ROBOT FRAMEWORK USING PYTHON IN VS CODE

What Is the Robot Framework?

Robot Framework is a Test Automation tool in which the test cases are written using keywords that makes it easy to learn and use.

These keywords are written in a tabular form. With Robot Framework, the Test Scripts are replaced by a few keywords thereby replacing the need for large pieces of code.

Let us understand the keyword-driven approach of this Framework with a simple example.

Example: Suppose, I want to test a website say Google.com, for which the very first step would be to open a Browser and open the 'Google.com' webpage. Now to automate this step using Robot Framework, we have a keyword called "Open Browser".

The script for this step would look as shown below:

Open Browser	Google.com	Chrome

Test data sections

Different sections in the data

Section	Used for
Settings	1) Importing test libraries, resource files, and variable files. 2) Defining metadata for test suites and test cases.
Variables	Defining variables that can be used elsewhere in the test data.
Test Cases	Creating test cases from available keywords.
Tasks	Creating tasks using available keywords. A single file can only contain either tests or tasks.

Keywords	Creating user keywords from existing lower-level keywords
Comments	Additional comments or data. Ignored by Robot Framework.

How to Install Python on Windows

Installing and using Python on Windows is very simple. The installation procedure involves just three steps:

1. Download the binaries
2. Run the Executable installer
3. Add Python to PATH environmental variables

It is recommended to install the latest version of Python

Step 1: Download the Python Installer binaries

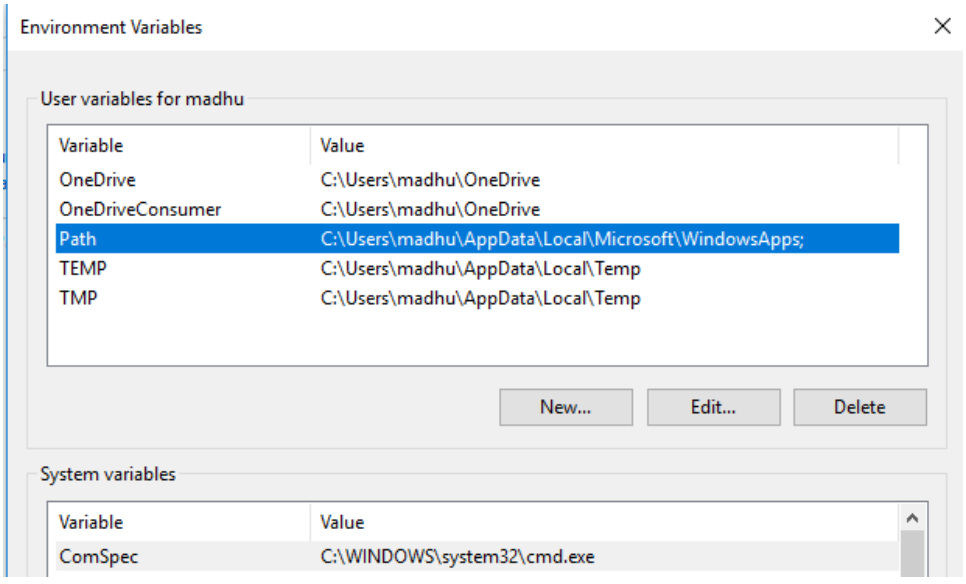
- Open the official Python website in your web browser. Navigate to the Downloads tab for Windows.
- Choose the latest Python 3 release. In our example, we choose the latest Python 3.7.3 version.
- Click on the link to download **Windows x86 executable installer** if you are using a 32-bit installer. In case your Windows installation is a 64-bit system, then download **Windows x86-64 executable installer**.

Step 2: Run the Executable Installer

- Once the installer is downloaded, run the Python installer.
- Check the **Install launcher for all users** check box. Further, you may check the **Add Python 3.7 to path** check box to include the interpreter in the execution path.
- Check the Add Python 3.7 to PATH.
- Click on the Install Now

Step 3: Add Python to environmental variables

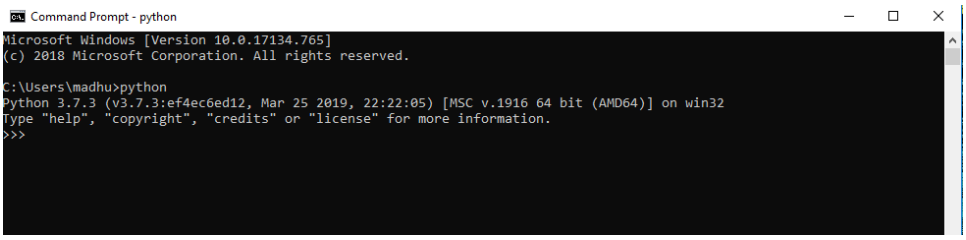
The last (optional) step in the installation process is to add Python Path to the System Environment variables. Because Add Python 3.7 to PATH will automatically set your path in your environment variable.



- Click on Edit>New>Paste
- Good to Go

Step 4: Verify the Python Installation

You have now successfully installed Python 3.7.3 on Windows 10. You can verify if the Python installation is successful either through the command line or through the IDLE app that gets installed along with the installation. Search for the command prompt and type “python”. You can see that Python 3.7.3 is successfully installed.

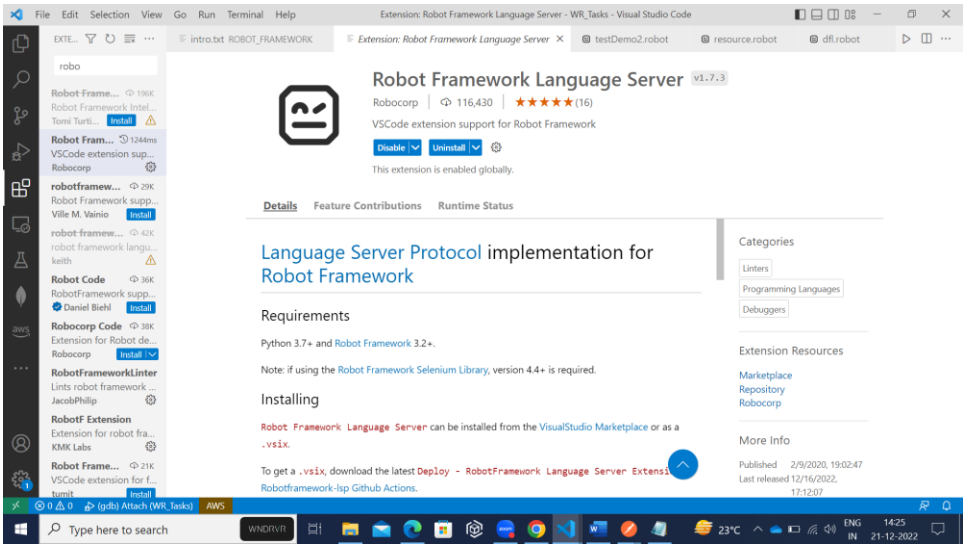


How to Install VS Code

- Download the Visual Studio Code installer for Windows.
- Once it is downloaded, run the installer (VSCodeUserSetup-{version}.exe). This will only take a minute.
- By default, VS Code is installed under `C:\Users\{Username}\AppData\Local\Programs\Microsoft VS Code`.

Link to download VS Code https://code.visualstudio.com/download

Extension Id to download Robot framework in VS Code robocorp.robotframework-lsp



How to Install Robot-Framework in Python

Installation

The recommended approach to install the robot framework on python is to use the pip. You can use the following command to install the framework.

1	**Installation of Robot Framework:** --> pip install robotframework

2	**Install the request Library using the following command:** --> pip install requests **RESTinstances:** --> pip install --upgrade RESTinstance **Install the following Libraries using CMD for API testing:** --> pip install robotframework --> pip install requests --> pip install robotframework-requests --> pip install robotframework-jsonlibrary **Verifying Installation:** --> robot --version --> rebot --version **Checking proper installation of robot framework:** --> pip show robotframework **Selenium Library** → `pip install robotframework-selenium2library==1.8.0` --> pip install --upgrade robotframework-seleniumlibrary --> pip install --upgrade robotframework-selenium2library

Verifying Installation

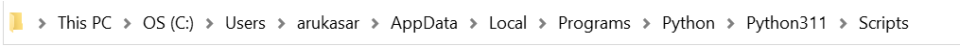
After the successful installation, you should be able to see both interpreter and robot framework versions using the –version option.

```
1 robot --version
2 rebot --version
```

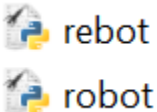
To verify if the robot framework is installed:

For mac users- To verify if the robot is installed correctly in

Go to: - Where your python is installed, then go to the script folder



And check if these two files are present in this path (In the Scripts Folder)

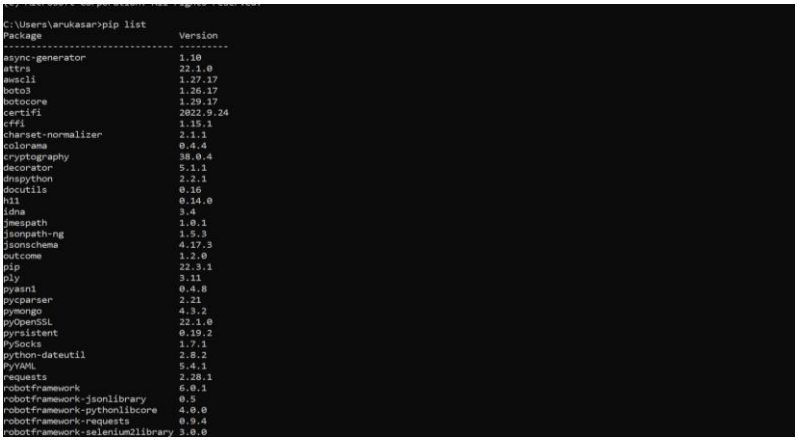


if yes then copy the way - for an exam (C:\Users\arukasar\AppData\Local\Programs\Python\Python311\Scripts)

and paste it into your environment variable (user variable section):-

Show the list of installed packages using

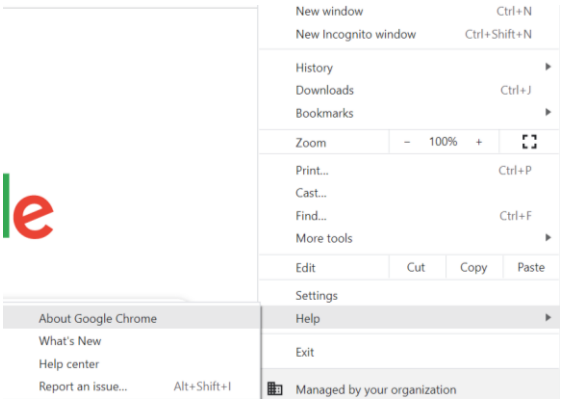
1	pip list



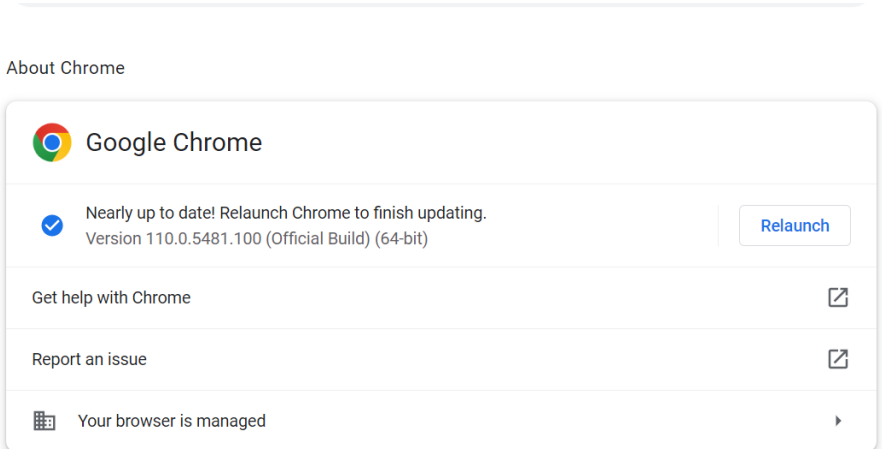
```
C:\Users\arukasar>pip list
Package                       Version
----------------------------- ---------
async-generator               1.10
attrs                         22.1.0
awscli                        1.27.17
boto3                         1.26.17
botocore                      1.29.17
certifi                       2022.9.24
cffi                          1.15.1
charset-normalizer            2.1.1
colorama                      0.4.4
cryptography                  38.0.4
decorator                     5.1.1
dnspython                     2.2.1
docutils                      0.16
h11                           0.14.0
idna                          3.4
jmespath                      1.0.1
jsonpath-ng                   1.5.3
jsonschema                    4.17.3
outcome                       1.2.0
pip                           22.3.1
ply                           3.11
pyasn1                        0.4.8
pycparser                     2.21
pymongo                       4.3.2
pyOpenSSL                     22.1.0
pyrsistent                    0.19.2
PySocks                       1.7.1
python-dateutil               2.8.2
PyYAML                        5.4.1
requests                      2.28.1
robotframework                6.0.1
robotframework-jsonlibrary    0.5
robotframework-pythonlibcore  4.0.0
robotframework-requests       0.9.4
robotframework-selenium2library 3.0.0
```

Chrome Driver Installation for Automating the Test Cases

Go to Three dots > Help > About Google Chrome



Check your chrome version (if not updated, please update it first)

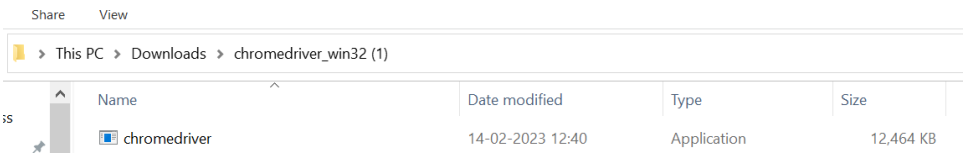


Go to download Chrome driver Link: https://chromedriver.chromium.org/downloads

Install According to your chrome version

- If you are using Chrome version 110, please download ChromeDriver 110.0.5481.77

Go to the location where you have installed chrome drive > do extract all > then go inside that folder just copy the chrome driver from there



and paste it into a Scripts folder where rebot and robot are there.

Now you are good to go to write and run any test cases.

Writing a first automation case

Following example shows how to write your first robot automation case:

- Create a new file and save it using an extension. robot, for example, Hello.robot
- Write in file `*** Test Cases ***`. This creates a section for test cases.

```
*** Test Cases ***

TestCase1
```

```
Log    Hello World
Log    Hello World1
Log    Hello World2
```

Command to run – robot filename.robot

GOOD TO GO