Basic GUI Application

What is PyQt5

PyQt5 is python module that allows you to build GUI applications very quickly.

Installing PyQt5

The first step to start using PyQt5 is to install it!
To do this we will need to use **pip**.
Open up your command prompt and try executing the following commands to install PyQt5.

- pip install pyqt5
- pip install pyqt5-tools

Creating a Basic GUI App

Now that we have installed PyQt we can start using it.
Open your favorite text editor and follow along with the code below.
We will start with importing the necessary modules and classes.

```
from PyQt5 import QtWidgets
from PyQt5.QtWidgets import QApplication, QMainWindow, QLabel
import sys
```

Whenever we create a PyQt application we need to define a **QApplication**. This will be where we can place our window and widgets. Next, we can decide on what we want to go in our application. In my case I've started by creating a **QMainWindow**. You can think of this like a container that will hold all our widgets (buttons, labels, etc.). Below you can see I've given my window a size and a title.

```
def main ():
   app = QApplication (sys. argv)
   win = QMainWindow ()
   win.setGeometry(200,200,300,300) # sets the windows x, y, width, heigh
   win.setWindowTitle("My first window!") # setting the window title
```

Sweet now we have a window with a title. Now it's time to add some stuff to it. For our purposes well start small and add a label.
To add a label, we will create one and tell PyQt where it should be going. Next, we'll set the text on the label and the position.

```
label = QLabel(win)
label. setText("my first label")
label.move(50, 50)
```

To create a button, we will follow a similar procedure as to when we created the label. Place the following into the function.

```
b1 = QtWidgets.QPushButton(win)
b1.setText("click me")
b1.move(50,50) to move the button
```

If we want to see the label and the button now, we will need to show the window.

```
win.show()
```

To ensure what we call a clean exit (one without an error) we need to add the following line to the bottom of our function.

```
sys.exit(app.exec_())
```

Events & Signals

The next thing to do is link our button to some kind of event so that something happens when it is clicked. This brings us to a new topic called **Event-Driven-Programming**. I won't explain this in detail as it's intuitive but essentially the way that PyQt and many other GUI frameworks work is by waiting for events or signals (terms interchangeable) to occur. An example of an event/signal is a mouse hover over a button. When these events occur, they trigger something to run (usually a function or method) that will handle what should happen.

For our case we need to handle a press on the button. So, we will start by creating a function that can be called when the event is triggered.

```
def clicked():
  print("clicked") # we will just print clicked when the button is pressed
```

Next, we will link the button click event the that main function. This is done with the following line:

```
b1.clicked.connect(clicked)
```