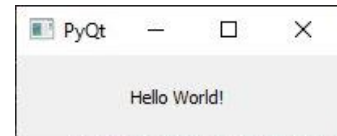


# CIV 2802 – Sistemas Gráficos para Engenharia – PUC-Rio

## Programa muito simples “Hello World” em Python usando Qt 5

### Arquivo LW1\_HelloWorld/simple/main.py

```
1 import sys
2 from PyQt5.QtCore import *
3 from PyQt5.QtWidgets import *
4
5 class AppWindow(QMainWindow):
6     def __init__(self):
7         # Window initialization
8         super(AppWindow, self).__init__()
9         self.setGeometry(100, 100, 200, 50)
10        self.setWindowTitle("PyQt")
11        self.label = QLabel("Hello World!")
12        self.label.setAlignment(Qt.AlignCenter)
13        self.setCentralWidget(self.label)
14
15 def main():
16     app = QApplication(sys.argv)
17     w = AppWindow()
18     w.show()
19     sys.exit(app.exec_())
20
21 if __name__ == "__main__":
22     main()
23
24
```



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

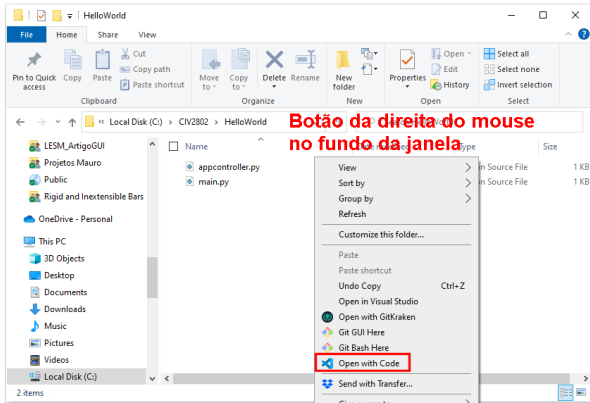
```
class AppWindow(QMainWindow):
    def __init__(self):
        # Window initialization
        super(AppWindow, self).__init__()
        self.setGeometry(100, 100, 200, 50)
        self.setWindowTitle("PyQt")
        self.label = QLabel("Hello World!")
        self.label.setAlignment(Qt.AlignCenter)
        self.setCentralWidget(self.label)
```

```
def main():
    app = QApplication(sys.argv)
    w = AppWindow()
    w.show()
    sys.exit(app.exec_())
```

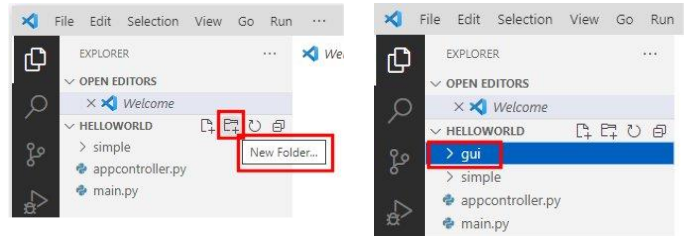
```
if __name__ == "__main__":
    main()
```

# Roteiro para criação de um programa simples "Hello World" em Python usando Qt 5, com criação de interface pelo Qt Designer

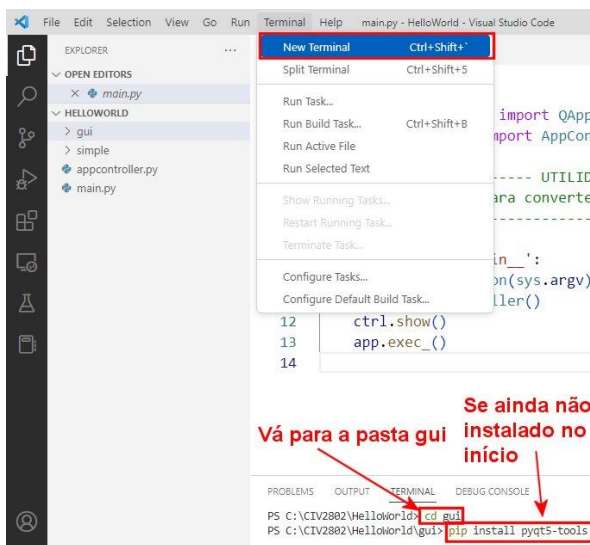
## 1. Abra o Visual Studio Code



## 2. Crie uma pasta "gui"

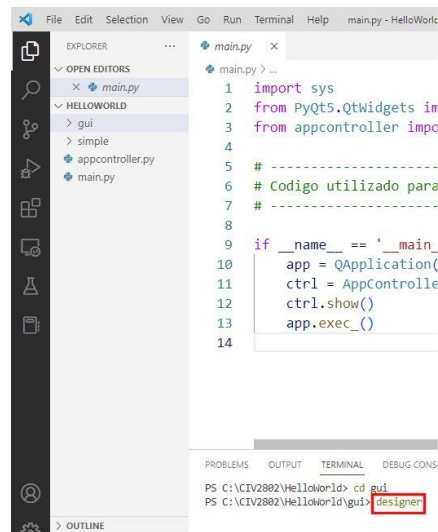


## 3. Abra um Terminal no Visual Studio Code e vá para a pasta "gui". Instale o pacote pyqt5-tools

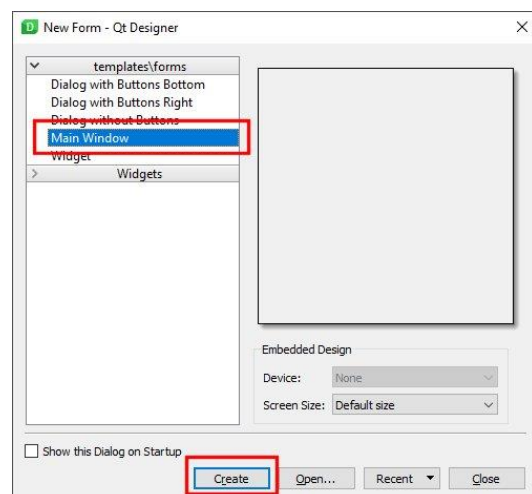
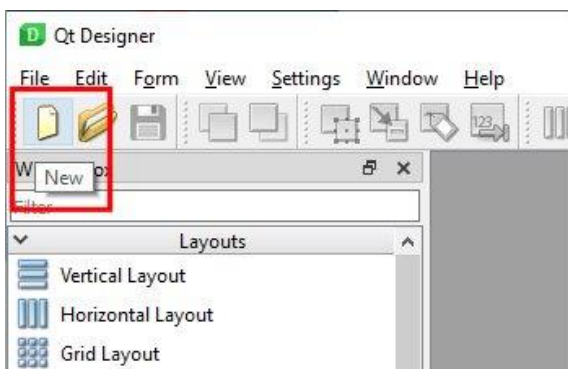


## 4. Abra o aplicativo Qt Designer

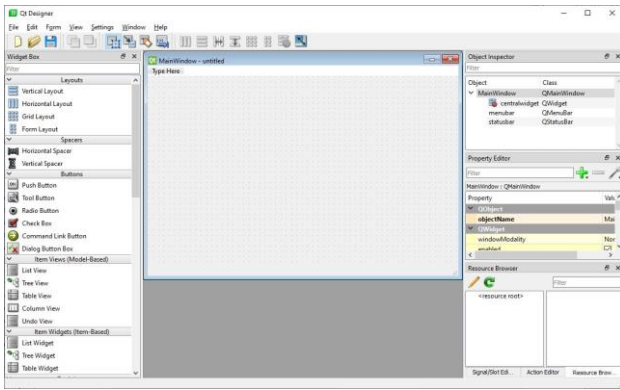
Antes adicione essa pasta no "path" do usuário:  
C:\Users\<user>\AppData\Local\Programs\Python\Python39\Lib\site-packages\qt5\_applications\Qt\bin



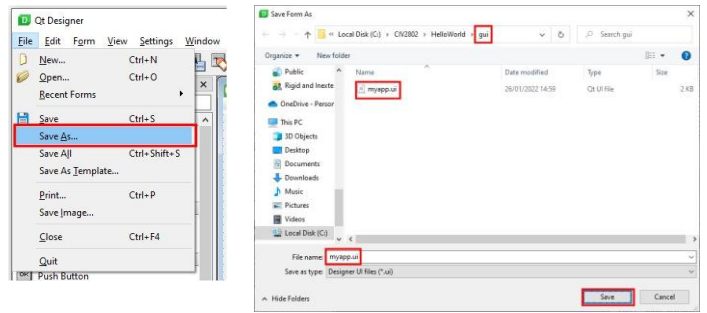
## 5. Execute o programa Qt Designer e crie uma nova Qt Main Window



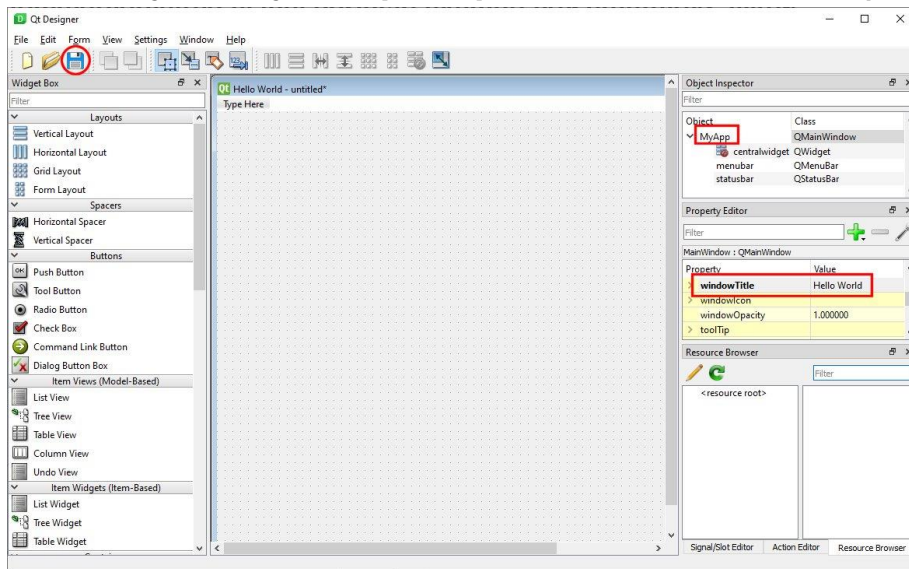
## 6. Pronto para criar interface gráfica



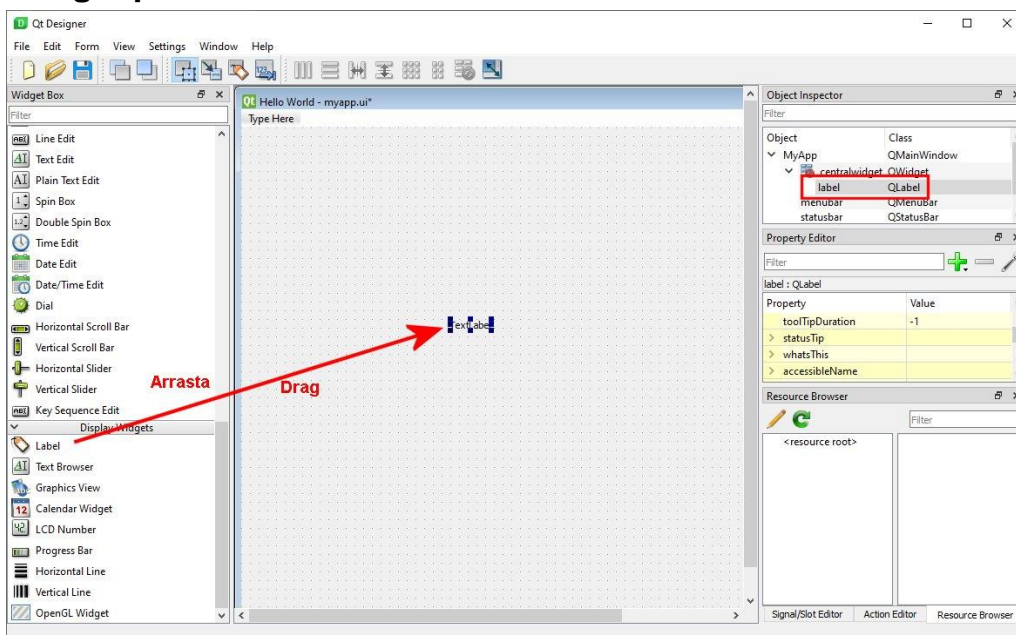
## 7. Salve arquivo com nome myapp.ui



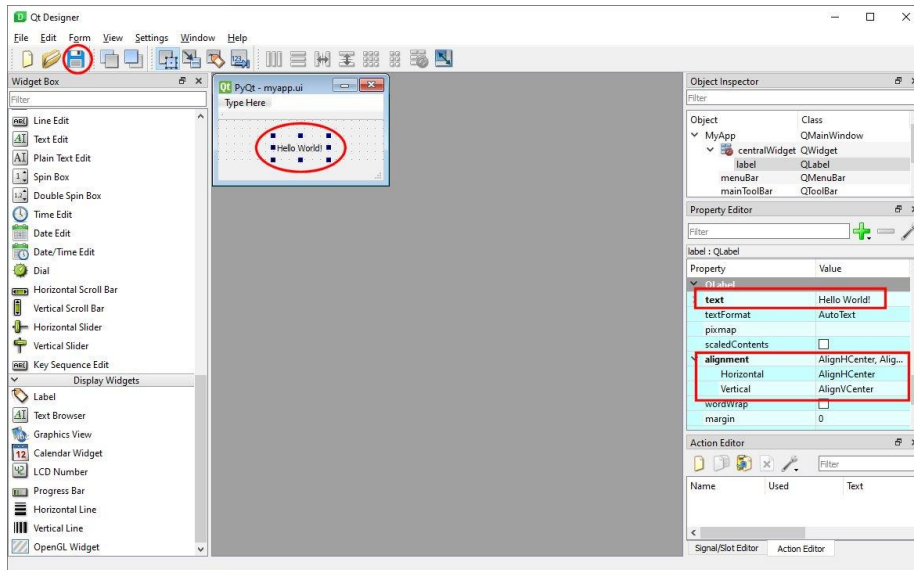
## 8. Muda o nome do objeto da janela principal e muda o seu título (salve)



## 9. Crie um widget para o labels



## 10. Modifique o tamanho da janela, edite o label e force alinhamento centrado na horizontal e vertical do texto do label (salve com o nome "myapp.ui")



## 11. Crie o arquivo myapp.py

```
cd gui
pyuic5 myapp.ui -o myapp.py
```

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

# Form implementation generated from reading ui file 'myapp.ui'
#
# Created by: PyQt5 UI code generator 5.15.6
#
# WARNING: Any manual changes made to this file will be lost when pyuic5 is
# run again. Do not edit this file unless you know what you are doing.
```

```
from PyQt5 import QtCore, QtGui, QtWidgets
```

```
class Ui_MyApp(object):
    def setupUi(self, MyApp):
        MyApp.setObjectName("MyApp")
        MyApp.resize(211, 110)
        self.centralWidget = QtWidgets.QWidget(MyApp)
        self.centralWidget.setObjectName("centralWidget")
        self.label = QtWidgets.QLabel(self.centralWidget)
        self.label.setGeometry(QtCore.QRect(70, 20, 71, 31))
        self.label.setAlignment(QtCore.Qt.AlignCenter)
        self.label.setObjectName("label")
        MyApp.setCentralWidget(self.centralWidget)
        self.menuBar = QtWidgets.QMenuBar(MyApp)
        self.menuBar.setGeometry(QtCore.QRect(0, 0, 211, 21))
        self.menuBar.setObjectName("menuBar")
        MyApp.setMenuBar(self.menuBar)
        self.mainToolBar = QtWidgets.QToolBar(MyApp)
        self.mainToolBar.setObjectName("mainToolBar")
        MyApp.addToolBar(QtCore.Qt.TopToolBarArea, self.mainToolBar)
        self.statusBar = QtWidgets.QStatusBar(MyApp)
        self.statusBar.setObjectName("statusBar")
        MyApp.setStatusBar(self.statusBar)

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

    def retranslateUi(self, MyApp):
        translate = QtCore.QCoreApplication.translate
        MyApp.setWindowTitle(translate("MyApp", "PyQt"))
        self.label.setText(translate("MyApp", "Hello World!"))
```

(não modifique esse arquivo)

## 12. Arquivo appcontroller.py (classe AppController herda de QMainWindow e Ui\_MyApp)

```
from PyQt5.QtWidgets import QMainWindow
from gui.myapp import Ui_MyApp
```

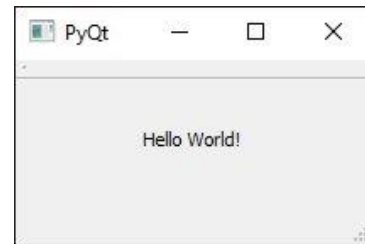
```
class AppController(QMainWindow, Ui_MyApp):
    def __init__(self):
        super().__init__()
        super().setupUi(self)
```

## 13. Arquivo main.py

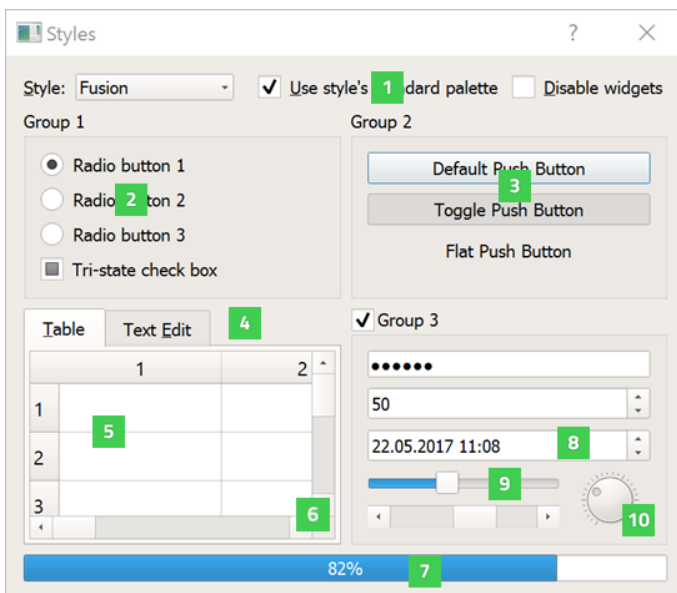
```
import sys
from PyQt5.QtWidgets import QApplication
from appcontroller import AppController

if __name__ == '__main__':
    app = QApplication(sys.argv)
    ctrl = AppController()
    ctrl.show()
    app.exec_()
```

## 14. Execução



## QWidget



- › `QCheckBox` (1) provides a checkbox with a text label.
- › `QRadioButton` (2) provides a radio button with a text or pixmap label.
- › `QPushButton` (3) provides a command button.
- › `QTabWidget` (4) provides a stack of tabbed widgets.
- › `QTableWidget` (5) provides a classic item-based table view.
- › `QScrollBar` (6) provides a vertical or horizontal scroll bar.
- › `QProgressBar` (7) provides a horizontal progress bar.
- › `QDateTimeEdit` (8) provides a widget for editing dates and times.
- › `QSlider` (9) provides a vertical or horizontal slider.
- › `QDial` (10) provides a rounded range control (like a speedometer or potentiometer).

## QMainWindow

