PyQt Day 1

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/41afcf8e-e3f8-42f2-977c-d6b6e19537f6/pyqt5tutorial.pdf

DB연결

```
!pip install pymysql
!pip install PyQt5

import pymysql
import requests
import pprint

conn = pymysql.connect(host='tlagusals.cafe24.com', user='tlagusals', password='Newnew11@', db='tlagusals', charset='utf8')

cur = conn.cursor()

cur.execute("SELECT * FROM user")
result = cur.fetchall()

pprint.pprint(result)
```

```
((b'0224ceo', b'ceo', b'0000', b'ceo@ceo.com', b'ceo'),
 (b'0224user', b'user', b'0000', b'user@user.com', b'user'),
 (b'admin',
  b'admin',
  b'1111',
  b'hyunmin04050@gmail.com',
  b'\xea\xb4\x80\xeb\xa6\xac\xec\x9e\x90'),
 (b'bright',
  b'user',
  b'1234',
  b'bright@naver.com',
  b'\xea\xb9\x80\xed\x98\x95\xec\xa4\x80'),
 (b'ceo', b'ceo', b'1234', b'ceo@naver.com', b'ceo'),
 (b'gusals',
  b'ceo',
  b'1111',
  b'hyunmin04050@gmail.com',
  b'\xed\x98\x84\xeb\xaf\xbc'),
 (b'hyunmin',
  b'user',
  b'1111',
  b'hyunmin04050@gmail.com',
  b'\xec\x8b\xac\xed\x98\x84\xeb\xaf\xbc'),
 (b'ksb', b'user', b'ksb', b'ksb@ksb', b'\xea\xb0\x95\xec\x84\xb8\xeb\xb9\x88'),
 (b't1', b'user', b't1', b't1@t1', b't11'))
```

PyQt ?

파이썬 기반의 GUI 라이브러리

TKinter

- 파이썬 기본 GUI
- 기본 설치
- 디자인이 떨어진다

wxPython

- ?

PyQt

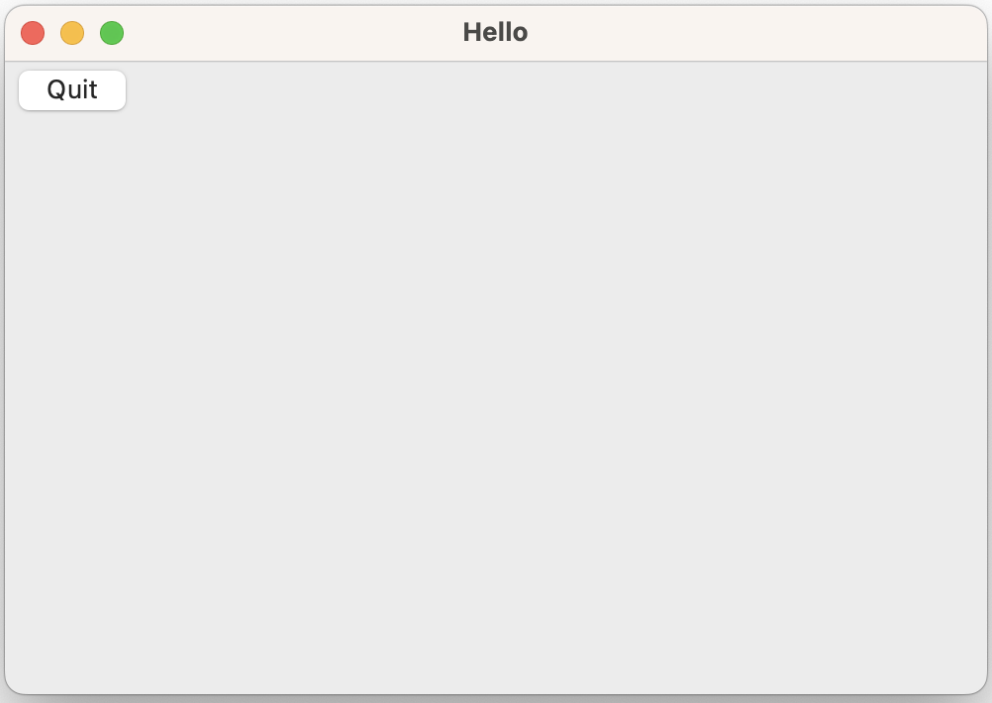
- Anacondar 기본 설치
- 디자인이 예쁘다
- riverbankcomputing 사
- C언어용으로 개발되었으나 파이썬용으로 전환
- 현재 버전 6 출시

윈도우 창 띄우기 (수동)

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget # 기본적인 창 구성 요소 제공 PyQt5.QtWidgets

class MyApp(QWidget):
    def __init__(self): # 생성자 설정
        super().__init__()
        self.initUI()

    def initUI(self): # 창 기본 설정 > 제목 설정
        self.setWindowTitle("Hello World!!!")
        self.show()

app = QApplication(sys.argv) # 파이썬은 인터프리터라서 코드 한줄 한줄을 실행 후 종료한다. 다음은 열린 파일 유지 시키는 코드 3줄
ex = MyApp()
sys.exit(app.exec_())
```

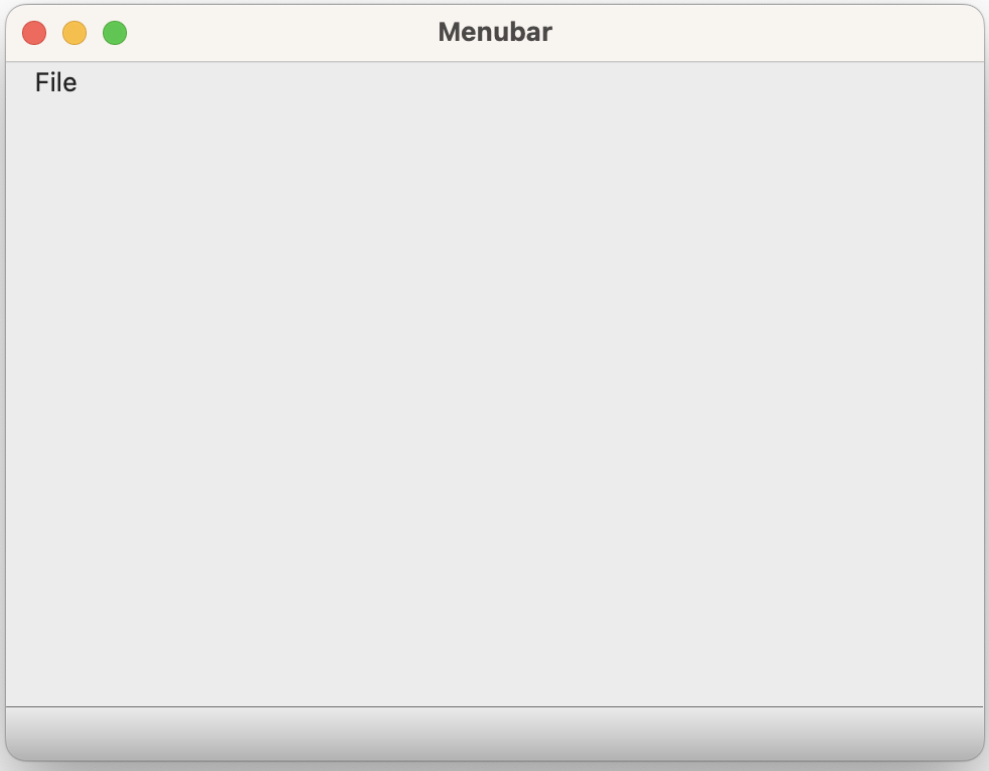
윈도우 창 띄우기 (수동) 2

```
# 윈도우 창  띄우기 (수동) 2
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton
from PyQt5.QtCore import QCoreApplication

class MyApp(QWidget):
    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        self.setWindowTitle("Hello")
        btn = QPushButton('Quit', self)
        btn.clicked.connect(QCoreApplication.instance().quit)
        self.show()

app = QApplication(sys.argv)
ex = MyApp()
sys.exit(app.exec_())
```



윈도우 메뉴 생성 (수동)

```
# 윈도우 메뉴 생성 (수동)
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow, QAction, qApp
from PyQt5.QtGui import QIcon

class MyApp(QMainWindow):

    def __init__(self):
        super().__init__()
        self.initUI()

    def initUI(self):
        exitAction = QAction(QIcon('exit.png'), 'Exit', self)
        exitAction.setShortcut('Ctrl+Q')
        exitAction.setStatusTip('Exit application')
        exitAction.triggered.connect(qApp.quit)

        self.statusBar()

        menubar = self.menuBar()
        menubar.setNativeMenuBar(False)
        filemenu = menubar.addMenu('&File')
        filemenu.addAction(exitAction)

        self.setWindowTitle('Menubar')
        self.setGeometry(300, 300, 300, 200)
        self.show()

if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = MyApp()
    sys.exit(app.exec_())
```

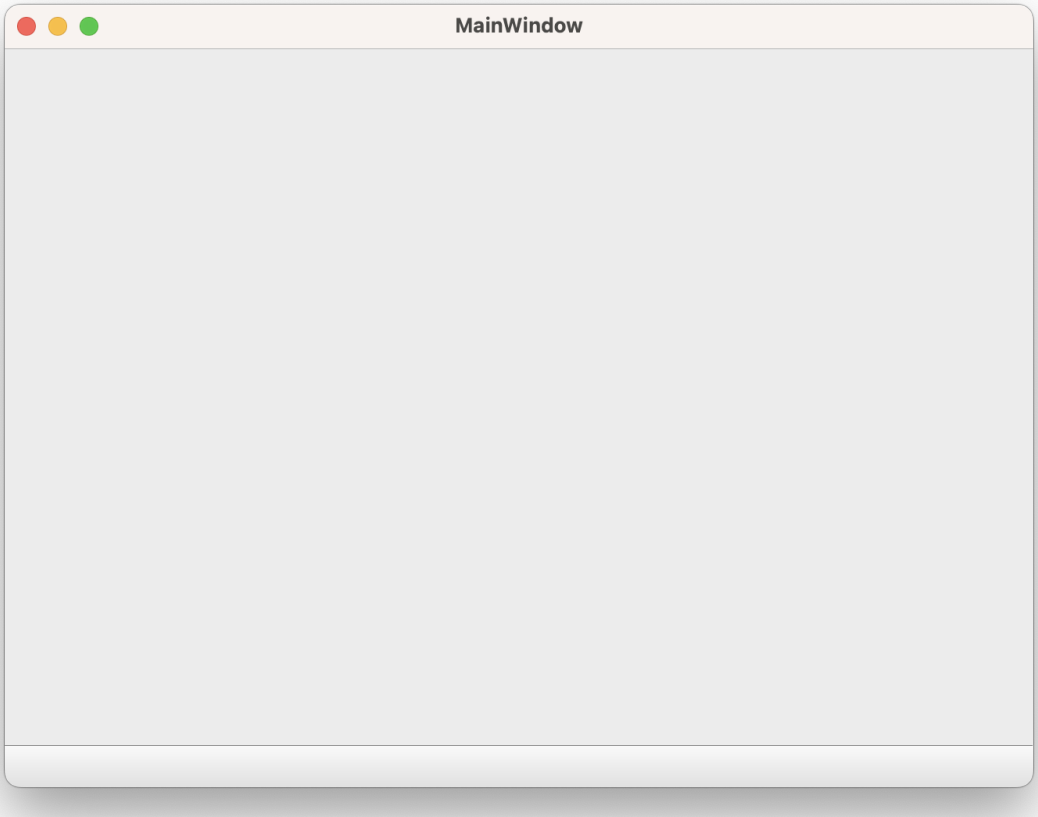


Designer

UI Connect

Qt Designer로 생성한 윈도우 여는 방법

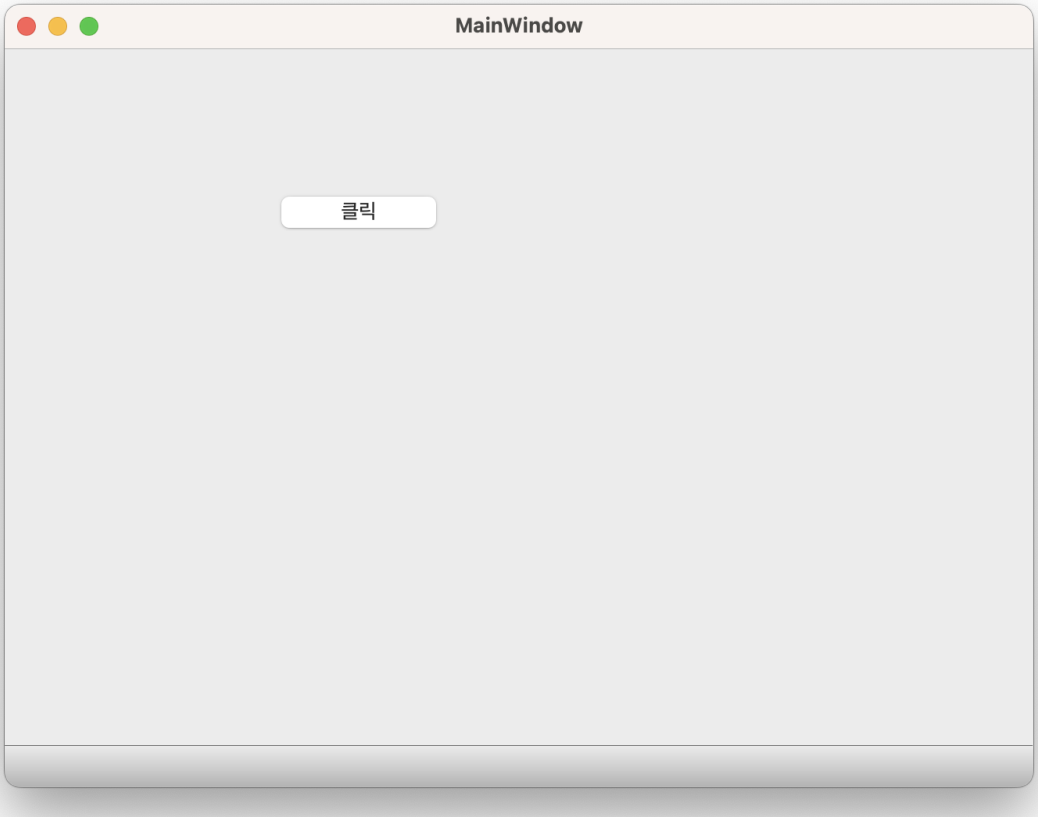
```
# Qt Designer로 생성한 윈도우 여는 방법
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType('mywindow.ui')[0]

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

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```



PushButton

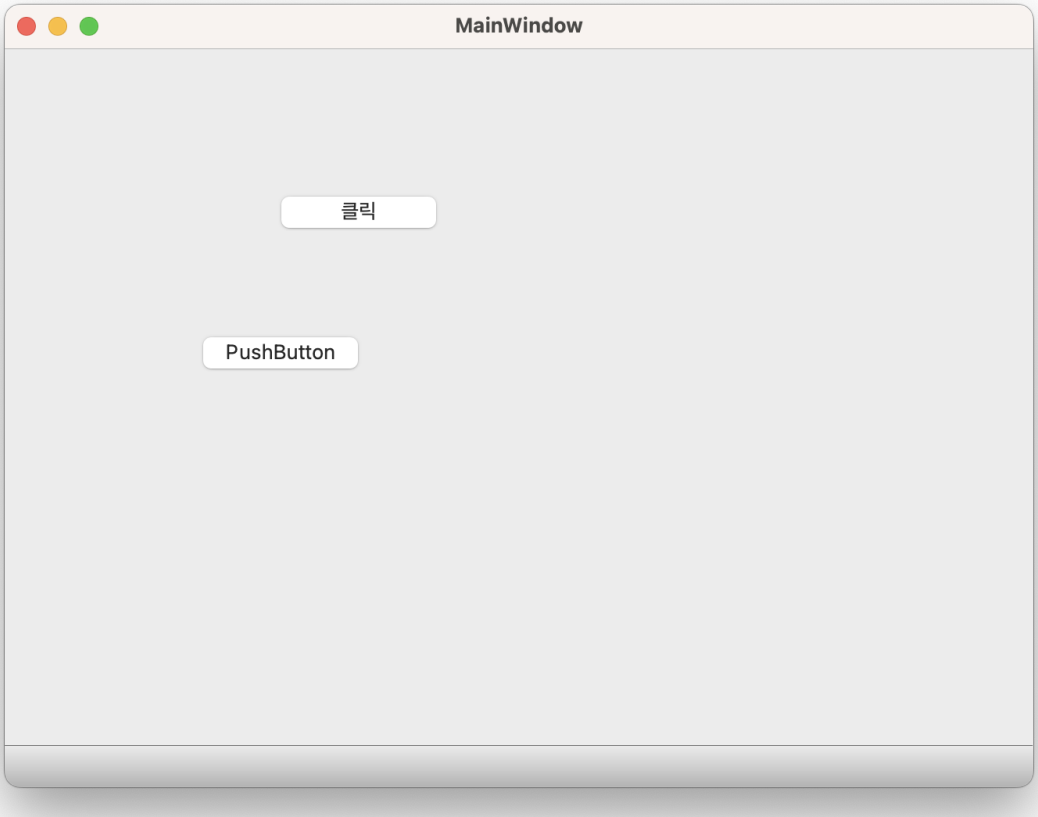
```
# 버튼 1개 인식
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType('mywindow1.ui')[0]

class MyWindow(QMainWindow, form_class):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.pushButton.clicked.connect(self.btn_clicked)

    def btn_clicked(self):
        print('클릭 확인')

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```

```
클릭 확인
```




```
# 버튼 2개 인식
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType('mywindow2.ui')[0]

class MyWindow(QMainWindow, form_class):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.pushButton_1.clicked.connect(self.btn1_clicked)
        self.pushButton_2.clicked.connect(self.btn2_clicked)

    def btn1_clicked(self):
        print('1 클릭 확인')

    def btn2_clicked(self):
        print('2 클릭 확인')

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```

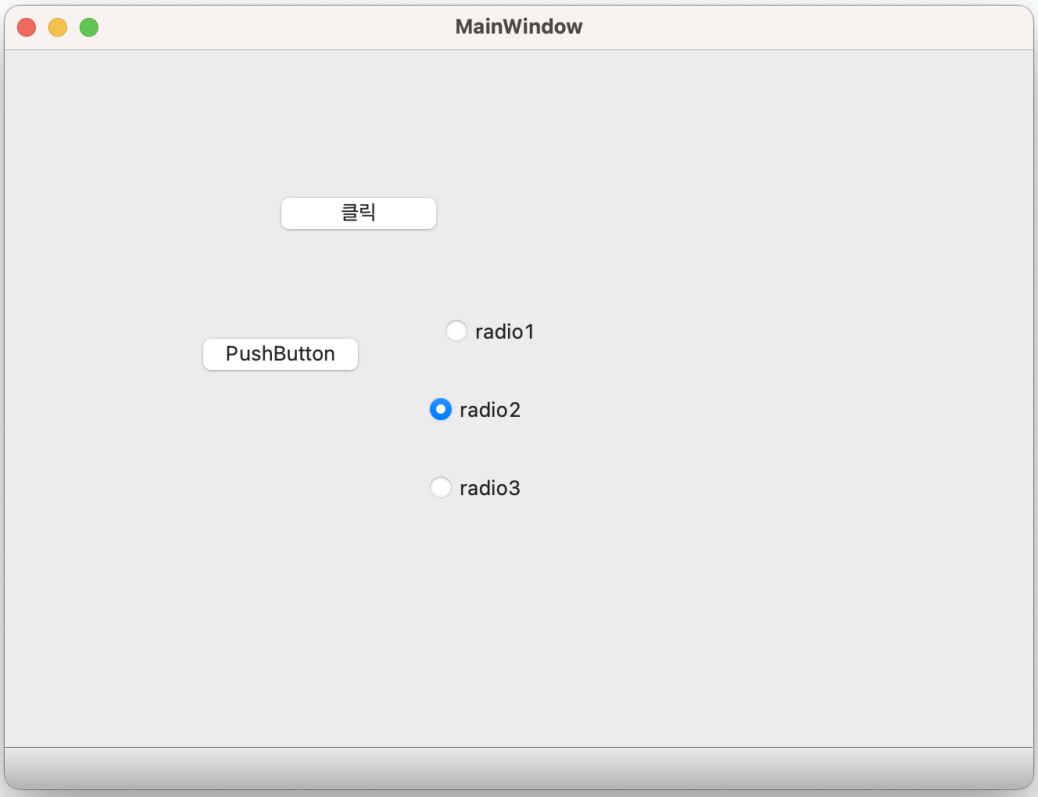
```
2 클릭 확인
1 클릭 확인
1 클릭 확인
2 클릭 확인
```



RadioButton

```
# radio 버튼 인식
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType('mywindow3.ui')[0]

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

        self.radioButton_1.clicked.connect(self.radioFunction)
        self.radioButton_2.clicked.connect(self.radioFunction)
        self.radioButton_3.clicked.connect(self.radioFunction)

    def radioFunction(self): # 한 개만 선택 가능
        if self.radioButton_1.isChecked():
            print('radio button 1 checked')
        elif self.radioButton_2.isChecked():
            print('radio button 2 checked')
        elif self.radioButton_3.isChecked():
            print('radio button 3 checked')

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```

```
radio button 1 checked
radio button 3 checked
radio button 2 checked
```

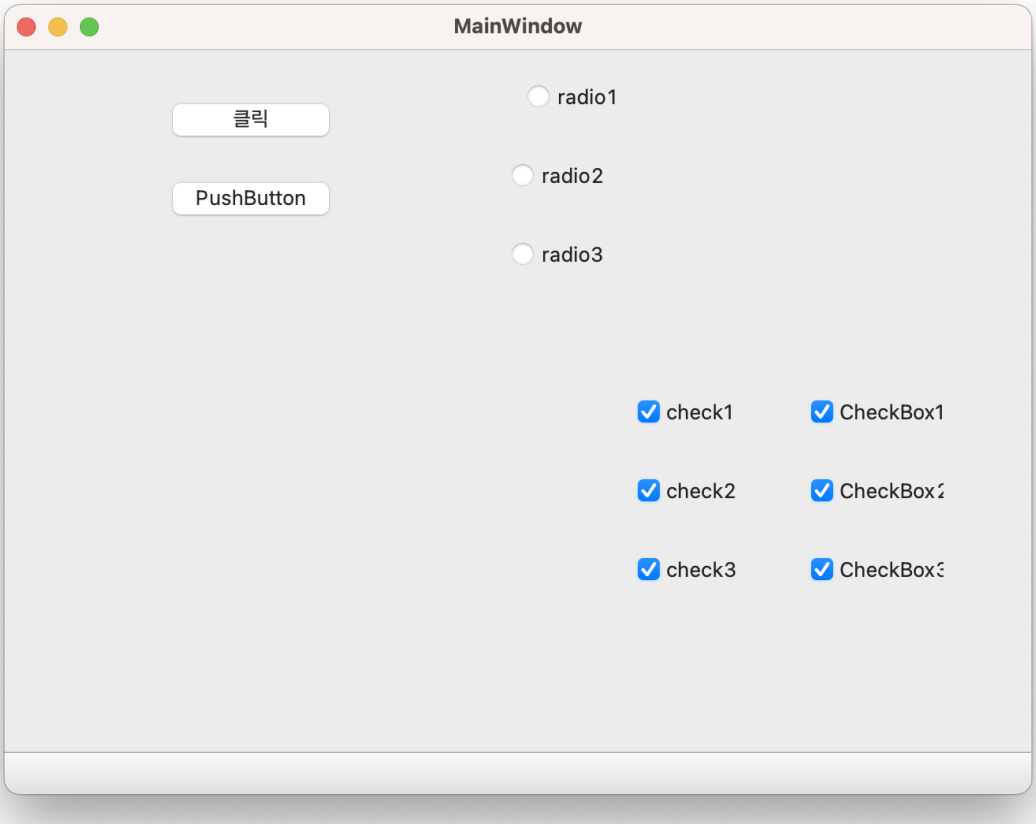


CheckBox

```
# checkbox 인식
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType('mywindow4.ui')[0]

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

        self.check1.stateChanged.connect(self.chkFunction)
        self.check2.stateChanged.connect(self.chkFunction)
        self.check3.stateChanged.connect(self.chkFunction)

        self.checkBox_1.stateChanged.connect(self.chkBoxFunction)
        self.checkBox_2.stateChanged.connect(self.chkBoxFunction)
        self.checkBox_3.stateChanged.connect(self.chkBoxFunction)

    def chkFunction(self): # 여러개 선택 가능
        if self.check1.isChecked():
            print('check1 checked')
        if self.check2.isChecked():
```

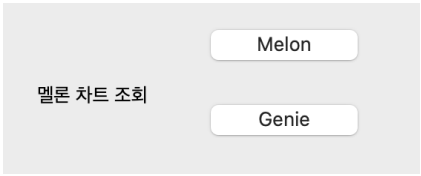
```
            print('check2 checked')
        if self.check3.isChecked():
            print('check3 checked')
        print()

    def chkBoxFunction(self):
        if self.checkBox_1.isChecked():
            print('checkBox_1 checked')
        if self.checkBox_2.isChecked():
            print('checkBox_2 checked')
        if self.checkBox_3.isChecked():
            print('checkBox_3 checked')
        print()

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```

```
checkBox_1 checked

check3 checked

checkBox_1 checked
checkBox_3 checked

check2 checked
check3 checked

check1 checked
check2 checked
check3 checked

checkBox_1 checked
checkBox_2 checked
checkBox_3 checked
```

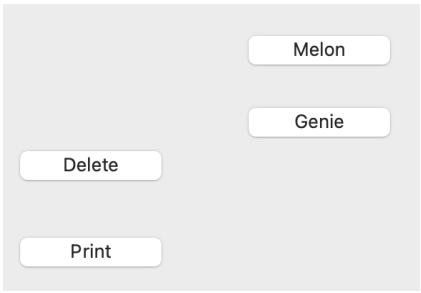


Label

```
# Label
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType('mywindow5.ui')[0]

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

        self.MelonButton.clicked.connect(self.melonFunction)
        self.GenieButton.clicked.connect(self.genieFunction)

    def melonFunction(self):
        self.label.setText('멜론 차트 조회')

    def genieFunction(self):
        self.label.setText('지니 차트 조회')

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```



```
#
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType('mywindow5.ui')[0]

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

        self.MelonButton.clicked.connect(self.melonFunction)
        self.GenieButton.clicked.connect(self.genieFunction)
        self.deleteButton.clicked.connect(self.deleteFunction)
        self.printButton.clicked.connect(self.printFunction)

    def melonFunction(self):
        self.label.setText('멜론 차트 조회')

    def genieFunction(self):
        self.label.setText('지니 차트 조회')

    def deleteFunction(self):
        self.label.clear()

    def printFunction(self):
        print(self.label.text())

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```

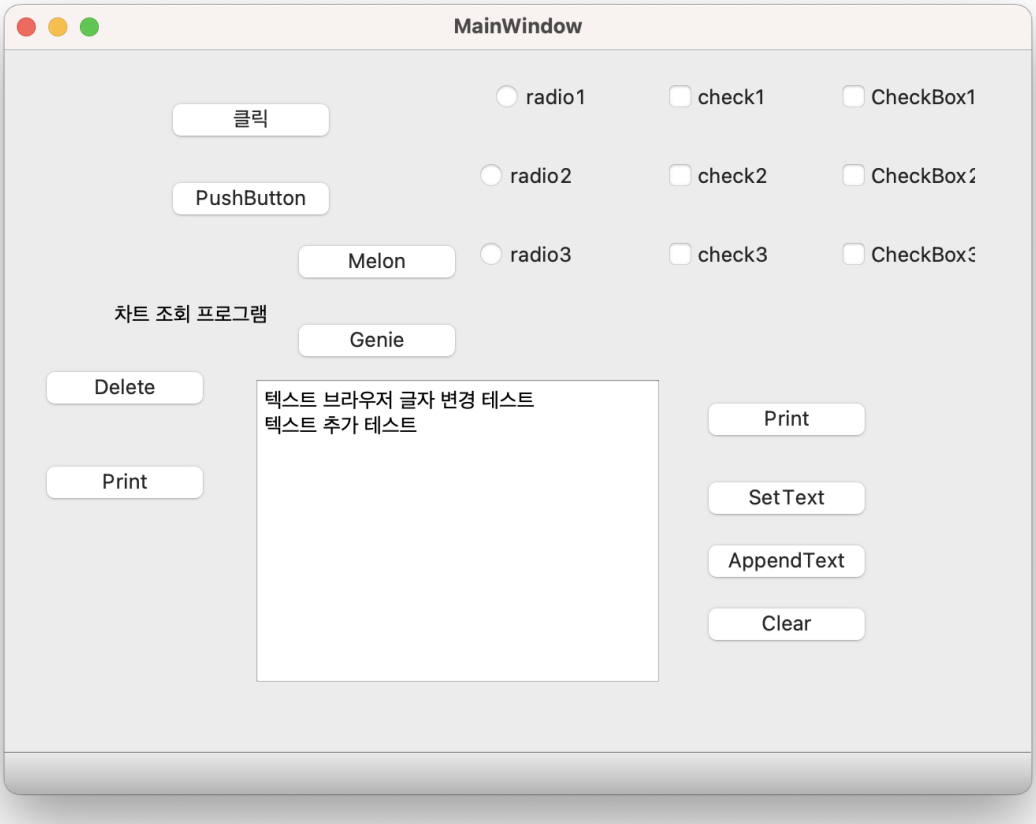



```
지니 차트 조회
```

TextBrowser

- 라벨은 화면상에서는 수정할수 없는 한줄 글자를 보여주는 위젯
- 그런데 라벨은 크기를 넘어가면 글자가 잘려서 보이지 않게 되는 것이 특징
- TextBrowser는 여러줄의 긴 데이터를 보여주기에 적합

```
# text browser
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType('mywindow.ui')[0]

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

        self.PrintBrowserButton.clicked.connect(self.printBrowserFunction)
        self.SetTextBrowserButton.clicked.connect(self.SetTextBrowserFunction)
        self.AppendTextBrowserButton.clicked.connect(self.AppendTextBrowserFunction)
        self.ClearBrowserButton.clicked.connect(self.ClearBrowserFunction)

    def printBrowserFunction(self):
        print(self.textBrowser.toPlainText())

    def SetTextBrowserFunction(self):
        self.textBrowser.setPlainText('텍스트 브라우저 글자 변경 테스트')

    def AppendTextBrowserFunction(self):
        self.textBrowser.append('텍스트 추가 테스트')

    def ClearBrowserFunction(self):
        self.textBrowser.clear()

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```




```
텍스트 브라우저 글자 변경 테스트
텍스트 추가 테스트
```

Line

- 한줄 짜리 글자를 입력 받을 수 있는 입력 위젯

```
# Line
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType('mywindow.ui')[0]

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

        self.lineEdit.textChanged.connect(self.lineeditFunction)
        self.lineEdit.returnPressed.connect(self.returnFunction)
        self.lineEditButton.clicked.connect(self.changeFunction)

    def lineeditFunction(self):
        self.label_text.setText(self.lineEdit.text())

    def returnFunction(self):
        print(self.lineEdit.text())
```

```
    def changeFunction(self):
        self.lineEdit.setText('글자 변경')

app = QApplication(sys.argv)
window = MyWindow()
window.show()
app.exec_()
```