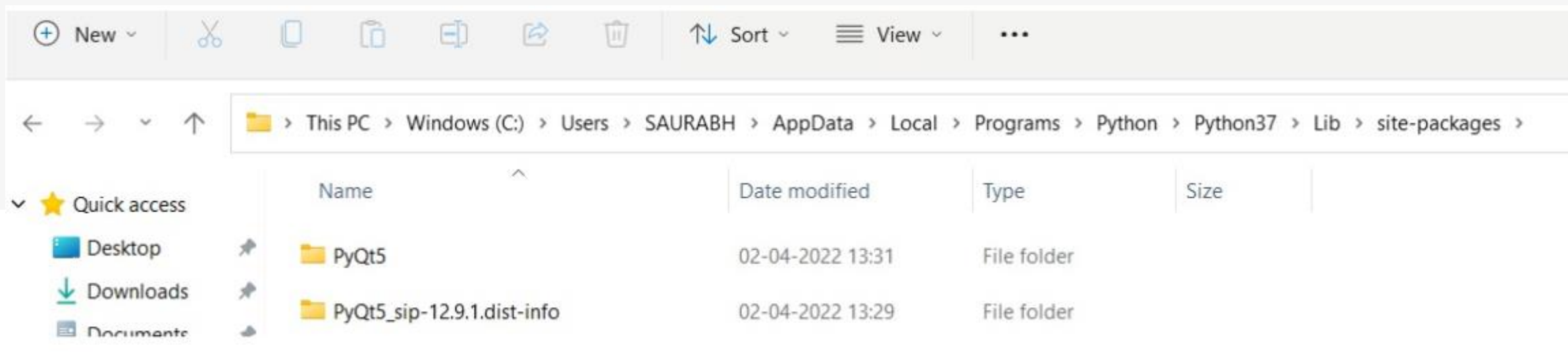


```
C:\Users\SAURABH\AppData\Local\Programs\Python\Python37\Lib\site-packages>python --version
Python 3.7.3

C:\Users\SAURABH\AppData\Local\Programs\Python\Python37\Lib\site-packages>pip install pyqt5
Requirement already satisfied: pyqt5 in c:\users\saurabh\AppData\Local\Programs\Python\Python37\Lib\site-packages (5.15.4)
Requirement already satisfied: PyQt5-Qt5>=5.15 in c:\users\saurabh\AppData\Roaming\Python\Python37\site-packages (from pyqt5) (5.15.2)
Requirement already satisfied: PyQt5-sip<13,>=12.8 in c:\users\saurabh\AppData\Local\Programs\Python\Python37\Lib\site-packages (from pyqt5) (12.9.1)
```



PyQt5 framework installation

Kiểm tra xem PyQt5 có chưa

```
C:\Users\SAURABH>python
Python 3.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 22:22:05) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from PyQt5 import QtGui
>>>
```

Tạo giao diện người dùng (GUI) PyQt5 không sử dụng CLASS.

```
import sys # L1
from PyQt5.QtWidgets import QWidget, QApplication # L2
myapp = QApplication([]) # L3
mywindow = QWidget() # L4
mywindow.show() # L5
sys.exit(myapp.exec_()) # L6
```

Ở dòng L1, nhập mô-đun sys để truy cập các đối số dòng lệnh.

Ở dòng L2, nhập lớp tạo giao diện người dùng.

Ở dòng L3, tạo một thể hiện của QApplication với tên myapp và truyền danh sách tham số dòng lệnh sys.argv cho ứng dụng. hoặc có thể truyền dưới dạng một danh sách trống như sau: myapp = QApplication([])

Trong L4, tạo một đối tượng của lớp QWidget bằng cách sử dụng tên biến là mywindow. Lớp QWidget là cha của tất cả các đối tượng giao diện người dùng trong Qt. Cửa sổ cấp cao nhất được tạo ra mà không cần truyền bất kỳ tham số nào cho lớp QWidget.

Ở dòng 5, để hiển thị các widget sau khi tạo đối tượng của lớp QWidget, ta phải gọi phương thức show().

Ở dòng 6, để thực hiện thực thi song song, PyQt5 sử dụng cơ chế vòng lặp sự kiện vì nó được viết chủ yếu bằng C++. Vì vậy, để khởi động cơ chế vòng lặp sự kiện, ta gọi phương thức myapp.exec_(), phương thức này sẽ được giữ bởi đối tượng ứng dụng.

Ta sử dụng exec_() trong PyQt5. Do đó, khi người dùng yêu cầu đóng giao diện người dùng đồ họa (GUI), myapp.exec_() sẽ cho phép chuyển quyền điều khiển cho Qt để kết thúc ứng dụng.

Thiết lập tiêu đề và kích thước

- **`mywindow.setWindowTitle('Basic GUI Form')`**
- **`mywindow.resize(400,350)`**

Tạo giao diện người dùng (GUI) PyQt5 bằng cách sử dụng lớp.

```
import sys # CL1
from PyQt5.QtWidgets import QWidget, QApplication # CL2
class GUIWindow(QWidget):
    def __init__(self):
        super(GUIWindow, self).__init__() # CL6
        self.initializationUI() # CL7
    def initializationUI(self):
        self.show() # CL8
if __name__ == '__main__':
    myapp = QApplication(sys.argv) # CL3
    mywindow = GUIWindow() #CL4
    sys.exit(myapp.exec_()) # CL5
```

Trong CL4, là tạo một đối tượng của lớp GUIWindow do người dùng định nghĩa. Lớp GUIWindow đang được bắt nguồn từ lớp cha QWidget, từ đó các không gian và cửa sổ được kế thừa trực tiếp hoặc gián tiếp.

Hơn nữa, tất cả các đối tượng UI sẽ được bắt nguồn từ lớp QWidget này.

Trong CL6, gọi phương thức `__init__` của lớp cha QWidget. Nói cách khác, có thể nói rằng mặc định hàm tạo của lớp QWidget được tạo.

Trong CL7, tạo phương thức khởi tạoUI trong lớp GUIWindow của mình.

Trong CL8, bên trong phương thức khởi tạoUI, ta gọi phương thức hiển thị

Có thể thiết lập tiêu đề và kích thước form trước khi `show()`

- `self.setGeometry(300,300,400,300)`
- `self.setWindowTitle("Basic GUI Form")`

self.setGeometry(x-axis, y-axis, width, height)

`self.setGeometry(300,300,400,300)`

- Tham số đầu tiên là tọa độ X, được đặt là 300.
- Tham số thứ hai là tọa độ Y, được đặt là 300.
- Tham số thứ ba là chiều rộng cửa sổ, được đặt là 400.
- Tham số thứ tư là chiều cao cửa sổ, được đặt là 300.
- Như vậy, có thể thấy rằng với sự trợ giúp của việc sử dụng các lớp, chúng ta có thể tạo một biểu mẫu GUI, đặt tiêu đề, xác định hình dạng form.