

Environment Building

pymycobot is a Python package used for serial communication with myCobot. It supports Python2, Python3.5 and later versions.

Before using pymycobot, make sure to build a Python environment. Follow the steps below to install Python.

1 Download and Installation of Python

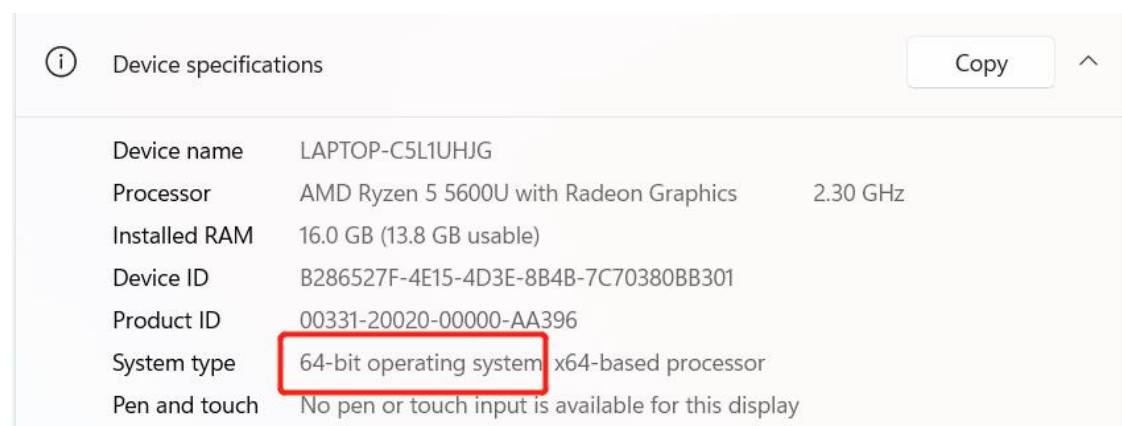
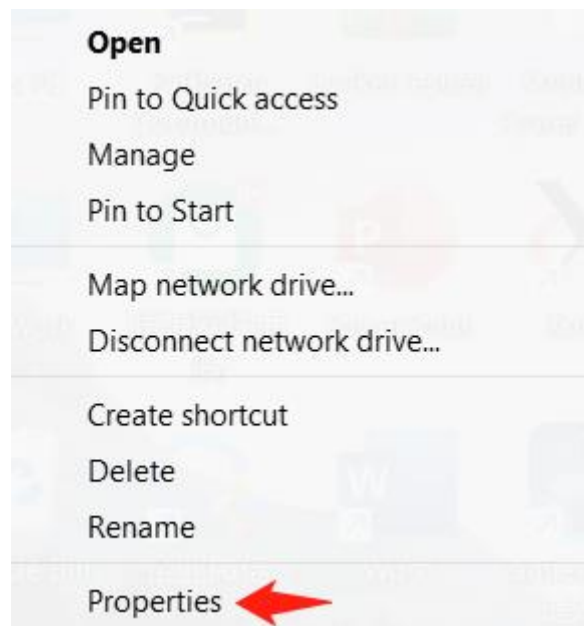
Python is applicable to:

- myCobot 320 M5
- myCobot 320 PI

At present, Python has two versions: 2.x and 3.x. These two versions are incompatible with each other. This section takes the version 3.x as an example due to its increasing popularity.

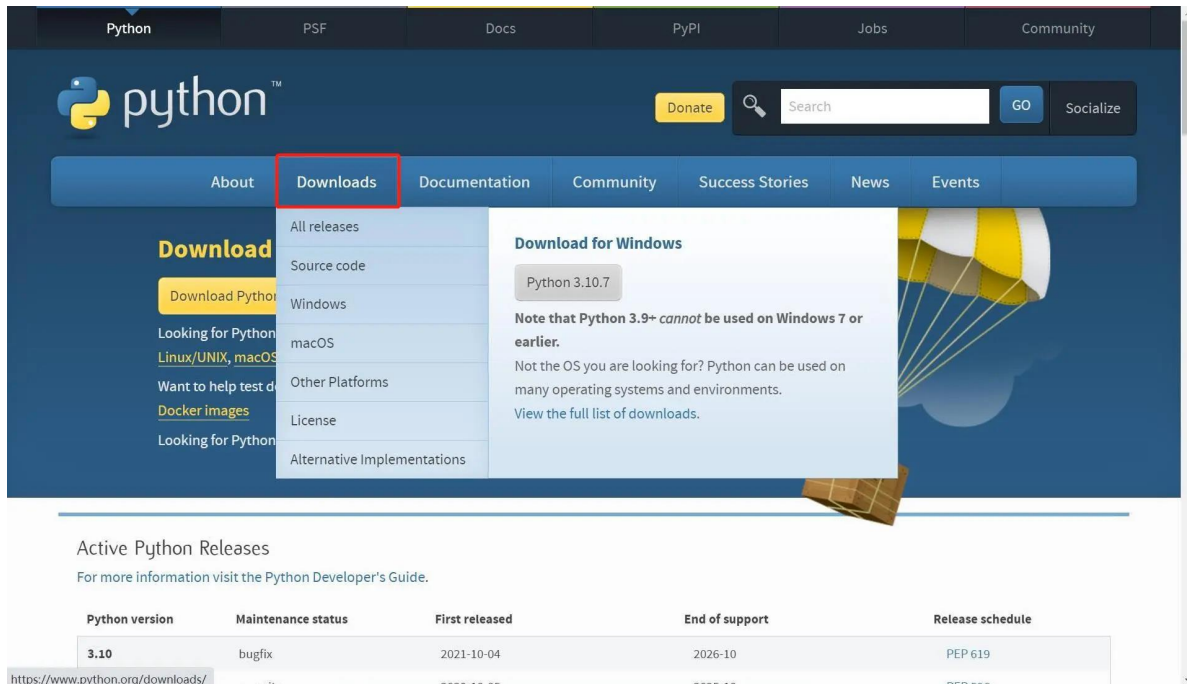
1.1 Installing Python

Notice: Before installation, check the operation system of PC. Press right button on the **My Computer** icon and then select **Properties**. Install the corresponding Python.



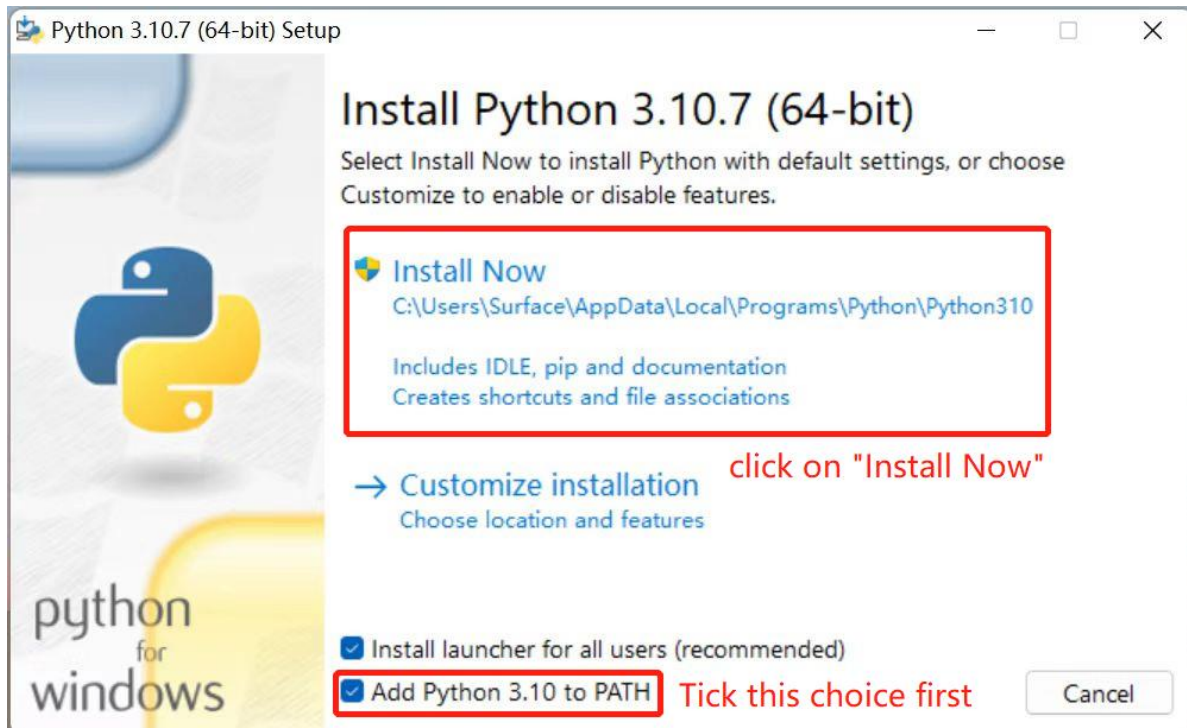
- Go to <http://www.python.org/download/> to download Python.

- Click on **Downloads**, and then download begins. Tick **Add Python 3.10 to PATH**. Click on **Install Now**, and then installation begins.

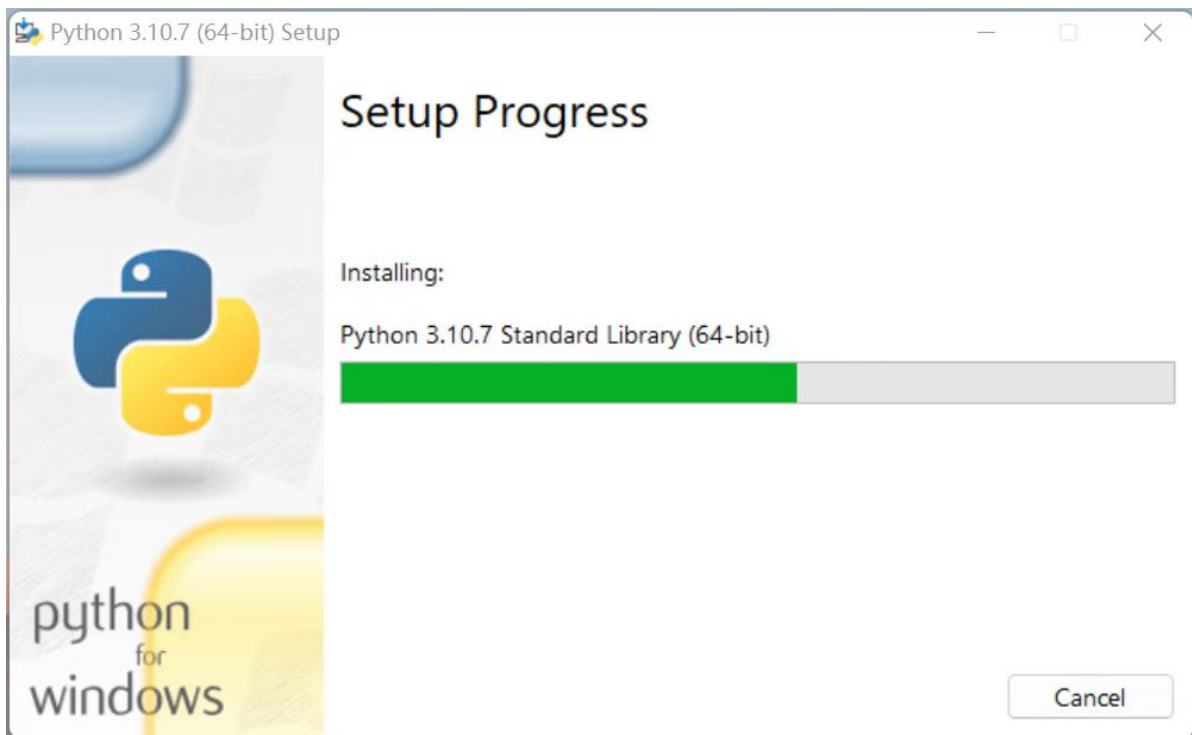


The screenshot shows the Python.org website. The 'Downloads' tab is highlighted in the top navigation bar. A dropdown menu is open, showing options like 'All releases', 'Source code', 'Windows', 'macOS', 'Other Platforms', 'License', and 'Alternative Implementations'. The 'Windows' option is selected, leading to the 'Download for Windows' section. This section features a 'Python 3.10.7' button and a note that Python 3.9+ cannot be used on Windows 7 or earlier. Below this, there is a table titled 'Active Python Releases'.

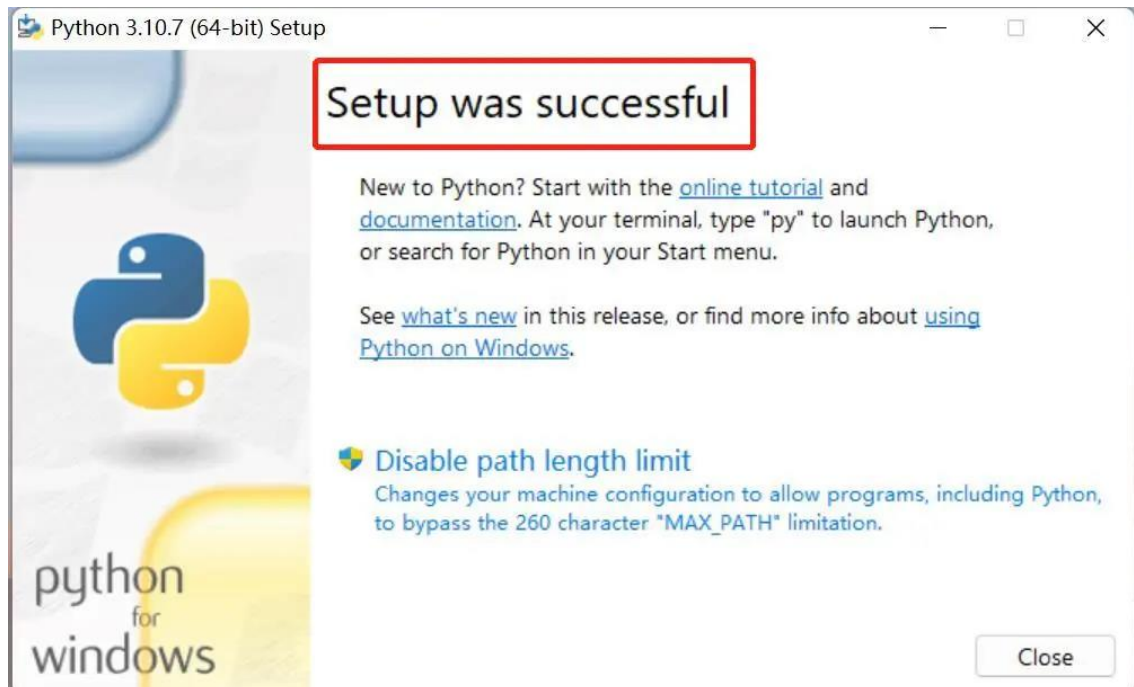
Python version	Maintenance status	First released	End of support	Release schedule
3.10	bugfix	2021-10-04	2026-10	PEP 619



The screenshot shows the 'Python 3.10.7 (64-bit) Setup' window. The title bar reads 'Python 3.10.7 (64-bit) Setup'. The main heading is 'Install Python 3.10.7 (64-bit)'. Below this, it says 'Select Install Now to install Python with default settings, or choose Customize to enable or disable features.' There are two main options: 'Install Now' and 'Customize installation'. The 'Install Now' option is highlighted with a red box and includes the path 'C:\Users\Surface\AppData\Local\Programs\Python\Python310', 'Includes IDLE, pip and documentation', and 'Creates shortcuts and file associations'. The 'Customize installation' option is also visible. At the bottom, there are two checkboxes: 'Install launcher for all users (recommended)' and 'Add Python 3.10 to PATH'. The 'Add Python 3.10 to PATH' checkbox is checked and highlighted with a red box. A 'Cancel' button is located at the bottom right.



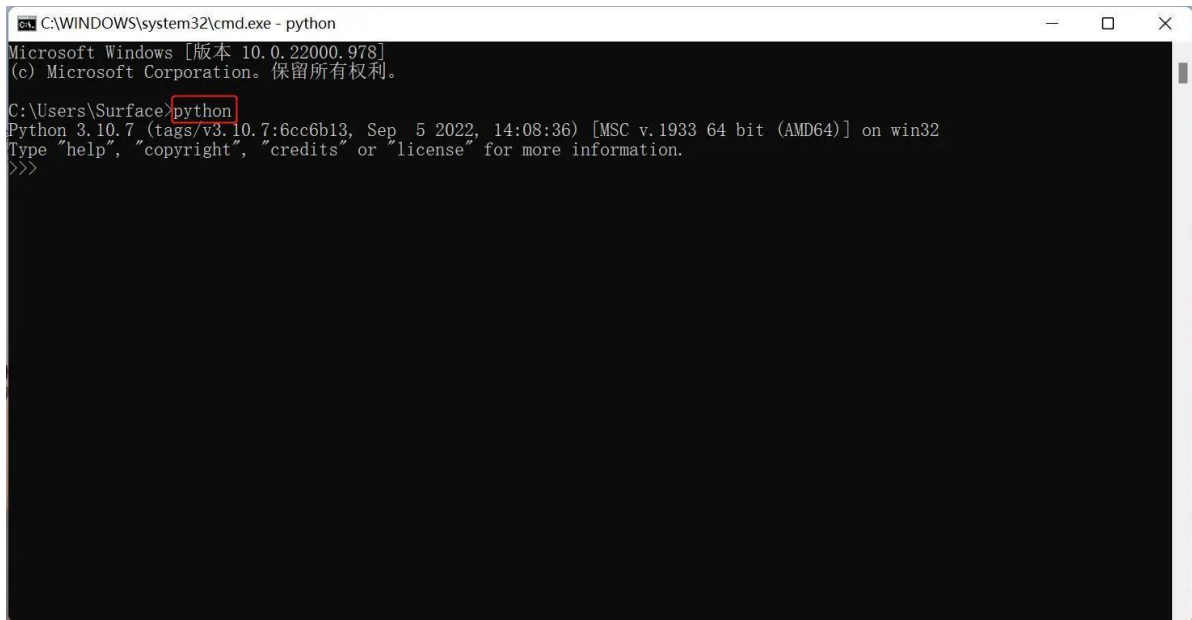
- **Download and installation complete.**



1.2 Running Python

Open the command prompt window (Win+R, input `cmd` and press `Enter`). Type `Python`.

Successful Installation:



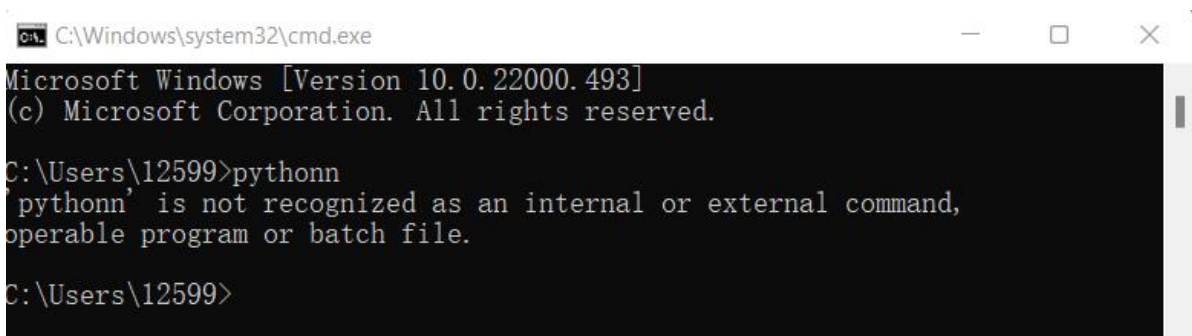
```
C:\WINDOWS\system32\cmd.exe - python
Microsoft Windows [版本 10.0.22000.978]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Surface>python
Python 3.10.7 (tags/v3.10.7:6cc6b13, Sep 5 2022, 14:08:36) [MSC v.1933 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

This on-screen instruction means that Python is successfully installed. The prompt `>>>` means Python interactive environment. If you input a Python code to get the execution result immediately.

Error Report:

If a wrong instruction is typed, for example "pythonn", the system may report an error.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.22000.493]
(c) Microsoft Corporation. All rights reserved.

C:\Users\12599>pythonn
'pythonn' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\12599>
```

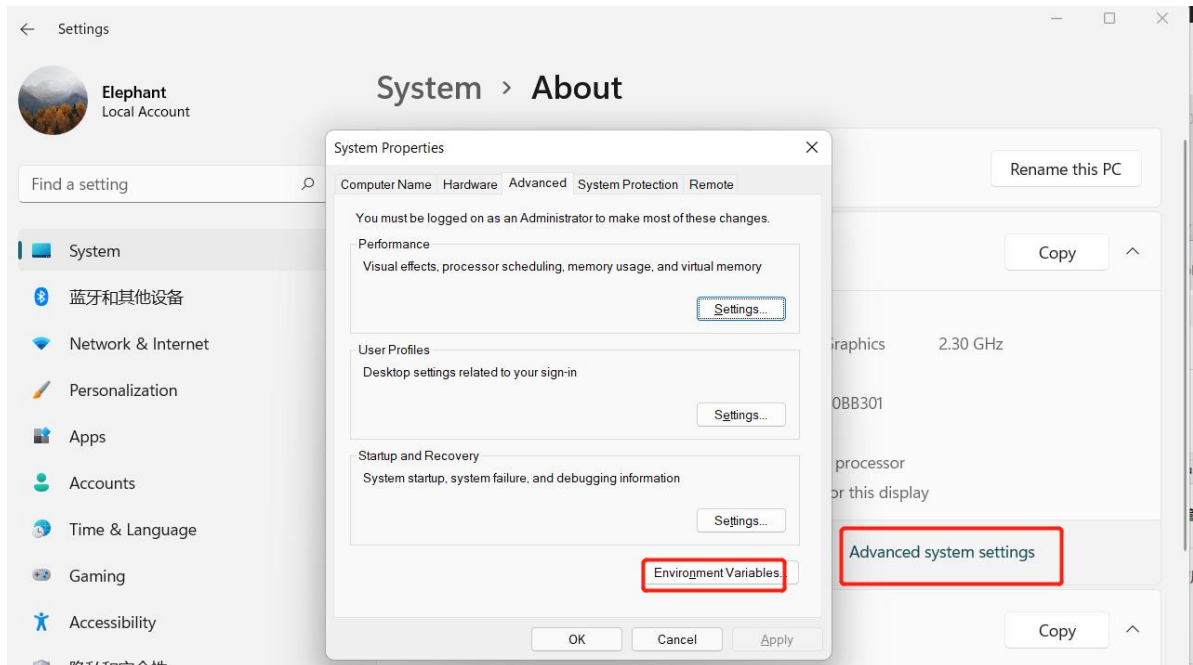
Notice: Generally, the error results from lack of environment configuration. Refer to **1.3 Environment Configuration** to solve problems.

1.3 Environment Variable Configuration

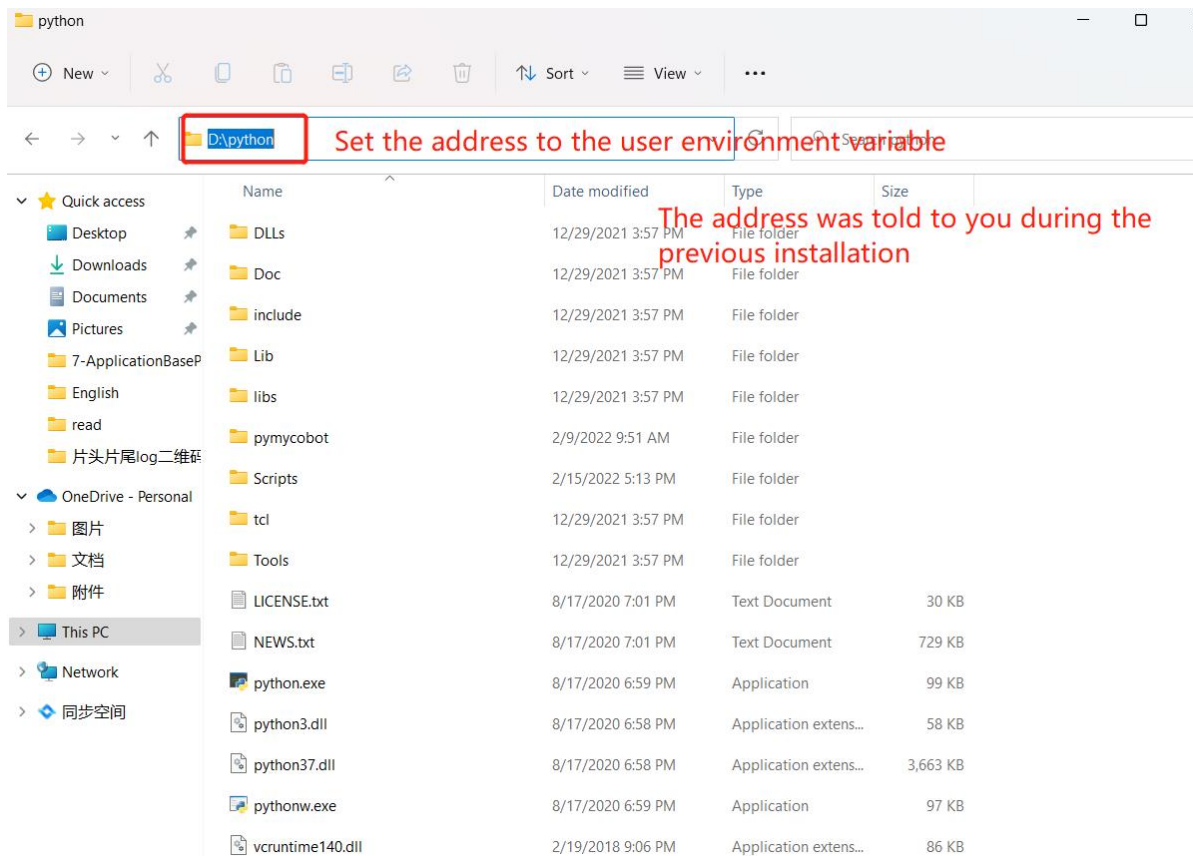
Windows follows the path set by a Path environment variable in search of **python.exe**. Otherwise, an error will be reported. If you fail to tick **Add Python 3.9 to PATH** during installation, you need to manually add the path where python.exe is located into environment variable or download python again. Remember to tick **Add Python 3.9 to PATH**.

Follow the steps below to add python into environment variable manually.

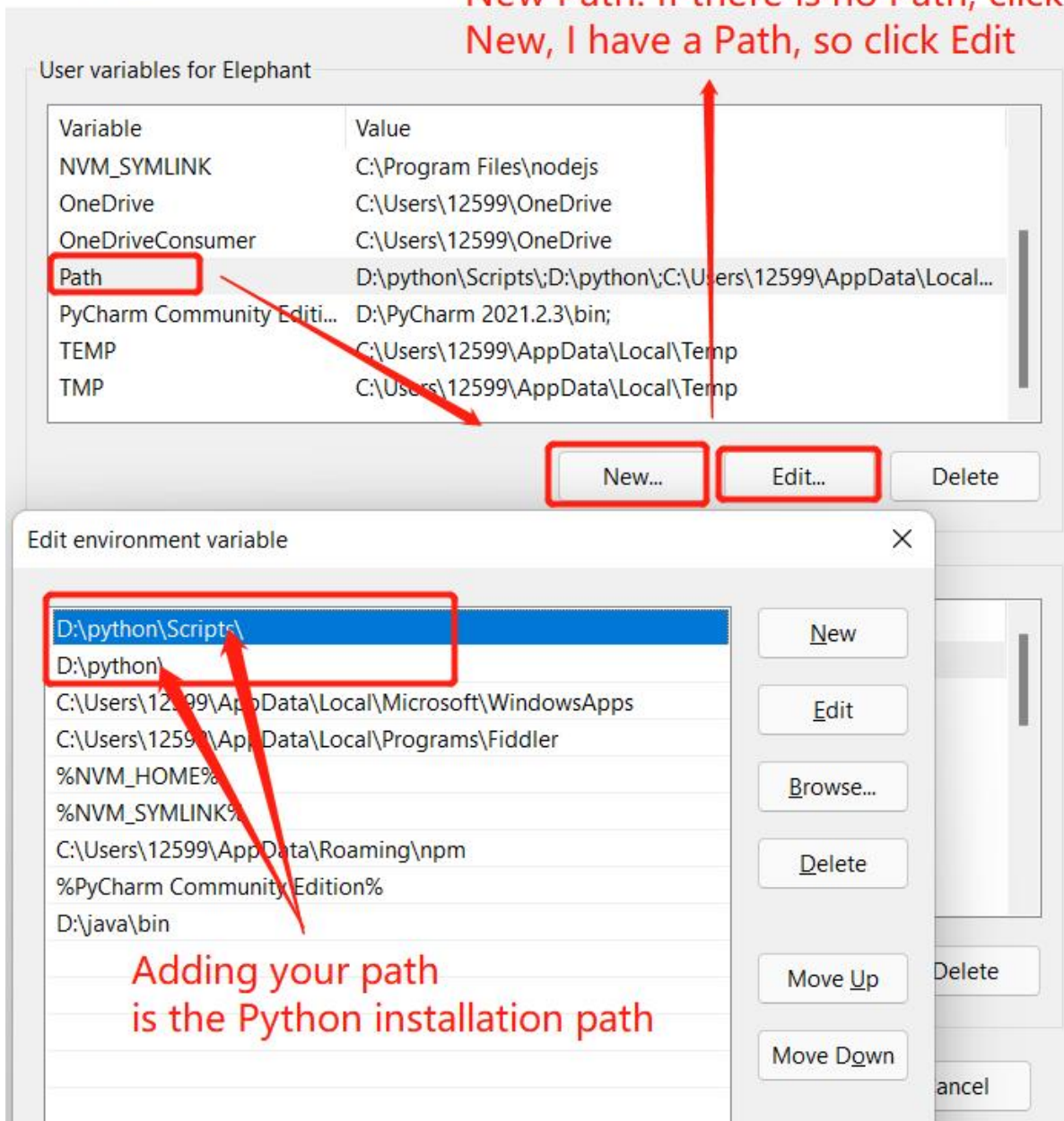
- Right click on **My Computer** icon --> Properties --> Advanced System Settings --> Environment Variables



- The environment variables include user variables and system variables. For user variables, users can utilize their own downloaded programs via `cmd` command. Write the absolute path of the target program into the user variables.



New Path: If there is no Path, click New, I have a Path, so click Edit



- After the configuration, open the command prompt window (Win+R; input `cmd` and press `Enter`), and type `Python`.

```

C:\Windows\system32\cmd.exe - python
Microsoft Windows [Version 10.0.22000.493]
(c) Microsoft Corporation. All rights reserved.

C:\Users\12599>python
Python 3.7.9 (tags/v3.7.9:13c94747c7, Aug 17 2020, 18:58:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
  
```

2 Installation of PyCharm

PyCharm is a powerful python editor with the nature of cross-platform. Follow the steps below to download and install PyCharm.

Go to [PyCharm](#) to download PyCharm.

2.1 Download and Installation

Official website view:



Version: 2022.2.3
Build: 222.4345.23
11 October 2022

[System requirements](#)

[Installation instructions](#)

[Other versions](#)

[Third-party software](#)

Download PyCharm

[Windows](#)

[macOS](#)

[Linux](#)

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

[Download](#)

Free 30-day trial available

Community

For pure Python development

[Download](#)

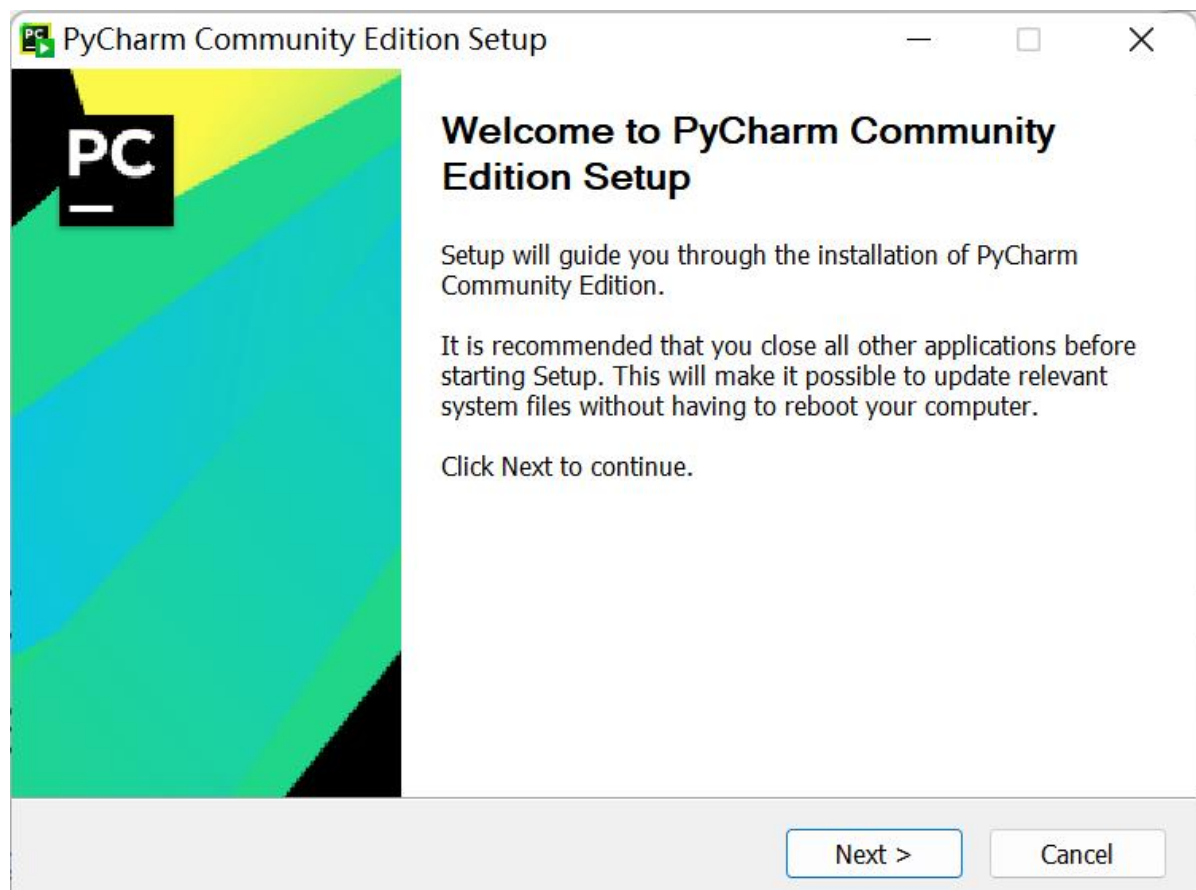
Free, built on open-source



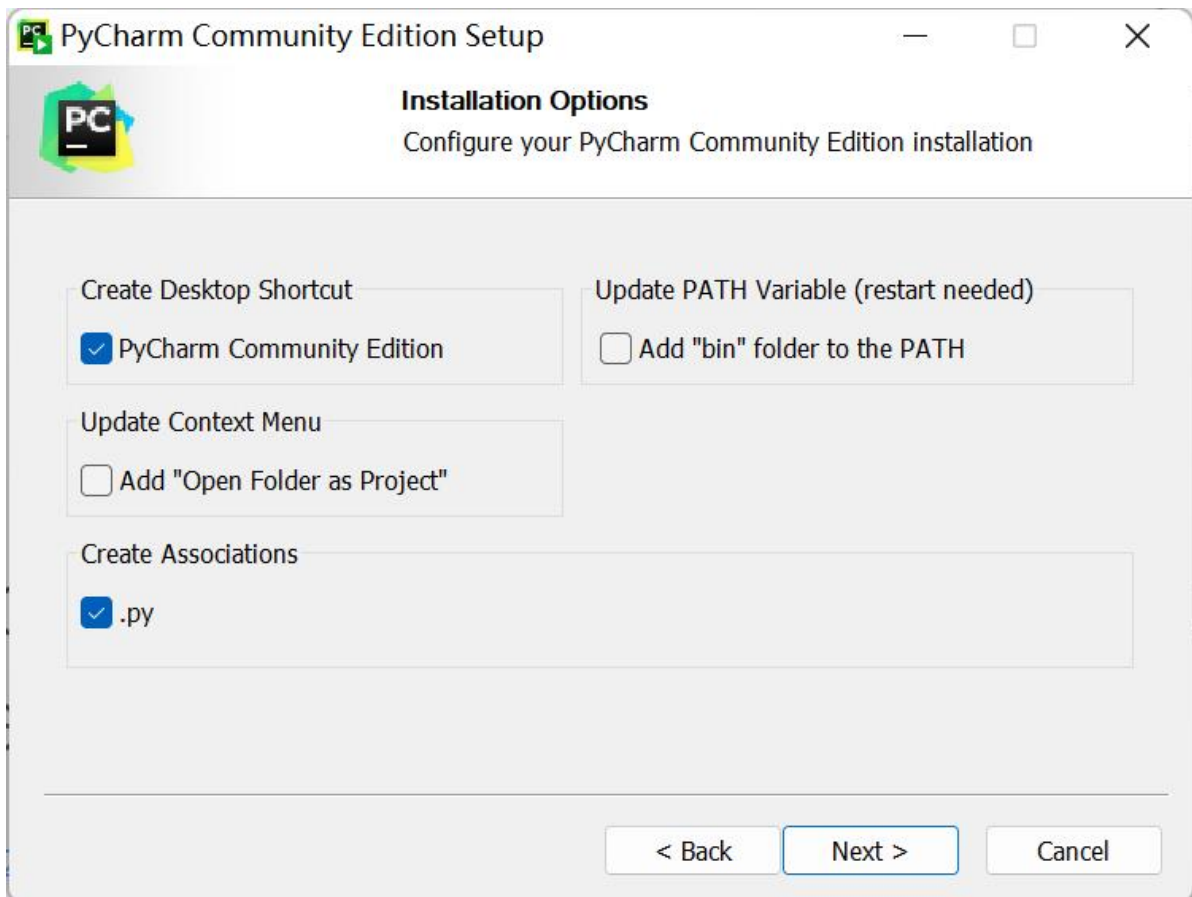
Get the Toolbox App to download PyCharm and its future updates with ease

It is recommended to install the free version.

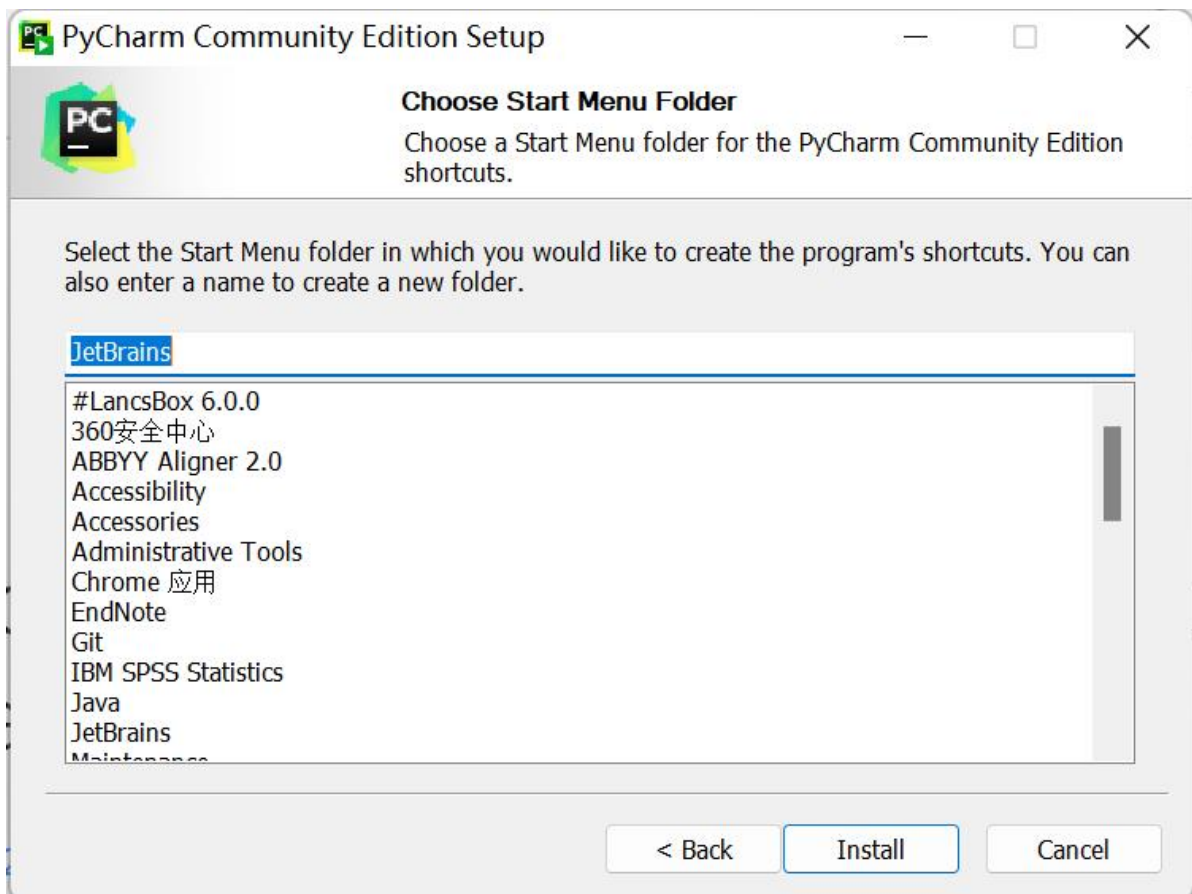
- Click on **Next** :



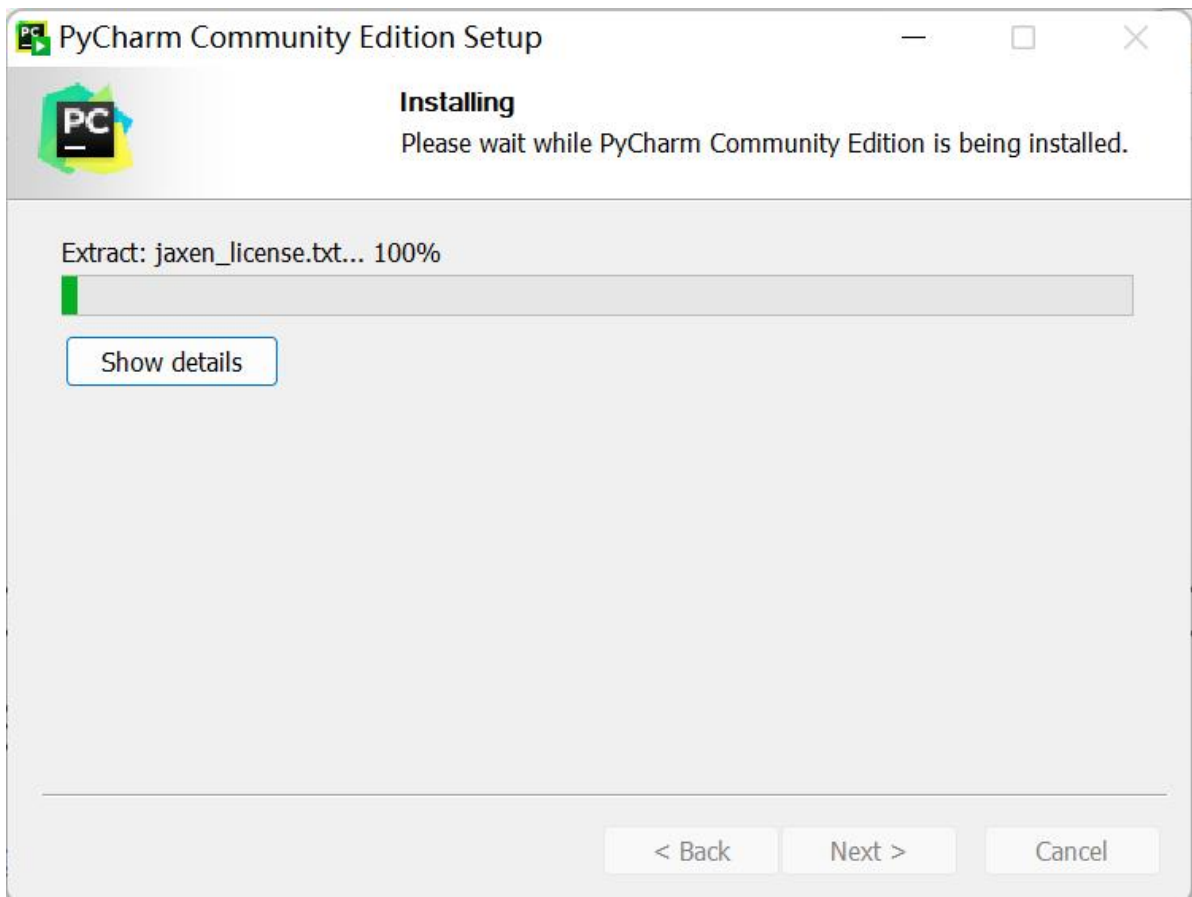
- Select options according to your needs and then select **Next** :



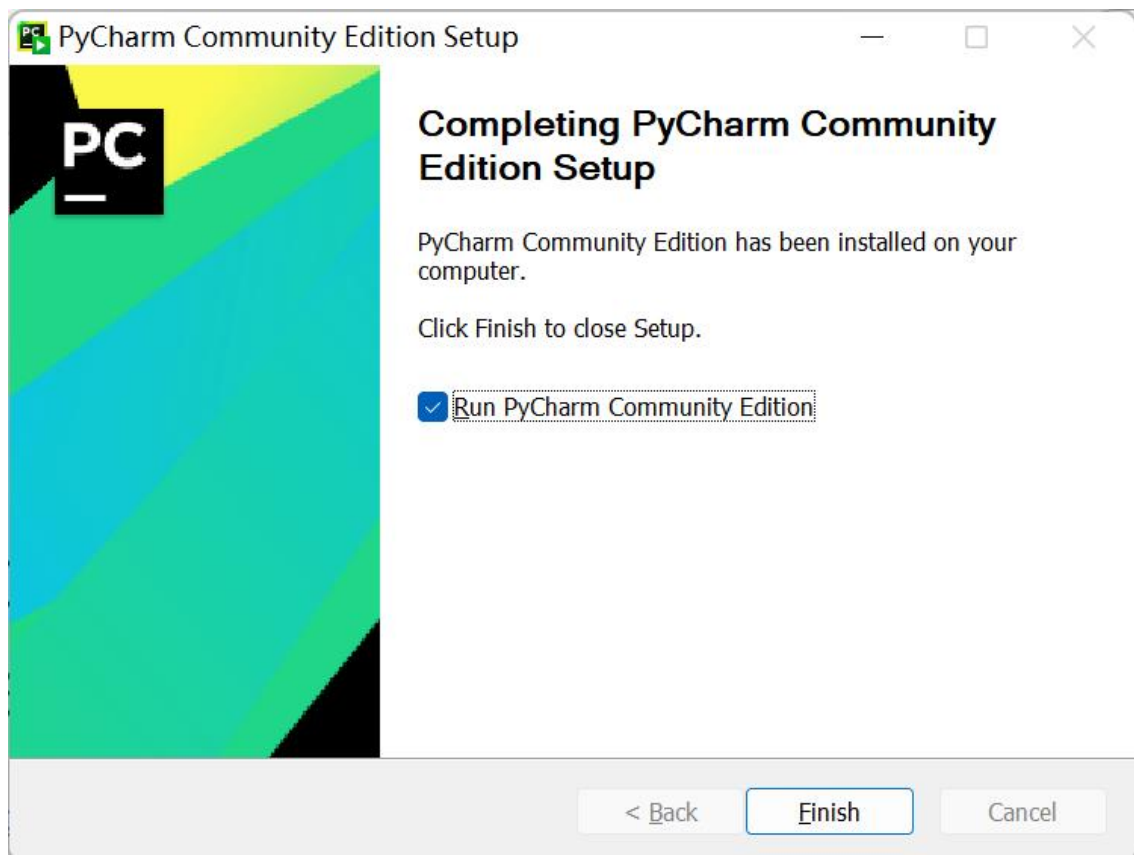
- Tap **Install**:



- Installing:

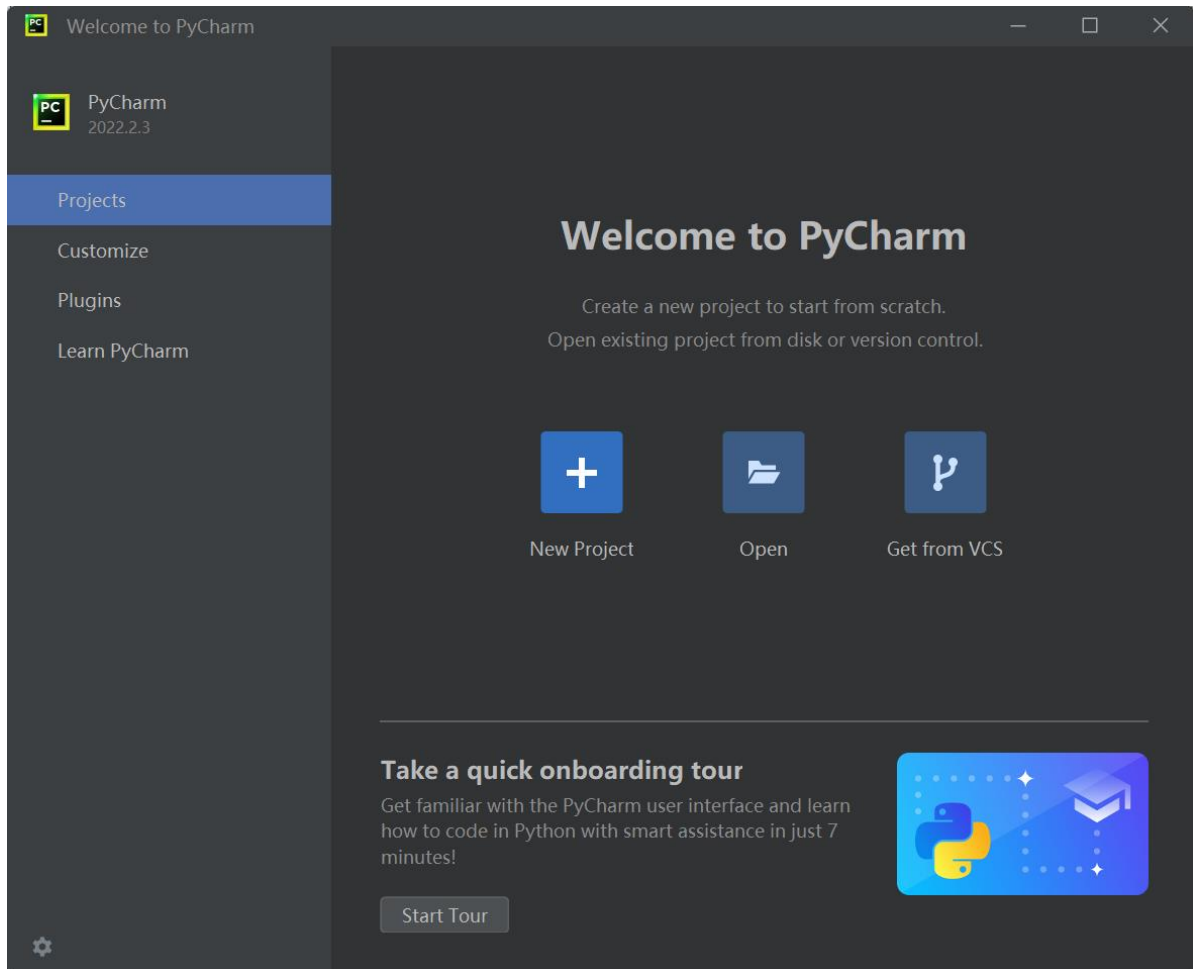


- Tap **Finish**

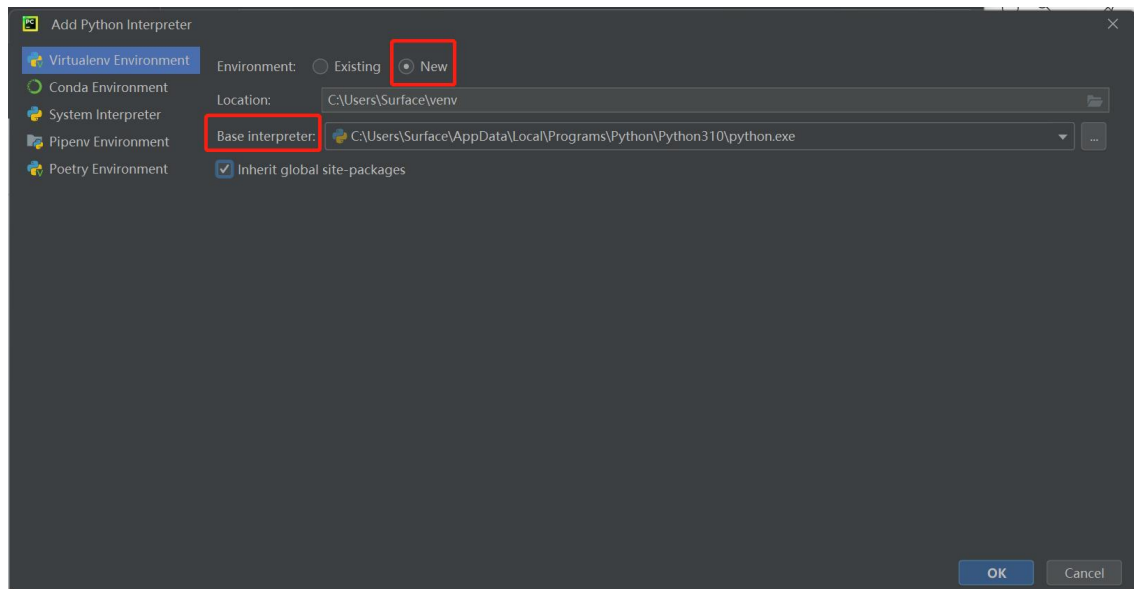
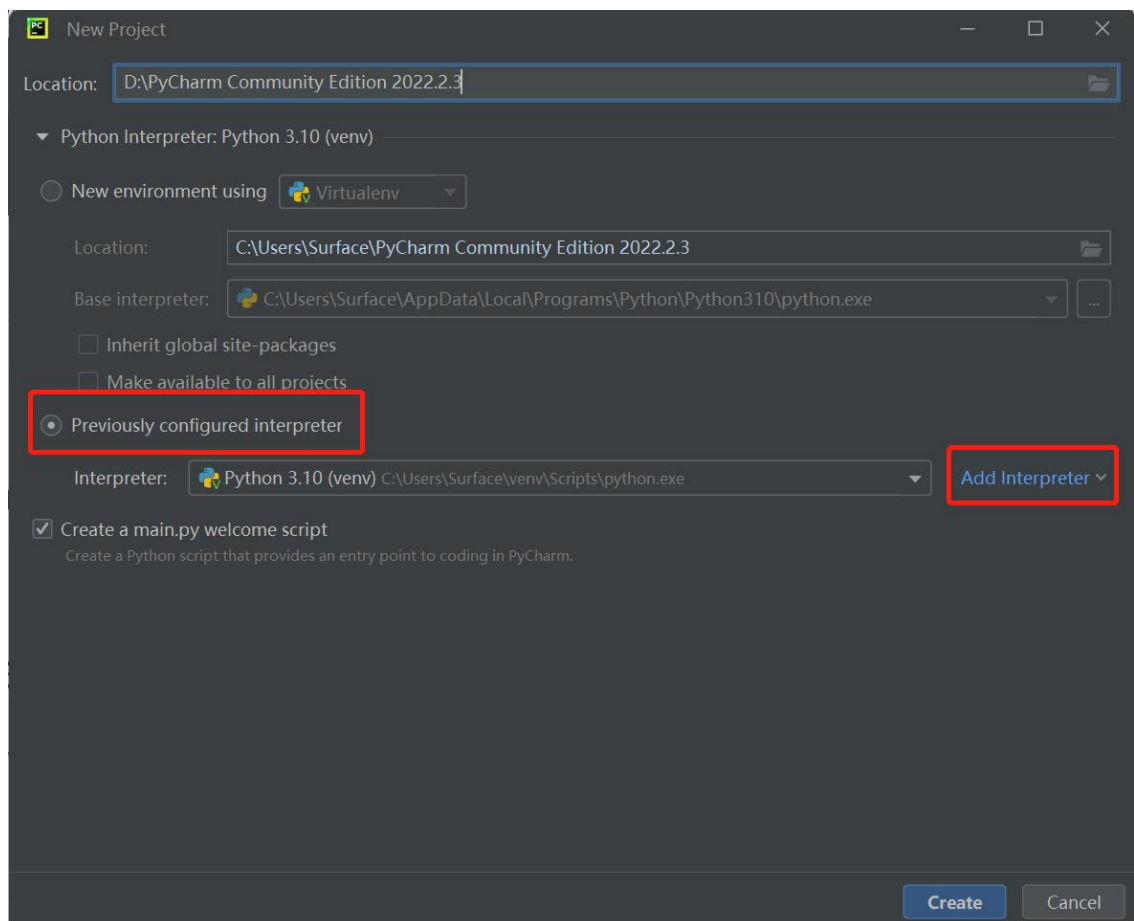


2.2 Create a new project

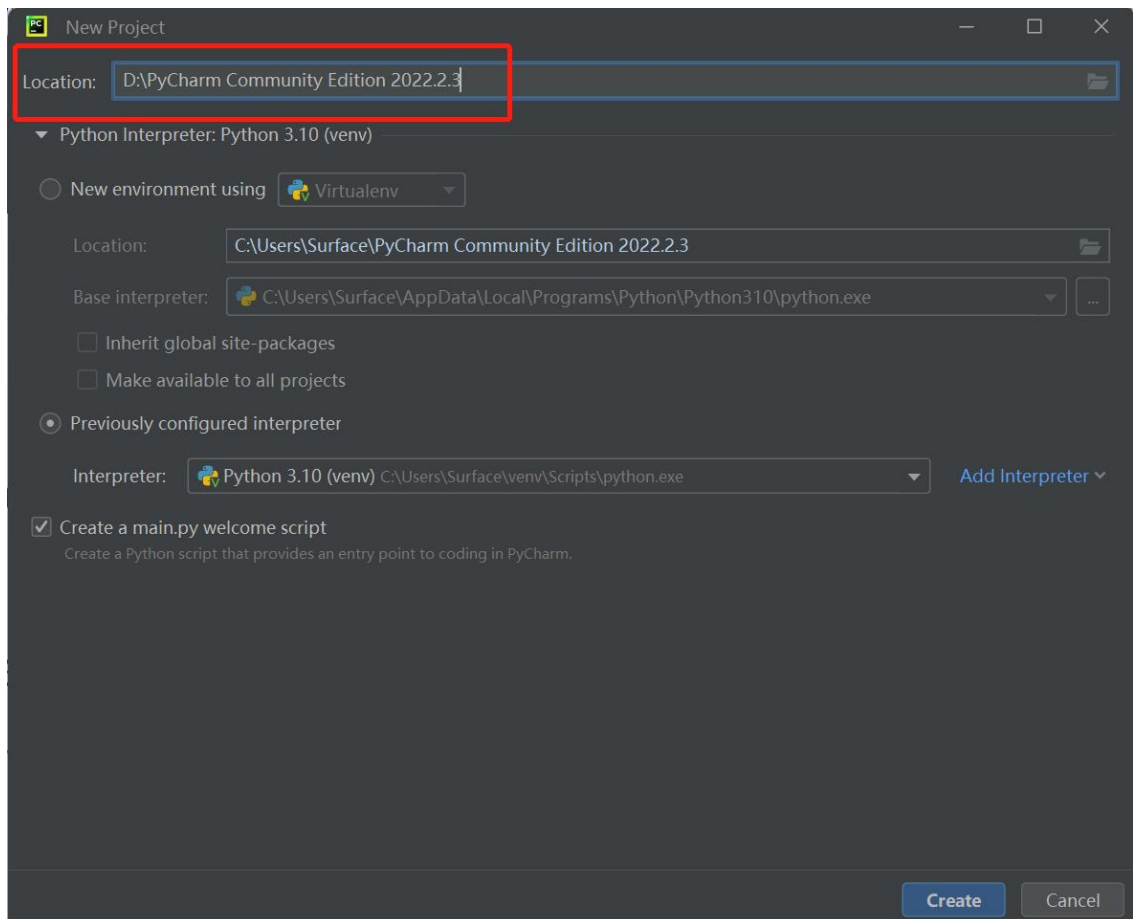
- Click `+New Project` :



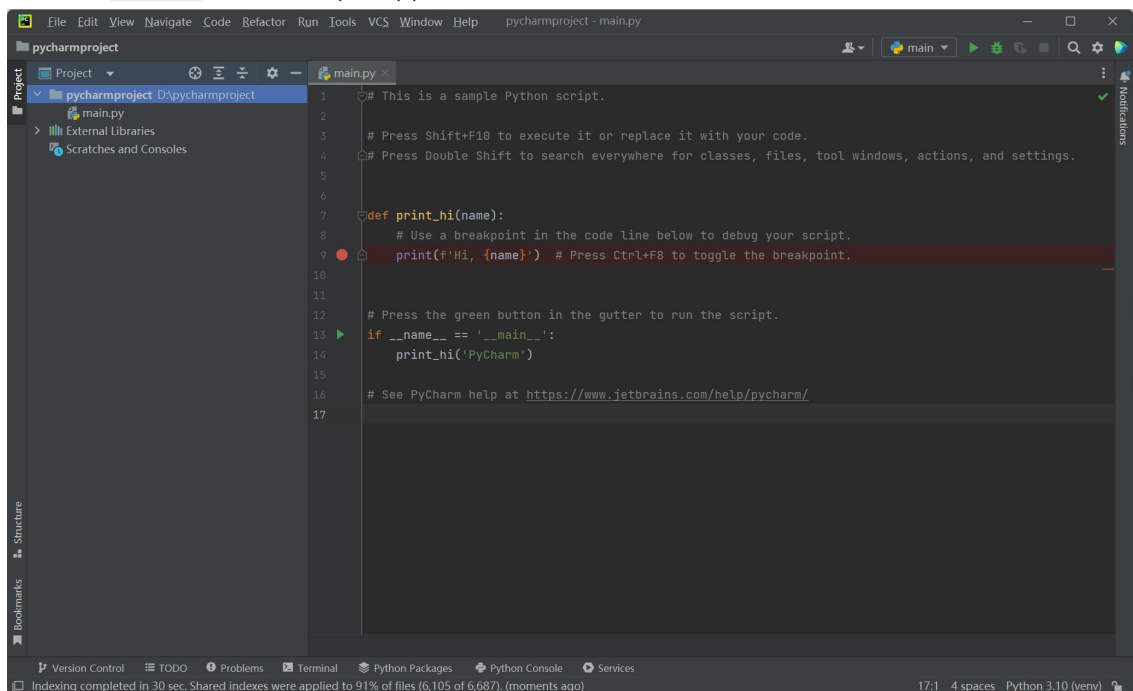
- The `Interpreter` is used to interpret python programs. Select `Add Interpreter` -> `New` to add base interpreter.



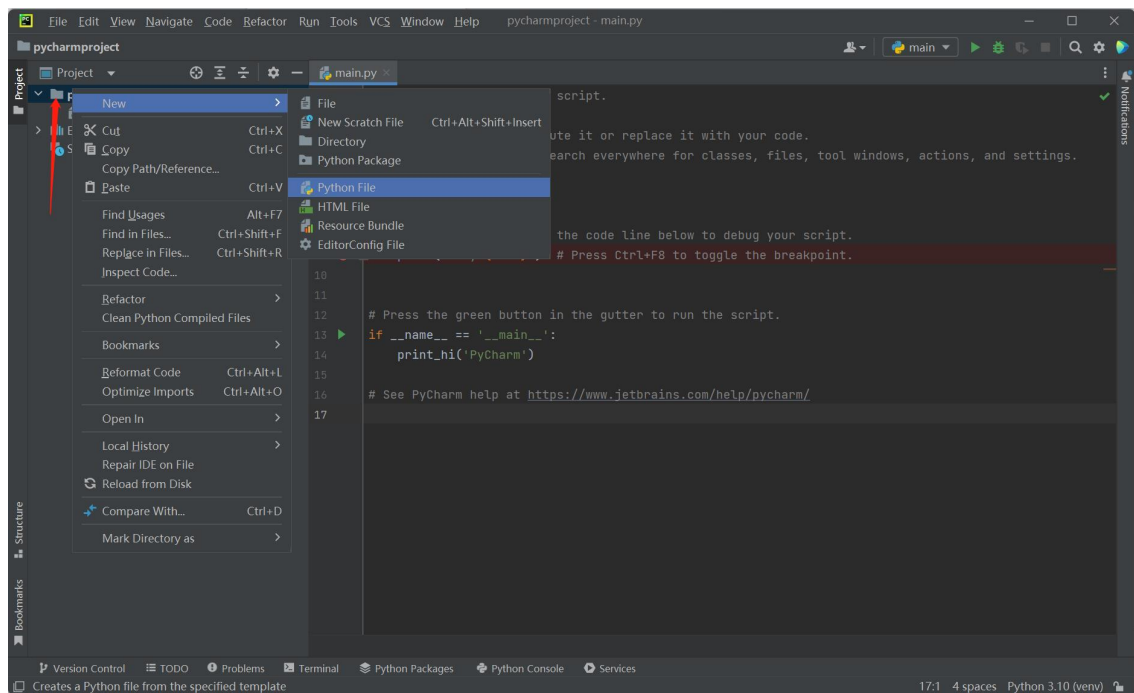
- **Location** refers to the place where to save python file. Choose a file to put your programs.



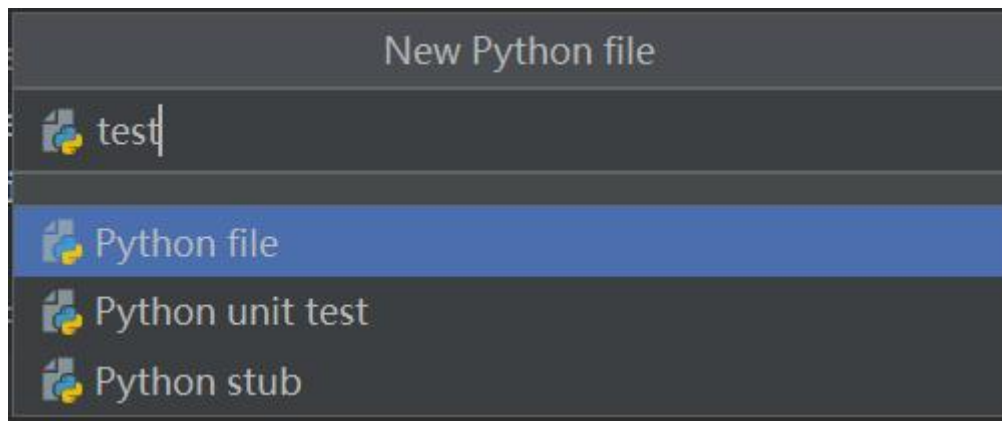
- Click on **Create** and a sample appears:



- Right click on the selection that the red arrow points, and create a new python file.



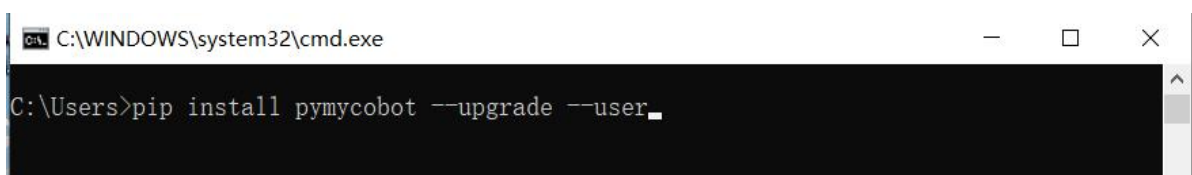
- Type name for the new file.



3 Preparations

- Firmware burning. Firmware serves as a driver for systems to control robots.
 - **M5Stack version** Make sure to burn `minirobot` for **Basic** at the bottom. And then choose `Transponder` function (to receive instructions from Basic), Press `Press A`. `Atom: OK` means connect successfully. Refer to [MyStudio](#) for more information about firmware burning.
 - **Pi \jetsonnano version** `AtomMain` for **Atom** at the top is factory burnt.
- pymycobot installation. Type `pip install pymycobot --upgrade --user` via terminal (Win+R) `cmd` command.

```
pip install pymycobot --upgrade --user
```



- Source code installation. Open a terminal (Win+R, input `cmd`), and type the command below to install.


```
git clone https://github.com/elephantrobotics/pymycobot.git <your-path>
#Fill in your installation address in <your-path>, do not choose the current
default path.
```

```
cd <your-path>/pymycobot
#Go to the pymycobot folder of the downloaded package.
```

```
#Run one of the following commands according to your python version.
```

```
# Install
```

```
python2 setup.py install
```

```
# or
```

```
python3 setup.py install
```

- Update pymycobot

```
pip install pymycobot --upgrade
```

4 Import of pymycobot

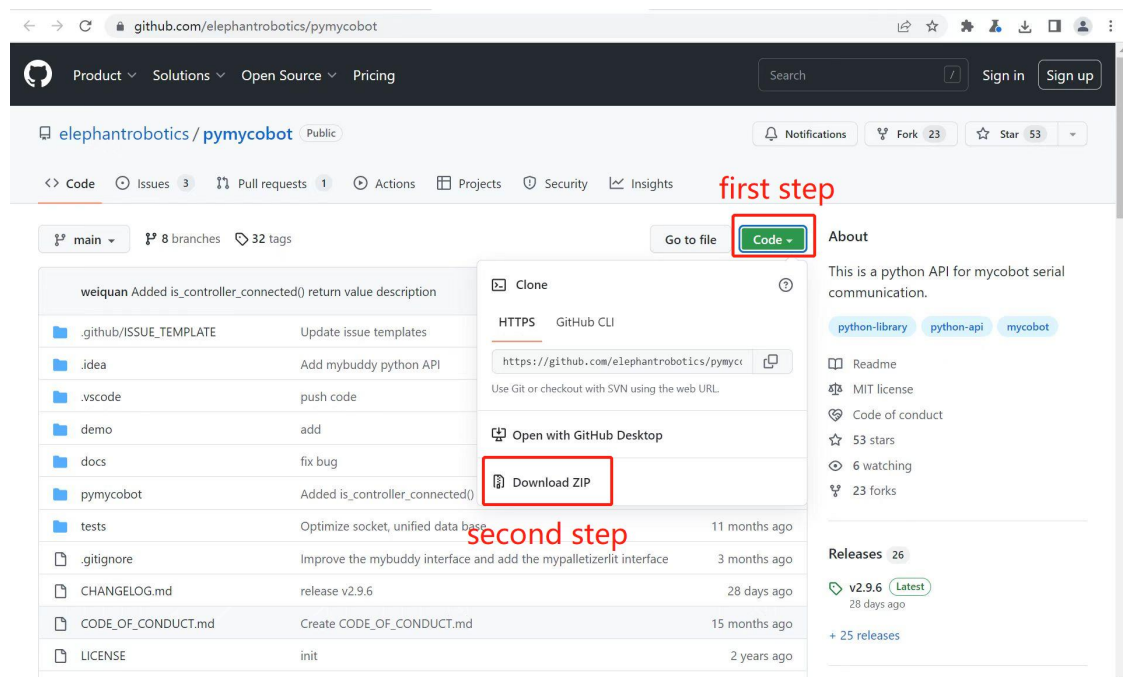
This part takes MyPalletizer 260-M5 as an example to introduce how to control a robot via python.

- Import pymycobot library for MyPalletizer :

```
from pymycobot.mypalletizer import MyPalletizer
```

Notice:

1. If no red wavy line appears below the codes, pymycobot is successfully installed.
2. if a red wavy line appears, got to the address <https://github.com/elephantrobotics/pymycobot> to download pymycobot manually and put it into python library.



5 Simple Demo

Create a new Python file, and type the following codes to set the color of RGB light panel. The codes are suitable to myCobot 280-M5, myCobot 320-M5 and myPalletizer 260.

Notice: The baud rates are different according to types of devices. Refer to [calculator device manager](#) to check the corresponding number.

- **Codes for MyCobot:**

```
# demo.py
from pymycobot.mycobot import MyCobot

from pymycobot import PI_PORT, PI_BAUD      # When using the Raspberry Pi
version of mycobot, you can refer to these two variables to initialize MyCobot,
if not, you can omit this line of code
import time
#The above codes are required to be written, which means importing the project
package

# MyCobot class initialization requires two parameters:
#   The first is the serial port string, such as:
#       linux: "/dev/ttyUSB0"
#       or "/dev/ttyAMA0"
#       windows: "COM3"
#   The second is the baud rate::
#       M5 version is: 115200
#
#   Example:
#       mycobot-M5:
#           linux:
#               mc = MyCobot("/dev/ttyUSB0", 115200)
#               or mc = MyCobot("/dev/ttyAMA0", 115200)
#           windows:
#               mc = MyCobot("COM3", 115200)
#       mycobot-raspi:
#           mc = MyCobot(PI_PORT, PI_BAUD)
#
# Initiate MyCobot
# Create object code here for windows version
mc = MyCobot("COM3", 115200)

i = 7
#loop 7 times
while i > 0:
    mc.set_color(0,0,255) #blue light on
    time.sleep(2)        #wait for 2 seconds
    mc.set_color(255,0,0) #red light on
    time.sleep(2)        #wait for 2 seconds
    mc.set_color(0,255,0) #green light on
    time.sleep(2)        #wait for 2 seconds
    i -= 1
```

Run the example file:

```
python3 demo.py
```

- **Codes for MyPalletizer:**

```

from pymycobot.mypalletizer import MyPalletizer
import time
##The above codes are required to be written at the beginning, which means
importing the project package

#initiate MyPalletizer
mc = MyPalletizer("COM3", 115200)

#    MyPalletizer initiation requires two parameters:
#    The first is the serial port string, such as:
#        linux: "/dev/ttyUSB0"
#            or "/dev/ttyAMA0"
#        windows: "COM3"
#    The second is the baud rate::
#        M5 version is: 115200
#
#    Example:
#        MyPalletizer-M5:
#            linux:
#                mc = MyPalletizer("/dev/ttyUSB0", 115200)
#            or mc = MyPalletizer("/dev/ttyAMA0", 115200)
#            windows:
#                mc = MyPalletizer("COM3", 115200)

i = 7
#loops 7 times
while i > 0:
    mc.set_color(0,0,255) #blue light on
    time.sleep(2) #wait for 2 seconds
    mc.set_color(255,0,0) #red light on
    time.sleep(2) #wait for 2 seconds
    mc.set_color(0,255,0) #green light on
    time.sleep(2) #wait for 2 seconds
    i -= 1

```

The blue, red, and green lights on the top of the robot flash 7 times continuously at an interval of 2 seconds.