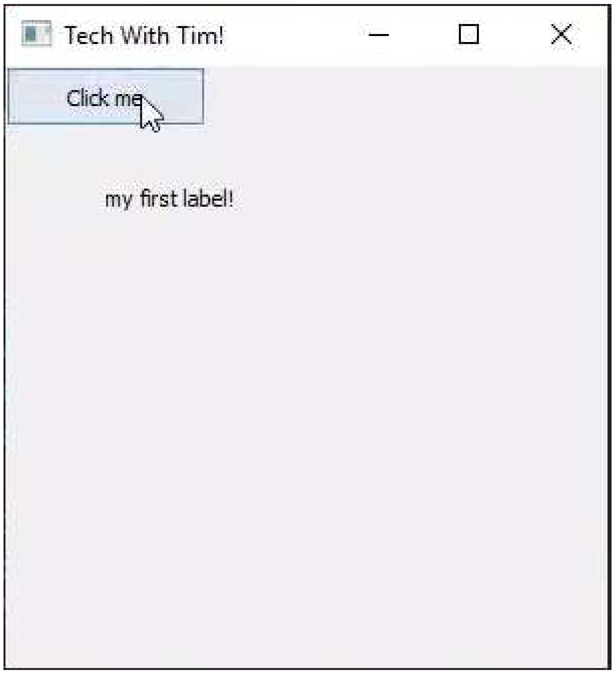
Creating Buttons

Our goal in this tutorial will be to create a button that will change the text of the label in the window.

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

```
    b1 = QtWidgets.QPushButton(win)
    b1.setText("click me")
    #b1.move(100,100) to move the button
```

And now we have a button!



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 pretty 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 to that function. This is done with the following line:

```
    b1.clicked.connect(clicked)  # inside main function
# Note: you don't need the brackets for the function argument (thing in brackets)
```

And now "clicked" will be printed each time we click the button.

OOP Implementation

Up until this point we have created a very basic GUI application inside of a function. This is fine for a small program, but will not scale very well as our GUI becomes more advanced. To fix this we are going to change our current code and write it using OOP (I have some python OOP tutorial here)!

Our code is translated into the following. Notice the use of inheritance. Since we are inherting from *QMainWindow* wherever we previously mentioned *win* we will replace with *self*.

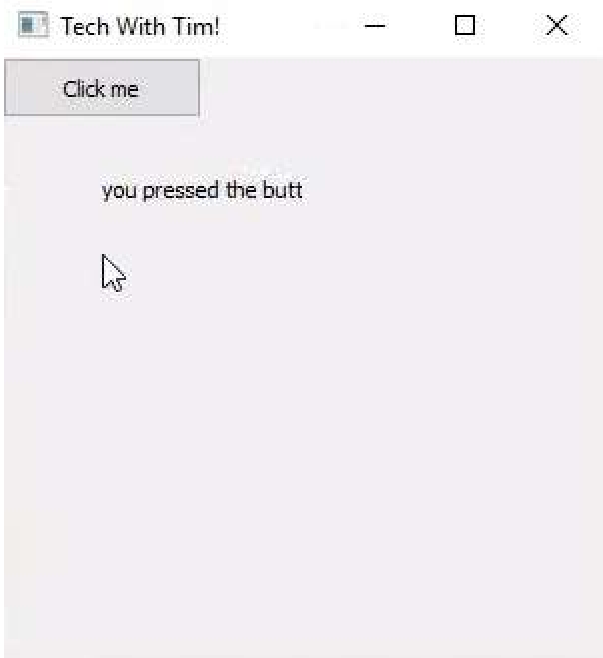
```python
from PyQt5 import QtWidgets
from PyQt5.QtWidgets import QApplication, QMainWindow
import sys


class MyWindow(QMainWindow):
    def __init__(self):
        super(MyWindow,self).__init__()
        self.initUI()

        def button_clicked(self):
            print("clicked")

        def initUI(self):
            self.setGeometry(200, 200, 300, 300)
            self.setWindowTitle("Tech With Tim")

            self.label = QtWidgets.QLabel(self)
            self.label.setText("my first label!")
            self.label.move(50,50)

            self.b1 = QtWidgets.QPushButton(self)
            self.b1.setText("click me!")
            self.b1.clicked.connect(self.button_clicked)



def window():
    app = QApplication(sys.argv)
    win = MyWindow()
    win.show()
    sys.exit(app.exec_())

window()
```

Changing Label Text

Now that we've changed to an OOP approach we can use the method button_clicked() to modify the label when the button is pressed.

```python
    def button_clicked(self):
        print("clicked")
        self.label.setText("you pressed the button")
```

But wait... When we click the button to change the label it cuts off the last few characters.



Luckily we can fix this by updating the labels size property after changing it.

Let's add an update method and call it from the button_clicked() method.

```
class MyWindow(QMainWindow):
    ...
    def clicked_button(self):
        self.label.setText("you pressed the button")
        self.update()

    ...

    def update(self):
        self.label.adjustSize()
```

And now our label resizes properly

Full Code

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


class MyWindow(QMainWindow):
    def __init__(self):
        super(MyWindow,self).__init__()
        self.initUI()

    def button_clicked(self):
        self.label.setText("you pressed the button")
        self.update()

    def initUI(self):
        self.setGeometry(200, 200, 300, 300)
        self.setWindowTitle("Tech With Tim")

        self.label = QtWidgets.QLabel(self)
        self.label.setText("my first label!")
        self.label.move(50,50)

        self.b1 = QtWidgets.QPushButton(self)
        self.b1.setText("click me!")
        self.b1.clicked.connect(self.button_clicked)

    def update(self):
        self.label.adjustSize()


def window():
    app = QApplication(sys.argv)
    win = MyWindow()
    win.show()
    sys.exit(app.exec_())

window()
```