProject  
robot TestCases/LoginTest.robot

# Headless Browser Testing in Robot Framework

# Headlesschrome

${Browser} *headlesschrome*

# Headlessfirefox

# Jenkins Integration with Robot Framework