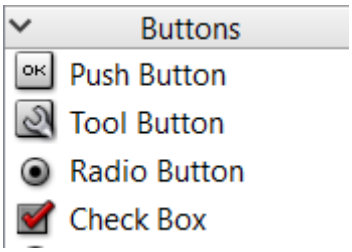
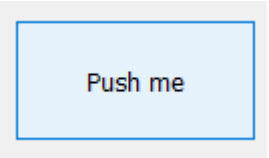
PyQt Dictionary

1. Buttons



Push Button: *QPushButton("text", parent_widget)* more info



The Push Button widget allows the user to press the button by clicking. It includes an animation when the user hovers over the button.

To use this, we will need to connect this button to a function when it is clicked. This means that when we click on the push button, it will cause a function to run.

Note: this function should be inside the Ui_MainWindow class

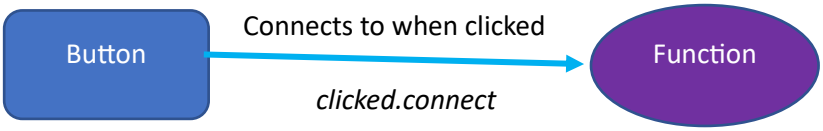
```
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(377, 225)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.text_1 = QtWidgets.QLabel(self.centralwidget)
        self.text_1.setGeometry(QtCore.QRect(20, 20, 231, 21))
        self.text_1.setObjectName("text_1")
        self.button1 = QtWidgets.QPushButton(self.centralwidget)
        self.button1.setGeometry(QtCore.QRect(20, 50, 141, 71))
        self.button1.setObjectName("button1")
```

```
        self.button1.clicked.connect(self.counter)
```

.clicked.connect(function):

When we define the button in setupUi, we use *.clicked.connect(<function>)* to connect our button to a function. When the button is clicked, *<function>* is run.

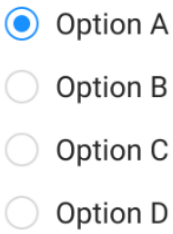
In the line above, we connect *button1* to *self.counter*.




```
def counter(self):
    global count
    count += 1
    self.lcdNumber.display(count)
```

In this example above, the *counter* function will increase the value of the variable *count* by 1. Then it will display this number on an LCD number screen.

Radio Button: *QPushButton("text", parent_widget)* more info



The radio buttons are buttons that you can click on to select an option. If there are multiple buttons in the same widget – the user will only be able to select one – not multiple. It is possible to change this if you'd like.

.setChecked():

This function will change the state of the radio button. It will make the button be selected or not selected. *Radiobutton1.setChecked(True)* will make Radiobutton1 be selected. Similarly (False) would make this be not selected.

.isChecked():

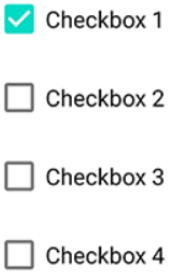
This function checks to see whether a radio button is selected. If, for example, Option A is selected as it is above, *option1.isChecked()* will return **True**.

.setCheckable():

This function decides whether the user is allowed to check this button or not. If we don't want the user to be able to select option A, we would write *optionA.setCheckable(False)*

.text():

This function will return the text associated with the radio button. For example above – *radiobutton1.text()* would return "Option A".



Checkbox: *QCheckBox("text", parent_widget)* more info

Checkboxes can be checked or unchecked by the user. We can check multiple checkboxes at the same time.

.setChecked():

.isChecked():

.setCheckable():

.text():

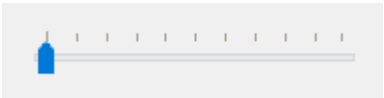
All work as they do for the Radio Button.

.setText():

Changes the label associated with the button, put the text that you'd like in the brackets – if you wanted to display "Option A" next to checkbox1, we'd need to write *checkbox1.setText("Option A")*

2. Inputs

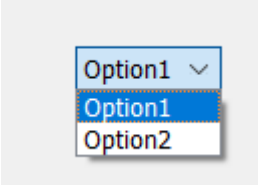
Sliders: *QSlider(Qt.Horizontal)* more info



Sliders can be used for user input too. The user can drag the slider across to change something incrementally.

If we want a vertical slider instead, we use Qt.Vertical instead of Qt.Horizontal.

Combobox: *QComboBox()* more info



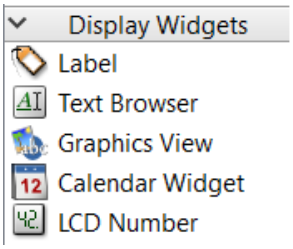
This is a dropdown box which allows the user to select one of the options.

addItems():

If we want to give the user more than one option, we can use: *addItems("Option1", "Option2")* and we'll get the options displayed in to combobox on the right.

You are welcome to try out any other input widgets that you think would be useful for your GUI. Information for most of the widgets can be found here: PyQt - Basic Widgets (tutorialspoint.com)

3. Displays



LCD Number: *QLCDNumber*

We only really need to know about this one and Label.

LCD Number is simply used to display a number to the user. We can control which number is displayed.

.display(number):

If we want LCD1 to display the number 7, we simply write *LCD1.display(8)*