

230306 파이썬 PyQt5

파이썬은 파이썬 기반의 GUI 라이브러리

- Tkinter - 파이썬 기본 GUI, 기본설치, 디자인이 떨어진다.
- PyQt - 아나콘다가 기본설치, 디자인이 좋다.

윈도우 기본창 띄우기

```
# 윈도우 기본창 띄우기

# 필요한 모듈 읽어오기 기본적인 창 구성요소를 제공하는 모든 것은 PyQt5.QtWidgets 에
# 만들어져 있다.
import sys
from PyQt5.QtWidgets import QApplication, QWidget

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

    # 창 기본 설정, 여기서는 제목만 설정하였다.
    def initUI(self):
        self.setWindowTitle("Hello World!!!")
        self.show()

# 파이썬은 기본적으로 인터프리터 이기 때문에 한줄한줄 코드를 실행하고 마치게 된다.
# 열린 윈도우 창을 계속 유지시켜야 한다.
app = QApplication(sys.argv)
ex = MyApp()
sys.exit(app.exec_())
```

정지 단추

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton
```

```

from PyQt5.QtCore import QApplication

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

    # 창 기본 설정, 여기서는 제목만 설정하였다.
    def initUI(self):
        self.setWindowTitle("Hello World!!!")
        btn = QPushButton('Quit', self)
        btn.clicked.connect(QCoreApplication.instance().quit)

        self.show()

# 파이썬은 기본적으로 인터프리터 이기 때문에 한줄한줄 코드를 실행하고 마치게 된다.
# 열려진 윈도우 창을 계속 유지시켜야 한다.
app = QApplication(sys.argv)
ex = MyApp()
sys.exit(app.exec_())

```

메뉴 생성 및 단추 작동

```

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_())

```

UI 연결

```

# QT Designer 로 생성한 기본 윈도우 창을 여는 방법

# 윈도우 기본창 띄우기

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_()

```

푸쉬버튼

```

# QT Designer 로 생성한 윈도우 창을 여는 방법

# 윈도우 기본창 띄우기

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_()

```

라디오 버튼

```

# 라디오 버튼은 여러개중 하나만 선택 가능

# 윈도우 기본창 띄우기

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

    def radioFunction(self):
        if self.radioButton_1.isChecked():
            print('라디오 버튼 1 체크')
        elif self.radioButton_2.isChecked():
            print('라디오 버튼 2 체크')
        elif self.radioButton_3.isChecked():
            print('라디오 버튼 3 체크')

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

```

체크 박스

```
# 체크 박스는 중복 선택이 가능함

# 윈도우 기본창 띄우기

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.checkBox1.stateChanged.connect(self.chkBoxFunction)
        self.checkBox2.stateChanged.connect(self.chkBoxFunction)
        self.checkBox3.stateChanged.connect(self.chkBoxFunction)

    # 여러개가 동시 선택될수 있기 때문에 elif를 사용하지 않는다.
    def chkFunction(self):
        if self.check1.isChecked():
            print('체크 버튼 1 체크')
        if self.check2.isChecked():
            print('체크 버튼 2 체크')
        if self.check3.isChecked():
            print('체크 버튼 3 체크')

    def chkBoxFunction(self):
        if self.checkBox1.isChecked():
            print('체크 박스버튼 1 체크')
        if self.checkBox2.isChecked():
            print('체크 박스버튼 2 체크')
        if self.checkBox3.isChecked():
            print('체크 박스버튼 3 체크')

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

라벨 처리

```
# 라벨 처리

# 윈도우 기본창 띄우기

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.melon.clicked.connect(self.melonFunction)
        self.genie.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_()
```

Text Browser

- 라벨은 화면상에서 수정할 수 없는 한줄 글자를 보여주는 위젯
- 하지만 라벨은 일정 크기를 넘어가면 글자가 잘리게 된다.
- 텍스트 브라우저는 여러줄의 긴 데이터를 보여주기에 적합하다.

```
# Text Browser

# 윈도우 기본창 띄우기

import sys
```

```

from PyQt5.QtWidgets import *
from PyQt5 import uic

form_class = uic.loadUiType("mywindow6.ui")[0]

class MyWindow(QMainWindow, form_class):

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

        self.printbrowserButton.clicked.connect(self.printFunction)
        self.settextButton.clicked.connect(self.setFunction)
        self.appendtextButton.clicked.connect(self.appendFunction)
        self.clearButton.clicked.connect(self.clearFunction)

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

    def setFunction(self):
        self.textBrowser.setPlainText("텍스트 브라우저 글자 변경 테스트")

    def appendFunction(self):
        self.textBrowser.append("텍스트 추가 테스트")

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

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

```

LineEdit

- 한줄 짜리 글자를 입력받을 수 있는 위젯

```

# Line Edit

# 윈도우 기본창 띄우기

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

form_class = uic.loadUiType("mywindow7.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.lineEditchangeButton.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_()

```