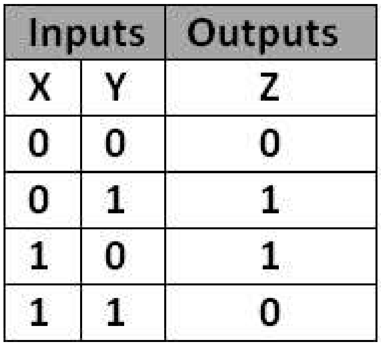
Creating The GUI

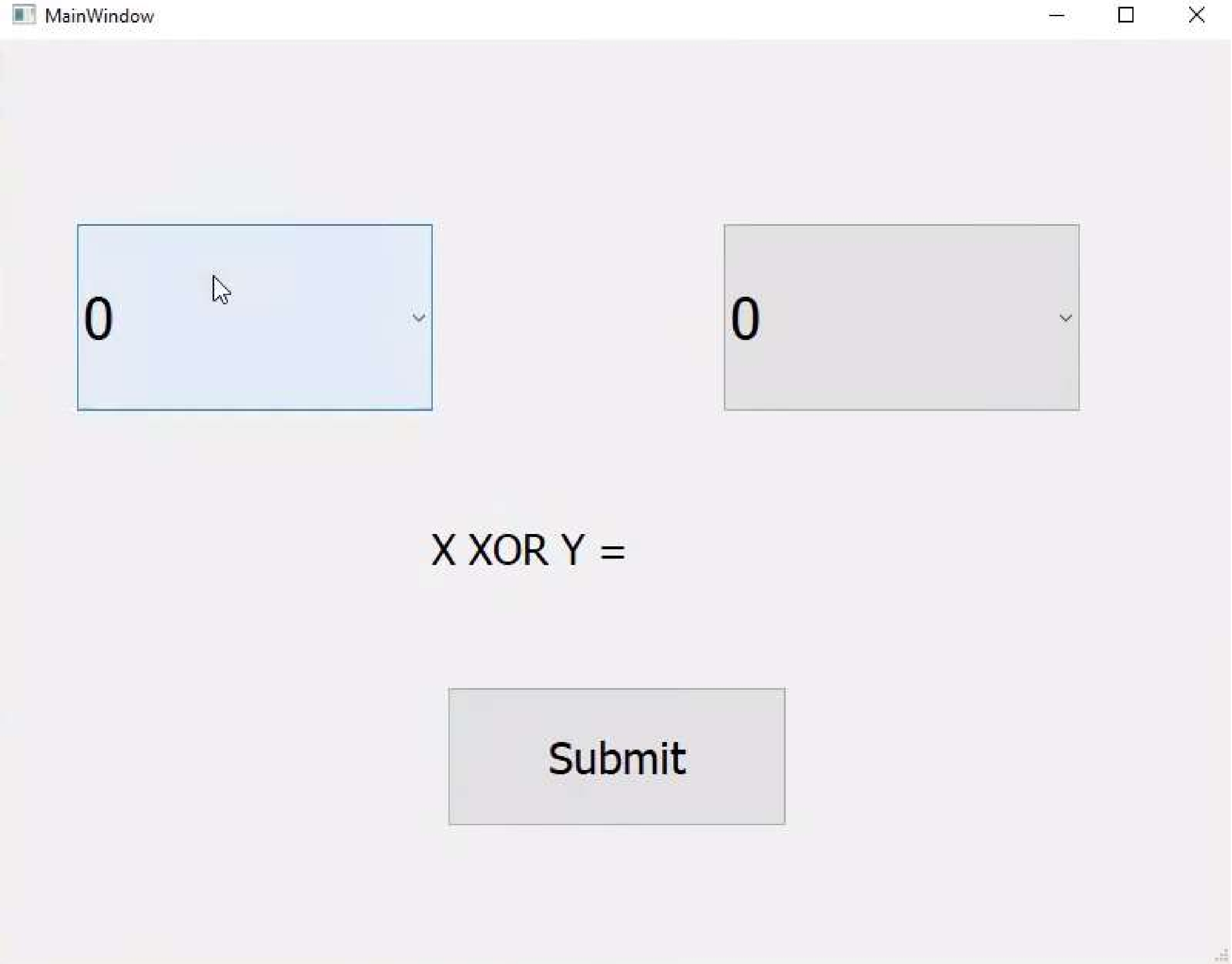
For this tutorial we will create a GUI that can simulate the **XOR** function. We will use comboboxes to allow the user to select two binary inputs (0 or 1) and then display the result of the XOR function applied to them.

Note: This is the truth table for the XOR function. It takes two binary inputs which we will get from our comboboxes and gives us an output of 0 or 1.



Inputs		Outputs
X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

This is the GUI we will be building:



0

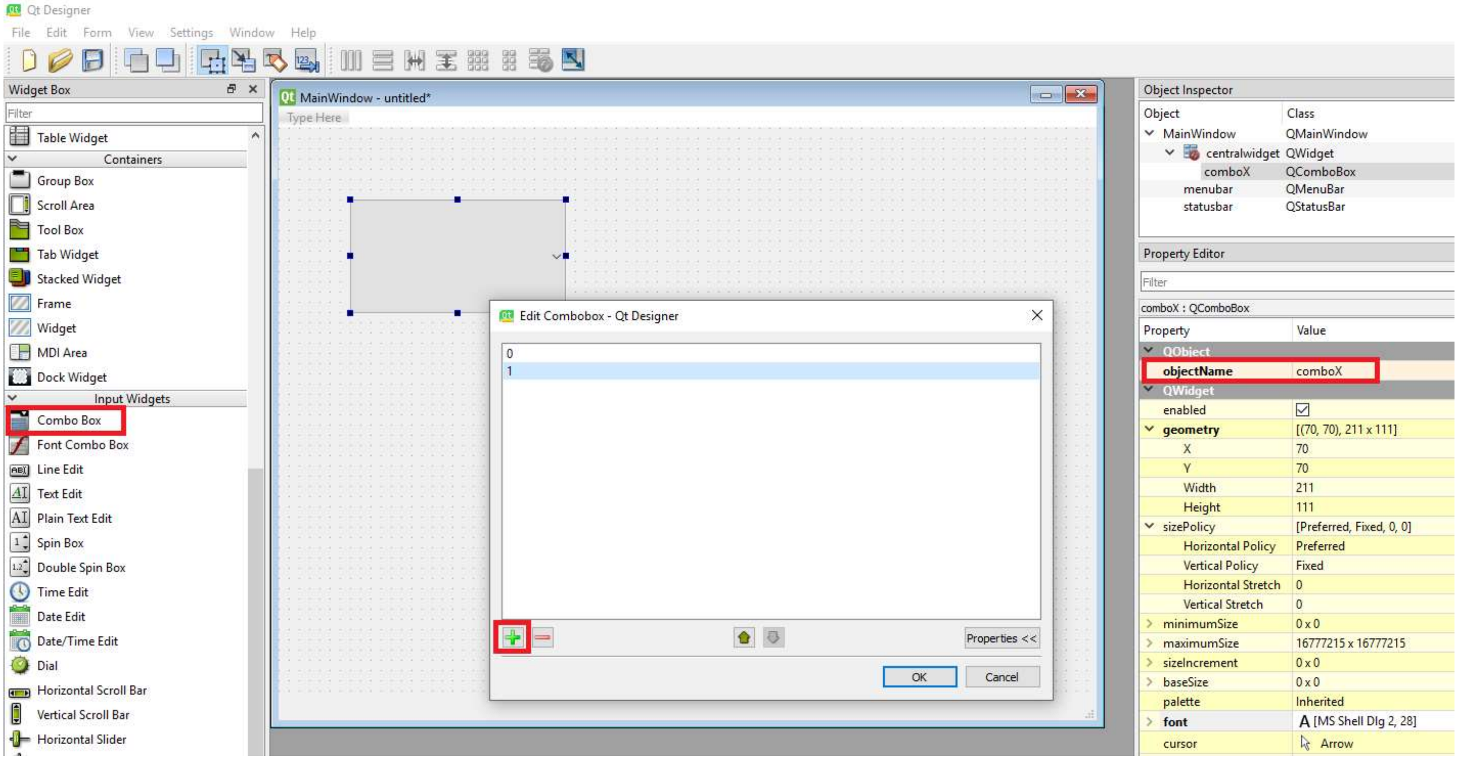
0

X XOR Y =

Adding ComboBoxes

To add a combobox to your GUI simply drag it in from the side bar. Once we have moved it to our desired location, resized it and named it we can add some items to it.

To add items to the combobox **double click it** and hit the green "+" button



Object	Class
MainWindow	QMainWindow
centralwidget	QWidget
comboX	QComboBox
menubar	QMenuBar
statusbar	QStatusBar

comboX : QComboBox

Property	Value
QObject	
objectName	comboX
QWidget	
enabled	☑
geometry	[(70, 70), 211 x 111]
X	70
Y	70
Width	211
Height	111
sizePolicy	[Preferred, Fixed, 0, 0]
Horizontal Policy	Preferred
Vertical Policy	Fixed
Horizontal Stretch	0
Vertical Stretch	0
minimumSize	0 x 0
maximumSize	16777215 x 16777215
sizeIncrement	0 x 0
baseSize	0 x 0
palette	Inherited
font	A [MS Shell Dlg 2, 28]
cursor	Arrow

I'll leave the rest of the GUI to you as I am sure by now you know how to build it.

Adding Items

In some cases you may want to add items to your combobox from code. You can do this using the **.addItem()** method on your desired combobox. In this example my combobox was named comboX.

```python
self.comboX.addItem("This is a new item")
```

Changing Default Item

Another useful thing you may want to do is the set the item that shows by default in the combobox. To do this we can find the index of that item and set the current index of the combobox. The indexes work the same a python lists. Where 0 is the first item and the amount of items -1 is the last item.

```python
index = self.comboX.findText("1", QtCore.QtMatchFixedString)  # finds the index of the item you want
self.comboX.setCurrentIndex(index)  # sets the index (you can also just put an integer for the variable index)
```

Getting Combobox Selection

The next thing we probably want to do is get the item that the user selected in the comboBox. For this example we are going to get the input from each combobox. Apply the XOR function to it and change a label to show the result. This means the first thing we need to do is create a new method and link it to our submit button.

```
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        ...
        self.submit.clicked.connect(self.pressed)

    def pressed(self):
        pass
```

Now from inside or pressed method we can get the comboBox text.

```
        x = int(self.comboX.currentText())
        y = int(self.comboY.currentText())
```

And now to get the result of the XOR function we will use a clever combinations of operators.

```
        xor = (x and not y) or (not x and y)
```

Finally, we can set the label text to show this result.

```
        self.label.setText("X XOR Y = " + str(xor))
```

Full Code

```
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'test.ui'
#
# Created by: PyQt5 UI code generator 5.11.3
#
# WARNING! All changes made in this file will be lost!

from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(800, 600)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.comboX = QtWidgets.QComboBox(self.centralwidget)
        self.comboX.setGeometry(QtCore.QRect(50, 120, 231, 121))
        font = QtGui.QFont()
        font.setPointSize(28)
        self.comboX.setFont(font)
        self.comboX.setObjectName("comboX")
        self.comboX.addItem("")
        self.comboX.addItem("")
        self.comboY = QtWidgets.QComboBox(self.centralwidget)
        self.comboY.setGeometry(QtCore.QRect(470, 120, 231, 121))
        font = QtGui.QFont()
        font.setPointSize(28)
        self.comboY.setFont(font)
        self.comboY.setObjectName("comboY")
        self.comboY.addItem("")
        self.comboY.addItem("")
        self.submit = QtWidgets.QPushButton(self.centralwidget)
        self.submit.setGeometry(QtCore.QRect(290, 420, 221, 91))
        font = QtGui.QFont()
        font.setPointSize(22)
        self.submit.setFont(font)
        self.submit.setObjectName("submit")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(280, 290, 221, 81))
        font = QtGui.QFont()
        font.setPointSize(20)
        self.label.setFont(font)
        self.label.setObjectName("label")
        MainWindow.setCentralWidget(self.centralwidget)
        self.menubar = QtWidgets.QMenuBar(MainWindow)
        self.menubar.setGeometry(QtCore.QRect(0, 0, 800, 21))
        self.menubar.setObjectName("menubar")
        MainWindow.setMenuBar(self.menubar)
        self.statusbar = QtWidgets.QStatusBar(MainWindow)
        self.statusbar.setObjectName("statusbar")
        MainWindow.setStatusBar(self.statusbar)

        self.submit.clicked.connect(self.pressed)

        self.retranslateUi(MainWindow)
        QtCore.QMetaObject.connectSlotsByName(MainWindow)

    def retranslateUi(self, MainWindow):
        _translate = QtCore.QCoreApplication.translate
        MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
        self.comboX.setItemText(0, _translate("MainWindow", "0"))
        self.comboX.setItemText(1, _translate("MainWindow", "1"))
        self.comboY.setItemText(0, _translate("MainWindow", "0"))
        self.comboY.setItemText(1, _translate("MainWindow", "1"))
        self.submit.setText(_translate("MainWindow", "Submit"))
        self.label.setText(_translate("MainWindow", "X XOR Y ="))

    def pressed(self):
        x = int(self.comboX.currentText())
        y = int(self.comboY.currentText())
        xor = (x and not y) or (not x and y)
        if xor == True:
            xor = 1
        else:
            xor = 0

        self.label.setText("X XOR Y =  " + str(xor))



if __name__ == "__main__":
    import sys
    app = QtWidgets.QApplication(sys.argv)
    MainWindow = QtWidgets.QMainWindow()
    ui = Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```