What is PyQt5

PyQt5 is python module that allows you to very quickly build GUI applications. It is often compared to Tkinter but has a few differences. The main one is the speed at which you can develop GUI's. PyQt comes with an amazing tool called Qt-Designer which is a drag and drop interface that automatically generates code for you based on the GUI you build. This makes it **much** faster to build good looking applications. Another difference is that Tkinter comes by default with python 3+, where PyQt must be installed.

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

If these don't work watch this video to learn how to fix pip: How to Fix Pip.

Creating a Basic GUI App

Now that we have installed PyQt we can start using it.

Open up 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 of 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, height
    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 (in later tutorials we will add much more).

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.

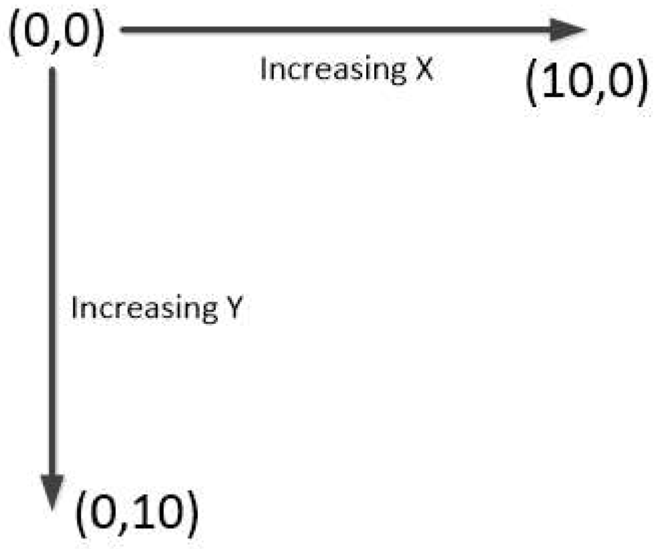
```
    label = QLabel(win)
    label.setText("my first label")
    label.move(50, 50)  # x, y from top left hand corner.
```

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

```
    win.show()
```

PyQt Coordinate System

A quick note here on how items are positioned in the pyqt window. In typical graphs (0,0) or the origin is positioned in the middle or the bottom left. Where increasing the Y value makes the point move up and increasing the X value makes the point move right. However, in pyqt and many GUI frameworks (0,0) is the top right. This means increasing Y actually makes the point move down.



Exiting the Program

In order 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_())
```

Full Code

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

def main():
    app = QApplication(sys.argv)
    win = QMainWindow()
    win.setGeometry(200,200,300,300)
    win.setWindowTitle("My first window!")

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

    win.show()
    sys.exit(app.exec_())

main()  # make sure to call the function
```