

2025년 서울 민간기업 맞춤형 매력일자리 사업

파이썬 GUI 프로그램 개발(PyQt5)

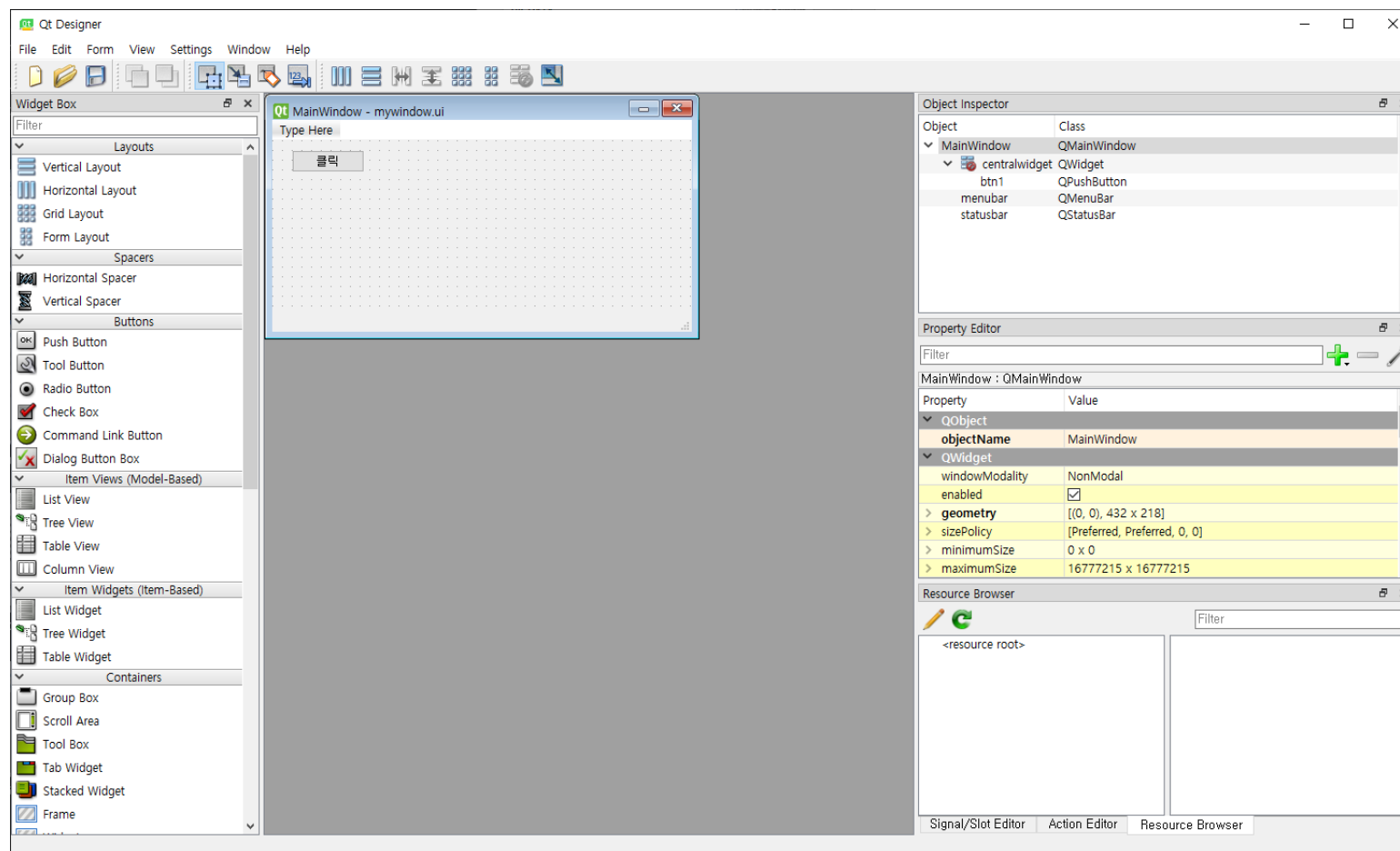
- 2025.05.07 -

PyQT 기초 다지기

PyQt5 프로그램 실습



PyQt5



PyQt5 프로그램 실습



PyQt5

<모듈설치하기>

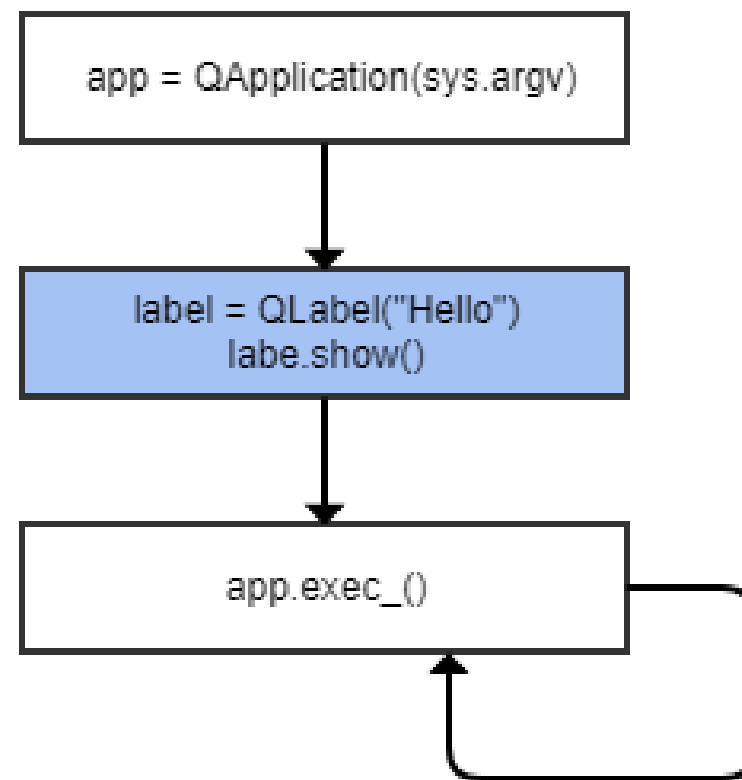
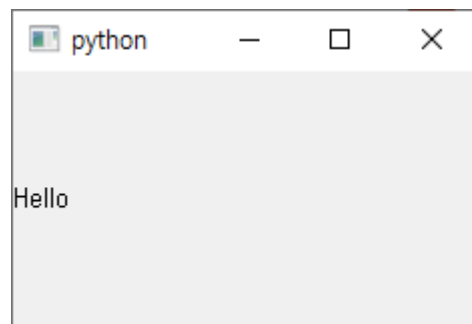
- `pip install PyQt5`
- `pip install -U PyQt5`
- **Mac** : `brew install pyqt5`

```
PS C:\Users\Wmatalcross> pip install PyQt5
WARNING: Ignoring invalid distribution -ip (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\wpython39\lib\site-packages)
Collecting PyQt5
  Using cached PyQt5-5.15.4-cp36.cp37.cp38.cp39-none-win_amd64.whl (6.8 MB)
Collecting PyQt5-Qt5>=5.15
  Using cached PyQt5_Qt5-5.15.2-py3-none-win_amd64.whl (50.1 MB)
Collecting PyQt5-sip<13,>=12.8
  Downloading PyQt5_sip-12.9.0-cp39-cp39-win_amd64.whl (63 kB)
    |████████████████████████████████████████| 63 kB 692 kB/s
WARNING: Ignoring invalid distribution -ip (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\wpython39\lib\site-packages)
Installing collected packages: PyQt5-sip, PyQt5-Qt5, PyQt5
WARNING: Ignoring invalid distribution -ip (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\wpython39\lib\site-packages)
Successfully installed PyQt5-5.15.4 PyQt5-Qt5-5.15.2 PyQt5-sip-12.9.0
WARNING: Ignoring invalid distribution -ip (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution -ip (c:\wpython39\lib\site-packages)
WARNING: Ignoring invalid distribution - (c:\wpython39\lib\site-packages)
PS C:\Users\Wmatalcross>
```

PyQt5 프로그램 실습

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

```
app = QApplication(sys.argv)  
label = QLabel("Hello")  
label.show()  
app.exec_()
```



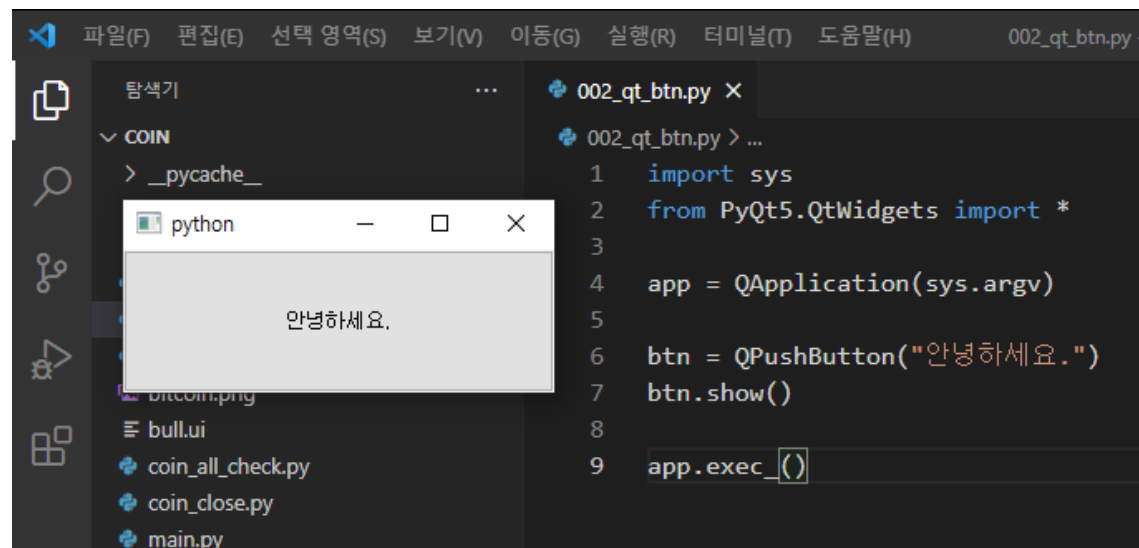
PyQt5 프로그램 실습

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

```
app = QApplication(sys.argv)
```

```
btn = QPushButton("안녕하세요")  
btn.show()
```

```
app.exec_()
```



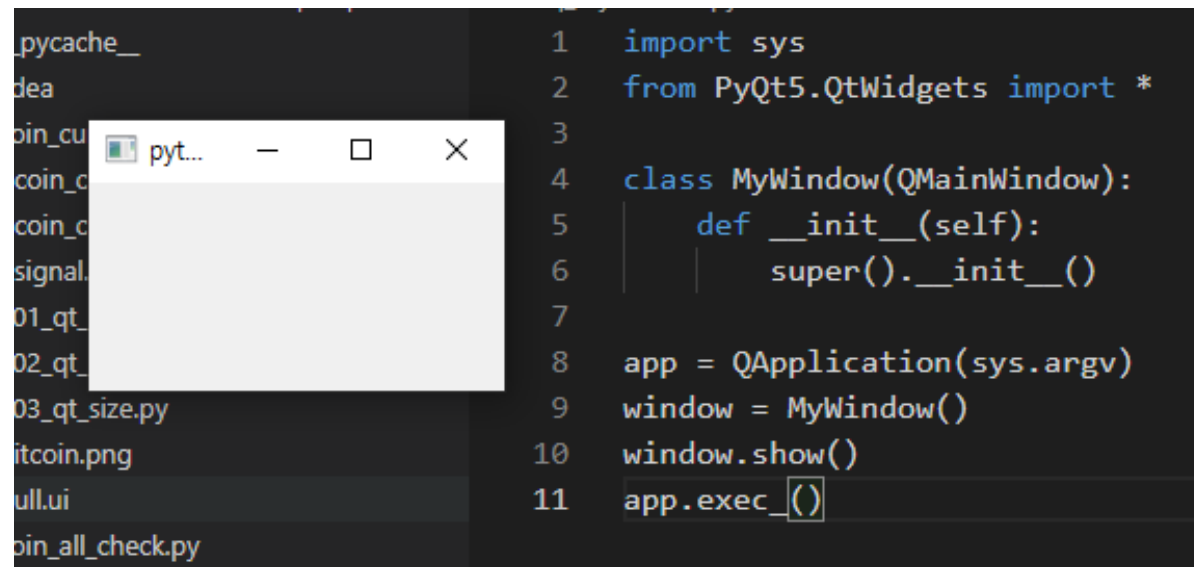
문제점 : 이러한 구조는 원하는 형식의 디자인 설계가 어렵다.

PyQt5 프로그램 실습

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

```
class MyWindow(QMainWindow)
    def __init__(self):
        super().__init__()
```

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



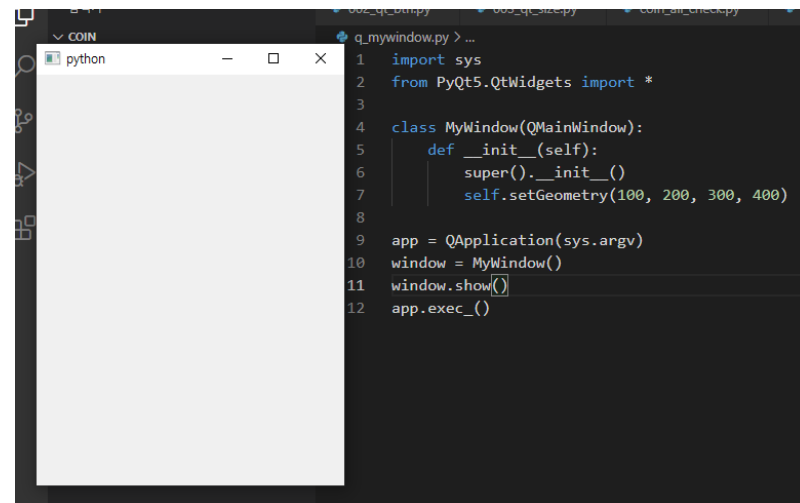
PyQt5 프로그램 실습

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

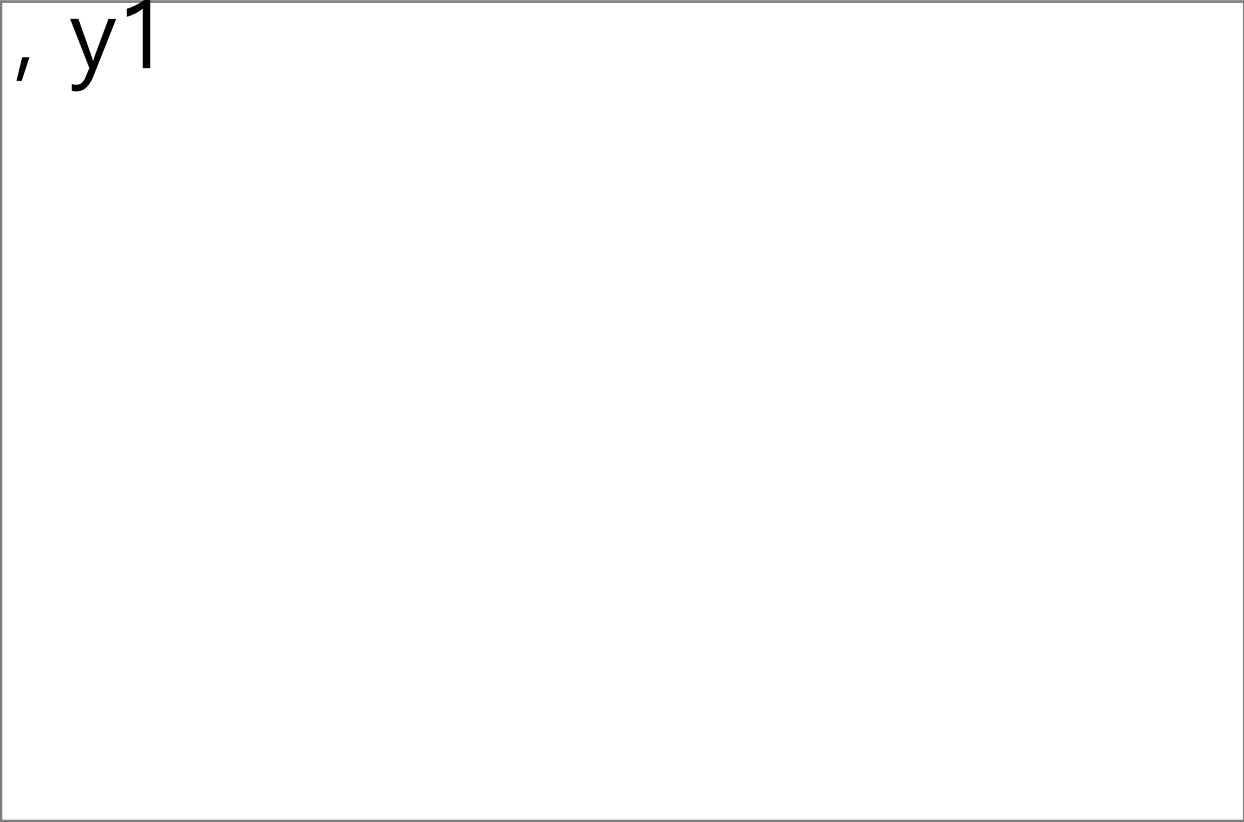
```
class MyWindow(QMainWindow):  
    def __init__(self):  
        super().__init__()
```

```
        self.setGeometry(100, 200, 300, 400)
```

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



x_1, y_1

A diagram showing a rectangle with its top-left corner labeled x_1, y_1 and its bottom-right corner labeled x_2, y_2 . The rectangle is defined by a thin black border. The labels are in a black serif font. The top-left label is positioned just outside the top-left corner, and the bottom-right label is positioned just outside the bottom-right corner.

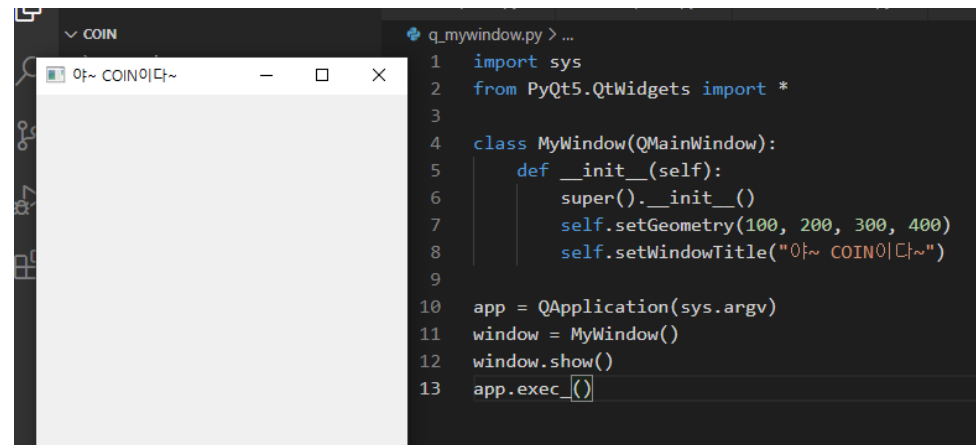
x_2, y_2

PyQt5 프로그램 실습

```
import sys
from PyQt5.QtWidgets import *

class MyWindow(QMainWindow)
    def __init__(self):
        super().__init__()
        self.setGeometry(100, 200, 300, 400)
        self.setWindowTitle("PyQt 테스트")

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



PyQt5 프로그램 실습

```
import sys
from PyQt5.QtWidgets import *
from PyQt5.QtGui import *

class MyWindow(QMainWindow)
    def __init__(self):
        super().__init__()
        self.setGeometry(100, 200, 300, 400)
        self.setWindowTitle("야~ COIN이다~")
        self.setWindowIcon(QIcon("아이콘이름 " ))

app = QApplication(sys.argv)
```



Search for icons, e.g. computer, facebook, folder...



Popular searches: [Instagram](#), [Facebook](#), [Arrow](#), [Phone](#), [Youtube](#), [Search](#), [User](#)

Search through 5,518,271 SVG or PNG icons

<https://www.iconfinder.com/>

PyQt5 프로그램 실습

```
import sys
```

```
from PyQt5.QtWidgets import *
```

```
from PyQt5.QtGui import *
```

```
class MyWindow(QMainWindow)
```

```
    def __init__(self):
```

```
        super().__init__()
```

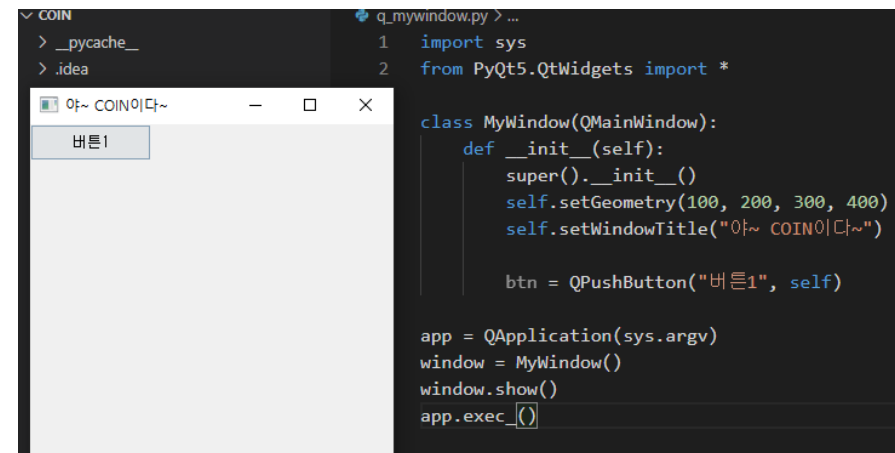
```
        self.setGeometry(100, 200, 300, 400)
```

```
        self.setWindowTitle("PyQt 테스트")
```

```
        self.setWindowIcon(QIcon("아이콘이름 " ))
```

```
        btn = QPushButton("버튼1", self)
```

```
app = QApplication(sys.argv)
```



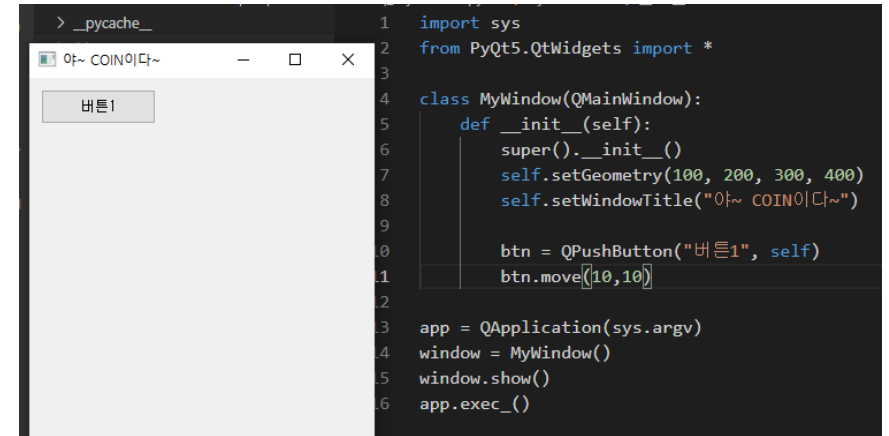
PyQt5 프로그램 실습

```
import sys
from PyQt5.QtWidgets import *
from PyQt5.QtGui import *

class MyWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.setGeometry(100, 200, 300, 400)
        self.setWindowTitle("야~ COIN이다~")
        self.setWindowIcon(QIcon("아이콘이름 "))

        btn = QPushButton("버튼1", self)
        btn.move(10, 10)

app = QApplication(sys.argv)
```




PyQt5 프로그램 실습

```
...  
class MyWindow(QMainWindow)  
    def __init__(self):  
        super().__init__()  
        self.setGeometry(100, 200, 300, 400)  
        self.setWindowTitle("야~ COIN이다~")  
        self.setWindowIcon(QIcon("아이콘이름 " ))  
  
        btn = QPushButton("버튼1", self)  
        btn.move(10, 10)  
        btn.clicked.connect(self.btn_clicked)  
  
    def btn_click(self):  
        print("버튼을 클릭하였습니다.")
```

PyQt5 프로그램 실습

디자인이 너무 어렵죠 ?

PyQt5 프로그램 실습



Qt Designer Download

Install Qt Designer on Windows or Mac.
Tiny download: Only 40MB!

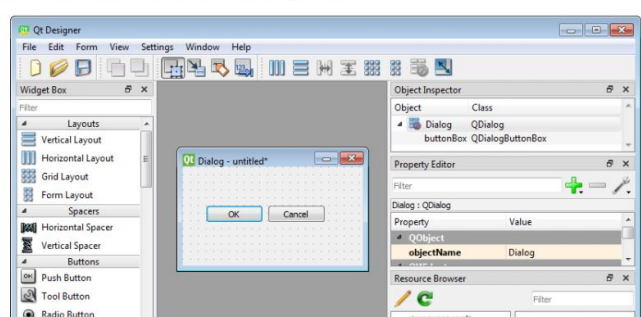
Many people want to use Qt Designer without having to download gigabytes of other software. Here are small, standalone installers of Qt Designer for Windows and Mac:

[Windows \(31 MB\)](#) [Mac \(40 MB\)](#)

If you encounter any problems, please just send us an email. We'd be happy to help.

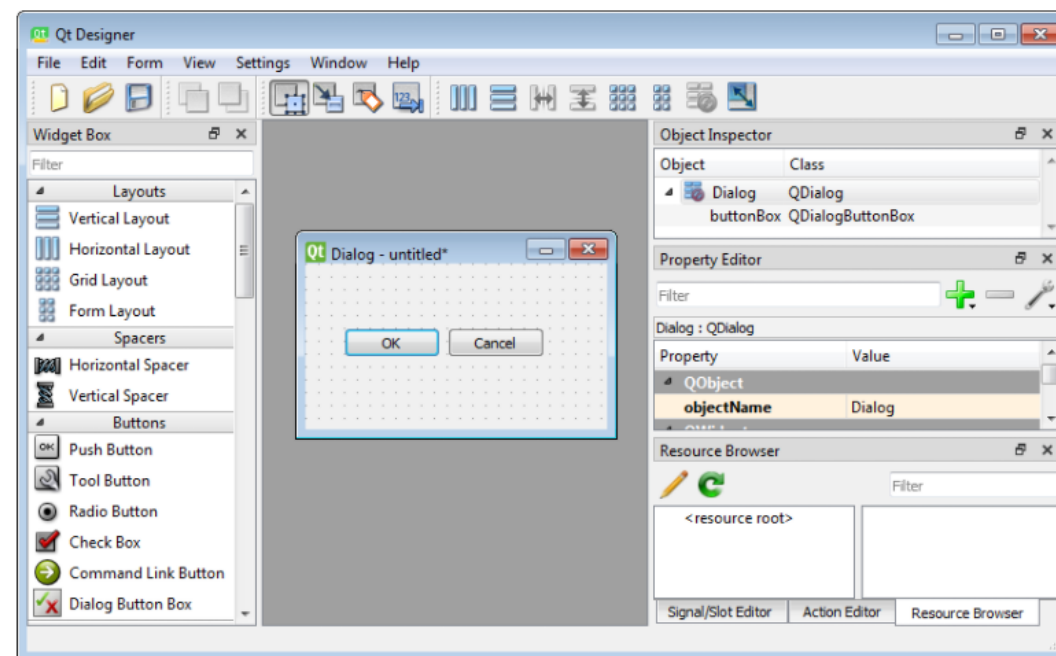
What is Qt Designer?

Qt Designer is a tool for quickly building graphical user interfaces with widgets from the Qt GUI framework. It gives you a simple drag-and-drop interface for laying out components such as buttons, text fields, combo boxes and more. Here is a screenshot of Qt Designer on Windows:



What is Qt Designer?

Qt Designer is a tool for quickly building graphical user interfaces with widgets from the Qt GUI framework. It gives you a simple drag-and-drop interface for laying out components such as buttons, text fields, combo boxes and more. Here is a screenshot of Qt Designer on Windows:



Qt Designer produces `.ui` files. This is a special XML-based format that stores your widgets as a tree. You can either load these files at runtime, or have them translated to a programming language such as C++ or Python.

PyQt5 프로그램 실습

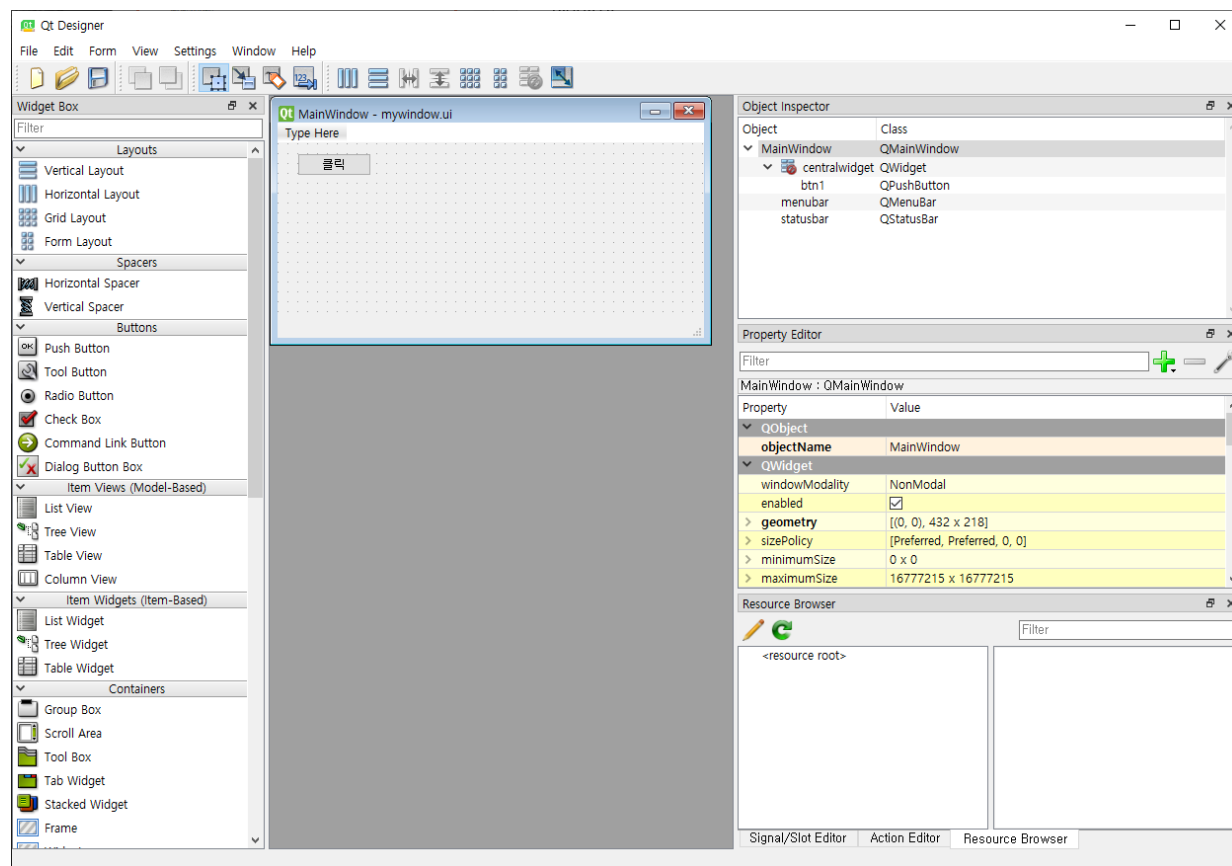
```
import sys
from PyQt5.QtWidgets import *
from PyQt5 import uic

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

class MyWindow(QMainWindow, form_class):
    def __init__(self):
        super().__init__()
        self.setupUi(self)
        self.btn1.clicked.connect(self.btn_clicked)

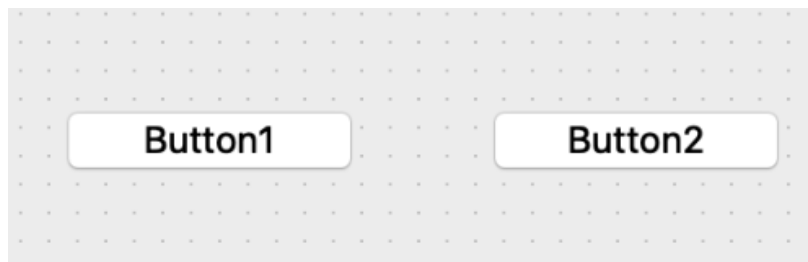
    def btn_clicked(self):
        print("버튼 클릭")

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



PyQt5 프로그램 실습 - QPushButton

`self.버튼이름.clicked.connect(함수)`



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

#UI파일 연결
#단, UI파일은 Python 코드 파일과 같은 디렉토리에 위치해야한다.
form_class = uic.loadUiType("pushbuttonTest.ui")[0]

#화면을 띄우는데 사용되는 Class 선언
class WindowClass(QMainWindow, form_class) :
    def __init__(self) :
        super().__init__()
        self.setupUi(self)

    #버튼에 기능을 연결하는 코드
    self.btn_1.clicked.connect(self.button1Function)
    self.btn_2.clicked.connect(self.button2Function)

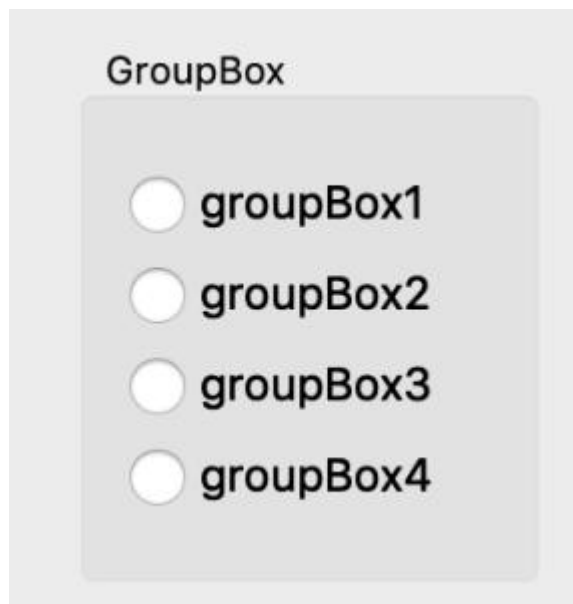
#btn_1이 눌리면 작동할 함수
def button1Function(self) :
    print("btn_1 Clicked")

#btn_2가 눌리면 작동할 함수
def button2Function(self) :
    print("btn_2 Clicked")

if __name__ == "__main__" :
    app = QApplication(sys.argv)
    myWindow = WindowClass()
    myWindow.show()
    app.exec_()
```


PyQt5 프로그램 실습 – QradioButton

`self.Radio버튼이름.clicked.connect(함수)`



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

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

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

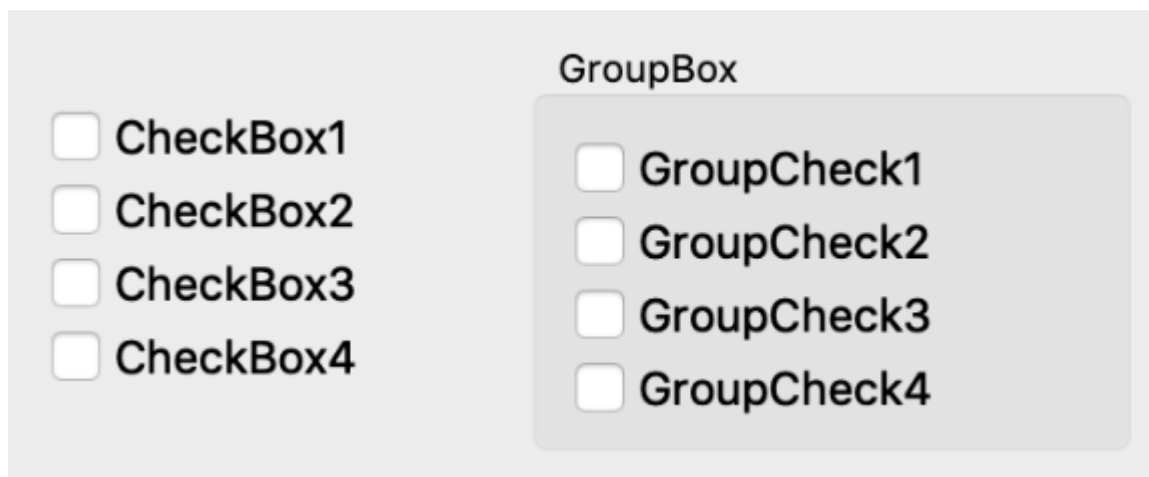
        #GroupBox안에 있는 RadioButton들을 연결합니다.
        #GroupBox의 자세한 설명은 02.14 GroupBox를 참고하세요.
        self.groupBox_rad1.clicked.connect(self.groupboxRadFunction)
        self.groupBox_rad2.clicked.connect(self.groupboxRadFunction)
        self.groupBox_rad3.clicked.connect(self.groupboxRadFunction)
        self.groupBox_rad4.clicked.connect(self.groupboxRadFunction)

    def groupboxRadFunction(self) :
        if self.groupBox_rad1.isChecked() : print("GroupBox_rad1 Chekced")
        elif self.groupBox_rad2.isChecked() : print("GroupBox_rad2 Checked")
        elif self.groupBox_rad3.isChecked() : print("GroupBox_rad3 Checked")
        elif self.groupBox_rad4.isChecked() : print("GroupBox_rad4 Checked")

if __name__ == "__main__" :
    app = QApplication(sys.argv)
    myWindow = WindowClass()
    myWindow.show()
    app.exec_()
```

PyQt5 프로그램 실습 – QradioButton

`self.체크박스이름.stateChanged.connect(함수)`



```
#GroupBox밖에 있는 CheckBox에 기능 연결
self.chk_1.stateChanged.connect(self.chkFunction)
self.chk_2.stateChanged.connect(self.chkFunction)
self.chk_3.stateChanged.connect(self.chkFunction)
self.chk_4.stateChanged.connect(self.chkFunction)
```

```
#GroupBox안에 있는 CheckBox에 기능 연결
self.groupchk_1.stateChanged.connect(self.groupchkFunction)
self.groupchk_2.stateChanged.connect(self.groupchkFunction)
self.groupchk_3.stateChanged.connect(self.groupchkFunction)
self.groupchk_4.stateChanged.connect(self.groupchkFunction)
```

```
def chkFunction(self) :
    #CheckBox는 여러개가 선택될 수 있기 때문에 elif를 사용하지 않습니다.
    if self.chk_1.isChecked() : print("chk_1 isChecked")
    if self.chk_2.isChecked() : print("chk_2 isChecked")
    if self.chk_3.isChecked() : print("chk_3 isChecked")
    if self.chk_4.isChecked() : print("chk_4 isChecked")
```

```
def groupchkFunction(self) :
    if self.groupchk_1.isChecked() : print("groupchk_1 isChecked")
    if self.groupchk_2.isChecked() : print("groupchk_2 isChecked")
    if self.groupchk_3.isChecked() : print("groupchk_3 isChecked")
    if self.groupchk_4.isChecked() : print("groupchk_4 isChecked")
```

PyQt5 프로그램 실습 – QLabel

This is Label - Change Text

ChangeText

PrintText

Method	설명
.text()	Label에 쓰여있는 글자를 가져옵니다.
.setText(String)	Label에 새롭게 글자를 작성합니다. Parameter에는 Label에 표시할 글자가 입력됩니다.
.clear()	Label에 쓰여있는 글자를 지웁니다.

PyQt5 프로그램 실습 – QLineEdit

This is Line Edit

This is Line Edit

Change Label Text

LineEdit의 글자가 바뀔 때 기능 실행

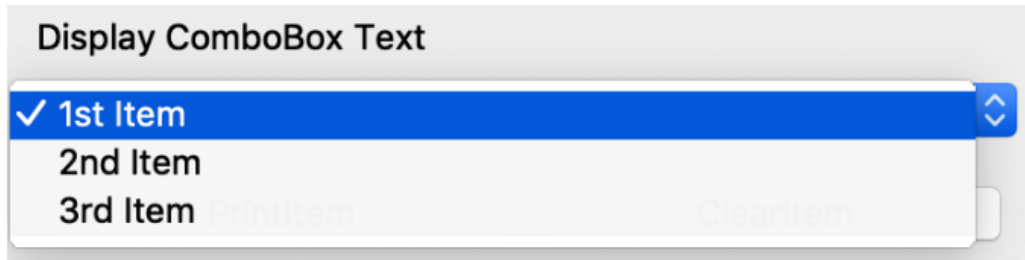
self.LineEdit이름.textChanged.connect(함수)

LineEdit에서 Return키(Enter키)가 눌렸을 때 기능 실행

self.LineEdit이름.returnPressed.connect(함수)

```
def __init__(self) :  
    super().__init__()  
    self.setupUi(self)  
  
    #버튼에 기능을 할당하는 코드  
    self.lineEdit_Test.textChanged.connect(self.lineEditTextFunction)  
    self.lineEdit_Test.returnPressed.connect(self.printTextFunction)  
    self.btn_changeText.clicked.connect(self.changeTextFunction)  
  
def lineeditTextFunction(self) :  
    self.lbl_textHere.setText(self.lineEdit_Test.text())  
  
def printTextFunction(self) :  
    #self.lineEdit이름.text()  
    #LineEdit에 있는 글자를 가져오는 메서드  
    print(self.lineEdit_Test.text())  
  
def changeTextFunction(self) :  
    #self.lineEdit이름.setText("String")  
    #LineEdit의 글자를 바꾸는 메서드  
    self.lineEdit_Test.setText("Change Text")
```

PyQt5 프로그램 실습 – QComboBox



```
self.ComboBox이름.currentIndexChanged.connect(함수)
```

항목의 반환

함수	설명
.currentIndex()	현재 ComboBox에서 선택된 항목의 Index를 반환합니다.
.currentText()	현재 ComboBox에서 선택된 항목의 글자를 반환합니다.
.count()	ComboBox에 몇개의 항목이 있는지를 그 개수를 반환합니다.
.itemText(index)	Index번째에 어떤 항목이 있는지 그 글자를 반환합니다. Parameter로 찾을 항목의 Index를 입력받습니다.

항목의 추가 및 삭제

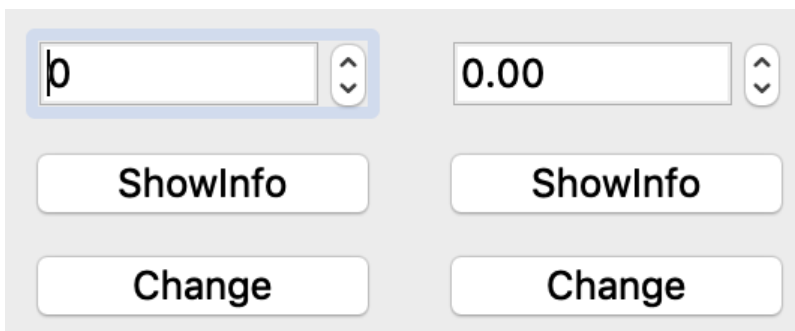
함수	설명
.addItem(String)	ComboBox의 맨 뒤에 항목을 추가합니다. Parameter로 추가할 항목의 글자를 입력받습니다.
.insertItem(index, String)	Index번째에 String이라는 항목을 추가합니다. Parameter로 항목을 추가할 위치(index)와 글자를 입력받습니다.
.removeItem(index)	index번째의 항목을 삭제합니다. Parameter로 삭제할 항목의 Index를 입력받습니다.
.clear()	ComboBox의 모든 항목을 삭제합니다.

PyQt5 프로그램 실습 – QComboBox

```
def __init__(self) :  
    super().__init__()  
    self.setupUi(self)  
  
    #프로그램 실행 시 두개의 ComboBox를 동기화시키는 코드  
    self.syncComboBox()  
  
    #ComboBox에 기능 연결  
    self.cmb_Test.currentIndexChanged.connect(self.comboBoxFunction)  
  
    #버튼에 기능 연결  
    self.btn_printItem.clicked.connect(self.printComboBoxItem)  
    self.btn_clearItem.clicked.connect(self.clearComboBoxItem)  
    self.btn_addItem.clicked.connect(self.addComboBoxItem)  
    self.btn_deleteItem.clicked.connect(self.deleteComboBoxItem)  
  
def syncComboBox(self) :  
    for i in range(0,self.cmb_Test.count()) :  
        self.cmb_second.addItem(self.cmb_Test.itemText(i))  
  
def comboBoxFunction(self) :  
    self.lbl_display.setText(self.cmb_Test.currentText())
```

```
def clearComboBoxItem(self) :  
    self.cmb_Test.clear()  
    self.cmb_second.clear()  
  
def printComboBoxItem(self) :  
    print(self.cmb_second.currentText())  
  
def addComboBoxItem(self) :  
    self.cmb_Test.addItem(self.lineEdit_addItem.text())  
    self.cmb_second.addItem(self.lineEdit_addItem.text())  
    print("Item Added")  
  
def deleteComboBoxItem(self) :  
    self.delidx = self.cmb_second.currentIndex()  
    self.cmb_Test.removeItem(self.delidx)  
    self.cmb_second.removeItem(self.delidx)  
    print("Item Deleted")
```

PyQt5 프로그램 실습 – QSpinBox



- Minimum : 이 SpinBox에서 가질 수 있는 최소값을 의미합니다.
- Maximum : 이 SpinBox에서 가질 수 있는 최대값을 의미합니다.
- SingleStep : 버튼을 눌렀을 때 숫자가 한번에 얼마나 늘어나고 줄어들지를 지정합니다.
- Value : SpinBox에서 값을 조절하지 않았을 때, 기본적으로 보여줄 값을 지정합니다.

QSpinBox		QDoubleSpinBox	
▶ suffix		▶ prefix	
▶ prefix		▶ suffix	
minimum	0	decimals	2
maximum	99	minimum	0.000000
singleStep	1	maximum	99.990000
value	0	singleStep	1.000000
displayIntegerBase	10	value	0.000000

QSpinBox와 QDoubleSpinBox의 Property Editor 화면

사진 02.10.02 QSpinBox와 QDoubleSpinBox의 Property Editor

`self.SpinBox/DoubleSpinBox이름.valueChanged.connect(함수)`

현재 상태의 반환

함수	설명
<code>.value()</code>	현재 SpinBox에 있는 값을 반환합니다.
<code>.minimum()</code>	현재 SpinBox의 최소값을 반환합니다.
<code>.maximum()</code>	현재 SpinBox의 최대값을 반환합니다.
<code>.singleStep()</code>	현재 SpinBox의 singleStep의 크기를 반환합니다.

PyQt5 프로그램 실습 – QSpinBox

SpinBox의 상태 변경

함수	설명
<code>.setRange(min,max)</code>	SpinBox의 최대,최솟값을 변경합니다. Parameter로 SpinBox의 새로운 최솟값과 최댓값을 입력받습니다. QSpinBox에서는 정수값을, QDoubleSpinBox에서는 실수값을 입력할 수 있습니다.
<code>.setStep(Value)</code>	SpinBox의 Step크기를 변경합니다. Parameter로 새로운 Step의 값을 입력받습니다. QSpinBox에서는 정수값을, QDoubleSpinBox에서는 실수값을 입력할 수 있습니다.

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

    self.spinbox_Test.valueChanged.connect(self.printValue)
    self.btn_showInfo.clicked.connect(self.printInfo)
    self.btn_changeRangeStep.clicked.connect(self.changeRangeStep)

    self.doublespinbox_Test.valueChanged.connect(self.printDoubleValue)
    self.btn_doubleShowInfo.clicked.connect(self.printDoubleInfo)
    self.btn_doubleChangeRangeStep.clicked.connect(self.doubleChangeRangeStep)

def printValue(self) :
    print(self.spinbox_Test.value())

def printInfo(self) :
    print("Maximum value is",self.spinbox_Test.maximum())
    print("Minimum value is",self.spinbox_Test.minimum())
    print("Step Size is",self.spinbox_Test.singleStep())

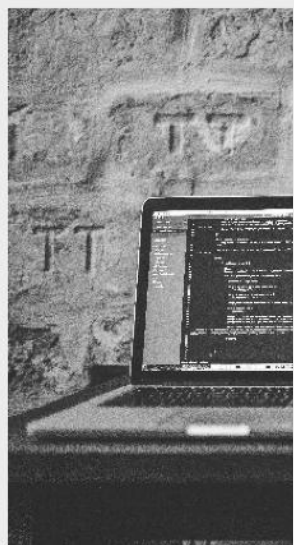
def changeRangeStep(self) :
    self.spinbox_Test.setRange(0,1000)
    self.spinbox_Test.setStep(10)

def printDoubleValue(self) :
    print(self.doublespinbox_Test.value())

def printDoubleInfo(self) :
    print("Maximum value is",self.doublespinbox_Test.maximum())
    print("Minimum value is",self.doublespinbox_Test.minimum())
    print("Step Size is",self.doublespinbox_Test.singleStep())

def doubleChangeRangeStep(self) :
    self.doublespinbox_Test.setRange(0,1000)
    self.doublespinbox_Test.setStep(1.5)
```

PyQt5 프로그램 시스템 QPixmap



TextLabel

QPixmap을 Label에 표시하기

```
label이름.setPixmap(QPixmap 객체)
```

이미지의 Load/Save

함수	설명
.load(fileName)	이미지 파일에서 이미지를 가져오는 함수입니다. Parameter로 이미지의 경로를 문자열의 형태로 입력해야 합니다.
.loadFromData(Data)	이미지의 정보값을 갖고 있는 객체에서 이미지를 가져오는 함수입니다. Parameter로 이미지의 정보값을 갖고 있는 객체를 입력해야 합니다. 주로 Web에서 이미지를 가져올 때 많이 사용합니다.
.save(fileName)	QPixmap이 가지고 있는 이미지를 파일로 저장할 때 사용합니다. Parameter로 파일의 이름을 문자열의 형태로 입력해야 합니다.

PyQt5 프로그램 실습 – QPixmap

이미지의 크기와 관련된 함수

이미지의 크기를 조절하고, 반환하는 함수들입니다. 이미지의 크기를 반환하는 함수는 `QSize`라는 형식의 객체를 반환합니다. `QSize`에 대한 자세한 설명은 이 페이지의 하위페이지에 있는 [02.15.01 QSize](#)의 설명을 읽어보시기 바랍니다.

함수	설명
<code>.scaled(width,height)</code>	이미지의 가로, 세로 크기를 조절합니다. Parameter로 가로, 세로 크기값을 입력받습니다. 사진의 원본 비율을 무시하고 크기를 조정해야 할 때 사용하기 좋습니다.
<code>.scaledToWidth(width)</code>	이미지의 가로크기를 조절합니다. 이 함수를 이용할 경우 세로크기는 새로운 가로값에 따라서 자동으로 조절되며, 이미지의 가로세로 비율은 원본과 동일하게 유지됩니다. Parameter로 가로크기값을 입력합니다.
<code>.scaledToHeight(height)</code>	이미지의 세로크기를 조절합니다. 이 함수를 이용할 경우, 가로크기는 새로운 세로값에 따라서 자동으로 조절되며, 이미지의 가로세로 비율은 원본과 동일하게 유지됩니다. Parameter로 세로크기값을 입력합니다.
<code>.size()</code>	<code>QPixmap</code> 이 가지고 있는 이미지의 크기를 <code>QSize</code> 형식으로 반환합니다.

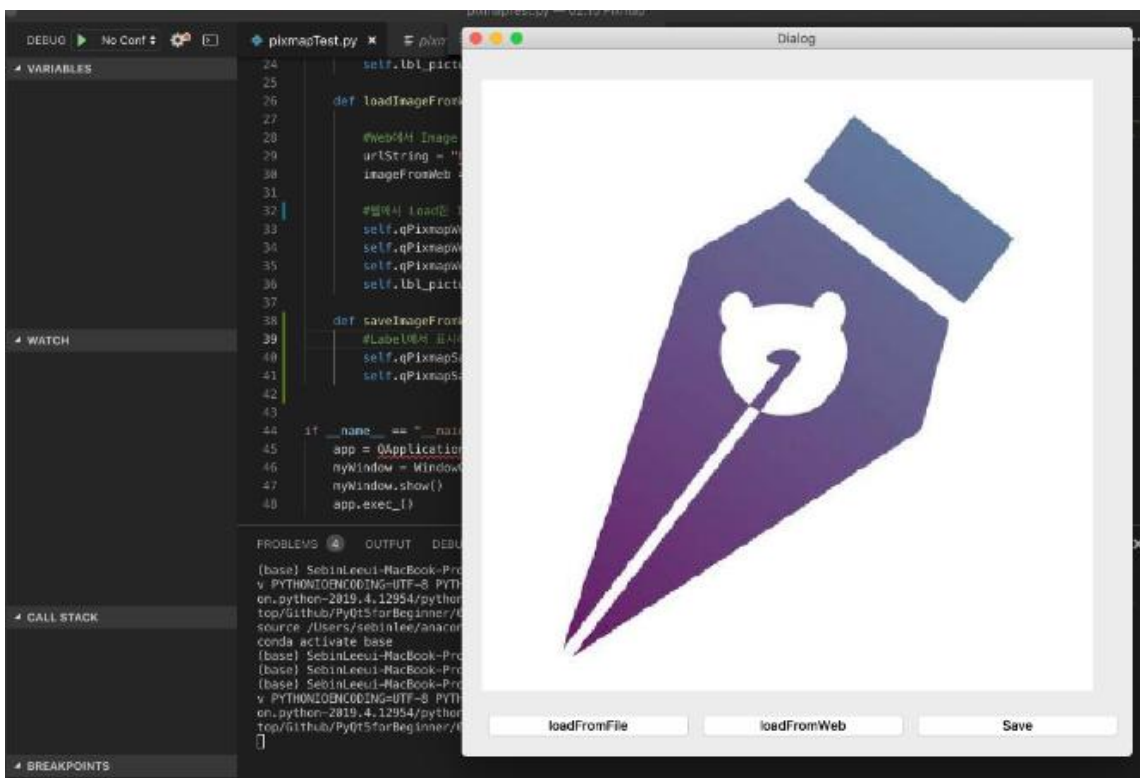
Pixmap과 관련된 Label의 함수

`Label`에는 `Pixmap`과 관련된 함수들이 몇가지 존재합니다. 이 함수들에 대해서 알아보도록 하겠습니다.

이 부분에서 소개하는 함수들은 위에서 설명한 함수들과 다르게 함수 앞에 `QPixmap`객체를 붙이는 것이 아니라 `label`의 이름을 붙여야 합니다.

함수	설명
<code>.setPixmap(QPixmap)</code>	<code>Label</code> 의 영역에 사진을 표시합니다. Parameter로 표시할 이미지값을 가지고 있는 <code>QPixmap</code> 객체가 필요합니다.
<code>.pixmap()</code>	<code>Label</code> 이 표시하고 있는 사진을 반환합니다. 반환된 값은 <code>QPixmap</code> 객체의 형식을 가집니다.

PyQt5 프로그램 실습 – QPixmap



```
def __init__(self) :
```

```
    super().__init__()
```

```
    self.setupUi(self)
```

```
    self.btn_loadFromFile.clicked.connect(self.loadImageFromFile)
```

```
    self.btn_loadFromWeb.clicked.connect(self.loadImageFromWeb)
```

```
    self.btn_savePicture.clicked.connect(self.saveImageFromWeb)
```

```
def loadImageFromFile(self) :
```

#QPixmap 객체 생성 후 이미지 파일을 이용하여 QPixmap에 사진 데이터 Load하고, Label을 이용하여 화면에 표시

```
    self.qPixmapFileVar = QPixmap()
```

```
    self.qPixmapFileVar.load("testImage.jpg")
```

```
    self.qPixmapFileVar = self.qPixmapFileVar.scaledToWidth(600)
```

```
    self.lbl_picture.setPixmap(self.qPixmapFileVar)
```

```
def loadImageFromWeb(self) :
```

```
    #Web에서 Image 정보 로드
```

```
    urlString = "https://avatars1.githubusercontent.com/u/44885477?s=460&v=4"
```

```
    imageFromWeb = urllib.request.urlopen(urlString).read()
```

#웹에서 Load한 Image를 이용하여 QPixmap에 사진데이터를 Load하고, Label을 이용하여 화면에 표시

```
    self.qPixmapWebVar = QPixmap()
```

```
    self.qPixmapWebVar.loadFromData(imageFromWeb)
```

```
    self.qPixmapWebVar = self.qPixmapWebVar.scaledToWidth(600)
```

```
    self.lbl_picture.setPixmap(self.qPixmapWebVar)
```

```
def saveImageFromWeb(self) :
```

#Label에서 표시하고 있는 사진 데이터를 QPixmap객체의 형태로 반환받은 후, save함수를 이용해 사진 저장

```
    self.qPixmapSaveVar = self.lbl_picture.pixmap()
```

```
    self.qPixmapSaveVar.save("SavedImage.jpg")
```

PyQt5 프로그램 실습 - 주가 분석 프로그램

IndentationError: expected an indented block

가상화폐 상승장/하락장 알리미 - bull.ui

	가상화폐	현재가	5일 이동평균	상승장/하락장
1	BTC			
2	ETH			
3	DOGE			
4	XRP			

종목명	현재가	5일 이동평균	상승장/하락장
삼성전자			
삼성전자			
삼성전자			
삼성전자			

5초에 한번씩 정보가 갱신되도록 프로그램처리

PyQt5 프로그램 실습 – financereader

<https://wikidocs.net/172650>

프로그램 설치

```
pip install -U finance-datareader
```

패키지 인포트

```
import FinanceDataReader as fdr
```

종목 가져오기

```
df_krx = fdr.StockListing('KRX')
```

```
df = fdr.DataReader('KS11', '2015')
```

```
df.head()
```

팔란티어 테크	14.7292	4,418.7677	+29,519.5623
PLTR	113.4482	34,034.4600	+668.04%
테슬라	175.4238	11,051.7040	+6,225.6060
TSLA	275.3700	17,348.3100	+56.33%

엔비디아	13.7946	13.7946	+98.1854
NVDA	112.2900	112.2900	+711.76%

PyQt5 프로그램 실습 – QWebEngineView

검색어를 입력하세요.

초보자를 위한 Python GUI 프...

WikiDocs

초보자를 위한 Python GUI 프로그래밍 - PyQt5



지은이 : 이세빈
최종 편집일시 : 2019년 5월 25일 1:11 오전
저작권 : 
4 명이 추천

저자소개

이세빈(lsbjeju@gmail.com)

송실대학교 IT대학 글로벌미디어학부를 휴학중이며, 현재 경기도 부천시 대명컴퓨터학원에서 강사로 근무하고 있다.

現) 대명컴퓨터학원 OA/프로그래밍/그래픽/영상과목 강사
現) 송실대학교 IT대학 글로벌미디어학부 휴학
前) 송실대학교 제 12대 IT대학 학생회 홍보국장
前) 송실대학교 글로벌미디어학부 디자인소모임 디자인술술회장

[초보자를 위한 Python GUI 프로그래밍 예제코드](#)

01. PyQt의 시작

01.01 PyQt란 무엇인가?

01.02 Qt Designer의 설치와 실행

01.03 Qt Designer의 화면구성

01.04 Qt Designer를 이용한 UI

02. 위젯

02.01 들어가기에 앞서(필독)

02.02 Button - QPushButton

02.03 Button - QRadioButton

02.04 Button - QCheckBox

02.05 Display - QLabel

02.06 Display - QTextBrowser

02.07 Input - QLineEdit

QWebEngineView	
url	https://wikidocs.net/book/2944
zoomFactor	1.000000

사진 02.16.02 QWebEngineView Property Editor

QWebEngineView의 시그널

QWebEngineView에서 새롭게 페이지를 로딩하거나, 웹페이지의 Url이 바뀌었을 때, 특정 기능을 수행하도록 하는 코드를 작성해보겠습니다.

QWebEngineView의 시그널은 다음과 같이 작성하며, 생성자의 내부에 작성해야 합니다.

WebEngineView에서 새로운 웹페이지의 Load를 시작했을 때

self.WebEngineView이름.loadStarted.connect(함수)

WebEngineView에서 새로운 웹페이지를 Load 하는 중일때

self.WebEngineView이름.loadProgress.connect(함수)

WebEngineView에서 새로운 웹페이지의 Load를 끝냈을 때

self.WebEngineView이름.loadFinished.connect(함수)

WebEngineView의 url이 바뀌었을 때

self.WebEngineView이름.urlChanged.connect(함수)

PyQt5 프로그램 실습 – QWebEngineView

QWebEngineView의 함수

함수	설명
.back()	웹브라우저에서 사용할 수 있는 이전 페이지로 이동 기능입니다.
.forward()	웹브라우저에서 사용할 수 있는 앞 페이지로 이동 기능입니다.
.reload()	웹페이지를 새로고침합니다.
.stop()	웹페이지의 로딩을 중단합니다.
.load(url)	QWebEngineView에 새로운 웹페이지를 로드합니다. Parameter로 새로운 웹 페이지의 url이 QUrl의 형식으로 입력되어야 합니다.
.url().toString()	현재 QWebEngineView에서 보여주고 있는 사이트의 url을 문자열의 형태로 반환합니다.

위의 .load()함수의 경우 Parameter로 QUrl형식의 객체를 입력해야합니다. 따라서 .load함수를 쓸때는 아래와 같이 사용해주셔야 합니다.

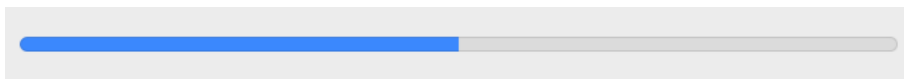
```
from PyQt5.QtCore import *  
self.QWebEngineView이름.load(QUrl("주소(문자열)"))
```

PyQt5 프로그램 실습 – QWebEngineView

```
def __init__(self) :  
    super().__init__()  
    self.setupUi(self)  
  
    #WebEngineView의 시그널  
    self.webEngineView_Test.loadStarted.connect(self.printLoadStart)  
    self.webEngineView_Test.loadProgress.connect(self.printLoading)  
    self.webEngineView_Test.loadFinished.connect(self.printLoadFinished)  
    self.webEngineView_Test.urlChanged.connect(self.urlChangedFunction)  
  
    #버튼들에 기능을 연결  
    self.btn_setUrl.clicked.connect(self.urlGo)  
    self.btn_back.clicked.connect(self.btnBackFunc)  
    self.btn_forward.clicked.connect(self.btnForwardFunc)  
    self.btn_reload.clicked.connect(self.btnRelaodFunc)  
    self.btn_stop.clicked.connect(self.btnStopFunc)  
  
    #WebEngineView의 시그널에 연결된 함수들  
    def printLoadStart(self) : print("Start Loading")  
    def printLoading(self) : print("Loading")  
    def printLoadFinished(self) : print("Load Finished")  
  
    def urlChangedFunction(self) :  
        self.line_url.setText(self.webEngineView_Test.url().toString())  
        print("Url Changed")
```

```
#버튼을 눌렀을 때 실행될 함수들  
def urlGo(self) :  
    self.webEngineView_Test.load(QUrl(self.line_url.text()))  
  
def btnBackFunc(self) :  
    self.webEngineView_Test.back()  
  
def btnForwardFunc(self) :  
    self.webEngineView_Test.forward()  
  
def btnRelaodFunc(self) :  
    self.webEngineView_Test.reload()  
  
def btnStopFunc(self) :  
    self.webEngineView_Test.stop()
```

PyQt5 프로그램 실습 – QProgressBar



QProgressBar	
minimum	0
maximum	100
value	-1
alignment	AlignLeft, AlignVCenter
textVisible	<input checked="" type="checkbox"/>
orientation	Horizontal
invertedAppear...	<input type="checkbox"/>
textDirection	TopToBottom
format	%p%

사진 02.17.02 QProgressBar Property Editor

- Minimum : 이 ProgressBar가 가질 수 있는 최소값을 의미합니다.
- Maximum : 이 ProgressBar가 가질 수 있는 최대값을 의미합니다.
- Value : 이 ProgressBar가 가지고 있는 값을 의미합니다.
- Orientation : ProgressBar의 방향을 지정합니다. 가로(Horizontal), 세로(Vertical)을 선택할 수 있습니다.

QProgressBar의 값이 변경되었을 때 `self.ProgressBar이름.valueChanged.connect(함수)`

QProgressBar의 값에 관련된 함수

함수	설명
<code>.value()</code>	QProgressBar가 가지고 있는 현재값을 반환합니다.
<code>.setValue(Value)</code>	QProgressBar에 새로운 Value값을 설정합니다. Parameter로 새로운 Value값이 필요합니다.
<code>.reset()</code>	QProgressBar의 Value값을 초기화합니다. 이때 Value값은 QProgressBar의 최솟값 - 1을 한 값으로 초기화됩니다.

PyQt5 프로그램 실습 – QProgressBar

QProgressBar의 범위와 관련된 함수

함수	설명
.minimum()	QProgressBar의 최솟값을 반환합니다.
.maximum()	QProgressBar의 최댓값을 반환합니다.
.setMinimum(min)	QProgressBar의 최솟값을 새로 설정합니다. Parameter로 새로운 최솟값이 필요합니다.
.setMaximum(max)	QProgressBar의 최댓값을 새로 설정합니다. Parameter로 새로운 최댓값이 필요합니다.
.setRange(min,max)	QProgressBar의 최솟값과 최댓값을 새로 설정합니다. Parameter로 새로운 최솟값, 최댓값이 필요합니다.

```
def __init__(self) :  
    super().__init__()   
    self.setupUi(self)  
  
    #ProgressBar의 시그널  
    self.progressBar_Test.valueChanged.connect(self.printValue)  
  
    #QTimer를 이용하여 매초마다 ProgressBar의 값이 1씩 늘어나게 설정합니다.  
    #QTimer의Interval을 1000으로 설정한 후, ProgressBar의 값이 늘어나게 하는 함수를 연결하  
    고 QTimer를 시작합니다.  
    #QTimer에 대한 설명은 02.17.01 QTimer에서 보실 수 있습니다.  
    self.timerVar = QTimer()  
    self.timerVar.setInterval(1000)  
    self.timerVar.timeout.connect(self.progressBarTimer)  
    self.timerVar.start()  
  
def progressBarTimer(self) :  
    self.time = self.progressBar_Test.value()  
    self.time += 1  
    self.progressBar_Test.setValue(self.time)  
  
    #ProgressBar의 값이 최댓값 이상이 되면 Timer를 중단시켜 ProgressBar의 값이 더이상 증  
    가하지 않게 합니다.  
    if self.time >= self.progressBar_Test.maximum() :  
        self.timerVar.stop()  
  
def printValue(self) :  
    print(self.progressBar_Test.value())
```


PyQt5 프로그램 실습 – QTimer

QTimer의 조작과 관련된 함수

함수	설명
.start()	QTimer가 시간을 체크하기 시작합니다.
.start(msec)	QTimer가 Parameter만큼의 시간이 지난 후 시간을 체크하기 시작합니다. Parameter로 시간을 입력해야 하며, 단위는 ms입니다.
.stop()	QTimer를 중지합니다.

QTimer의 Interval과 관련된 함수

함수	설명
.setInterval(msec)	QTimer의 Interval을 설정합니다. Parameter로 Interval을 ms단위로 입력합니다.
.timeout.connect(함수)	매 Interval마다 어떤 함수를 실행할지를 결정합니다. 만약, setInterval 함수로 Interval을 설정하지 않은 경우, 1ms마다 함수가 반복됩니다.

QTimer의 상태반환

함수	설명
.interval()	QTimer의 Interval을 반환합니다.
.isActive()	QTimer가 작동중인지 체크합니다. QTimer가 작동중이면 True를, 멈춰있으면 False를 반환합니다.

PyQt5 프로그램 실습 – Table Widget

	학번	학부	
이세빈	20170001	글로벌미디어...	
김민수	20170002	전자정보공학부	
홍길동	20180003	소프트웨어학부	
이석준	20190004	컴퓨터학부	
5			
6			

QTableWidget의 시그널

TableWidget의 Cell의 내용이 바뀌었을 때 기능 실행

self.TableWidget이름.cellChanged.connect(함수)

TableWidget에서 선택된 Cell이 바뀌었을 때 기능 실행

self.TableWidget이름.currentCellChanged.connect(함수)

TableWidget에서 Cell이 클릭 되었을 때 기능 실행

self.TableWidget이름.cellClicked.connect(함수)

TableWidget에서 Cell이 더블클릭 되었을 때 기능 실행

self.TableWidget이름.cellDoubleClicked.connect(함수)

PyQt5 프로그램 실습 – Table Widget

```
#QTableWidgetItem객체의 생성
qTableWidgetItemVar = QTableWidgetItem("항목이 가질 문자열 입력")

#QTableWidgetItem객체의 문자열 반환
qTableWidgetItemVar.text()

#QTableWidgetItem객체의 문자열 수정
qTableWidgetItemVar.setText("새로운 문자열")
```

현재상태의 반환

함수	설명
.item(row, col)	row번째 줄, col번째 열의 항목을 반환합니다. 이때, 반환된 항목은 QTableWidgetItem의 형식으로 반환됩니다.
.currentItem()	현재 선택하고 있는 항목을 반환합니다. 이때, 반환된 항목은 QTableWidgetItem의 형식으로 반환됩니다.
.currentRow()	현재 선택하고 있는 항목의 행을 반환합니다.
.currentColumn()	현재 선택하고 있는 항목의 열을 반환합니다.
.selectedItems()	선택한 항목들을 리스트 형식으로 반환합니다. 리스트 안에는 선택된 항목들이 QTableWidgetItem의 형식으로 포함되어 있습니다.
.selectedRanges()	현재 선택한 범위를 QTableWidgetItemSelectionRange의 형식으로 반환합니다.

PyQt5 프로그램 실습 – Table Widget

항목의 추가, 삭제와 관련된 함수

함수	설명
<code>.setItem(row,col,item)</code>	row번째 줄, col번째 칸에 Item이라는 항목을 추가합니다. 이때, Item은 QTableWidgetItem 형식의 객체여야 합니다.
<code>.takeItem(row,col)</code>	row번째 줄, col번째 칸에 있는 항목을 삭제합니다. 이때, 삭제된 항목은 QTableWidgetItem 형식의 객체로 반환됩니다.
<code>.clear()</code>	행과 열의 Header를 포함한 모든 항목을 삭제합니다.
<code>.clearContents()</code>	행과 열의 Header를 제외한 모든 항목을 삭제합니다.

Table의 행과 열에 관련된 함수

함수	설명
<code>.currentColumnCount()</code>	현재 Table Widget에 존재하는 열의 개수를 반환합니다.
<code>.currentRowCount()</code>	현재 Table Widget에 존재하는 행의 개수를 반환합니다.
<code>.setColumnCount(col)</code>	현재 Table Widget의 열의 개수를 col개로 설정합니다.
<code>.setRowCount(row)</code>	현재 Table Widget의 행의 개수를 row개로 설정합니다.

함수	설명
<code>.horizontalHeaderItem(col)</code>	col번째 열의 Header를 QTableWidgetItem 형식의 객체로 반환합니다.
<code>.takeHorizontalHeader(col)</code>	col번째 열의 Header를 삭제합니다. 이때 삭제된 항목은 QTableWidgetItem 형식의 객체로 반환됩니다.
<code>.setHorizontalHeaderItem(col, item)</code>	col번째 열의 Header를 Item이 가지고 있는 글자로 바꿉니다. 이때, Item은 QTableWidgetItem 형식의 객체여야 합니다.
<code>.setHorizontalHeaderLabels(List)</code>	열들의 Header를 일괄적으로 변경합니다. Parameter로 새로운 Header가 될 문자열이 들어있는 List가 필요하며, List의 0번째 항목이 0번째 열의 Header가 됩니다.

PyQt5 프로그램 실습 – Table Widget

함수	설명
<code>.verticalHeaderItem(row)</code>	row번째 줄의 Header를 <code>QTableWidgetItem</code> 형식의 객체로 반환합니다.
<code>.takeVerticalHeader(row)</code>	row번째 줄의 Header를 삭제합니다. 이때 삭제된 항목은 <code>QTableWidgetItem</code> 형식의 객체로 반환됩니다.
<code>.setVerticalHeaderItem(row, item)</code>	row번째 줄의 Header를 Item이 가지고 있는 글자로 바꿉니다. 이때, Item은 <code>QTableWidgetItem</code> 형식의 객체여야 합니다.
<code>.setVerticalHeaderLabels(List)</code>	행들의 Header를 일괄적으로 변경합니다. Parameter로 새로운 Header가 될 문자열이 들어있는 List가 필요하며, List의 0번째 항목이 0번째 행의 Header가 됩니다.

PyQT 기초 다지기

PyQt를 이용한 트레이닝설계

PyQt5 프로그램 실습

```
import pybithumb
```

```
btc = pybithumb.get_ohlcv("BTC")  
print(btc)
```

시가, 고가, 저가, 종가, 거래량

```
>>>  
>>> print(btc)
```

time	open	high	low	close	volume
2013-12-27 00:00:00	737000.0	755000.0	737000.0	755000.0	3.780000
2013-12-28 00:00:00	750000.0	750000.0	750000.0	750000.0	12.000000
2013-12-29 00:00:00	750000.0	750000.0	728000.0	739000.0	19.058000
2013-12-30 00:00:00	740000.0	772000.0	740000.0	768000.0	9.488973
2013-12-31 00:00:00	768000.0	800000.0	763000.0	768000.0	18.650350
...
2021-05-09 00:00:00	70227000.0	72909000.0	69156000.0	70670000.0	5048.683242
2021-05-10 00:00:00	70669000.0	72140000.0	70020000.0	70858000.0	4372.935281
2021-05-11 00:00:00	70750000.0	77974000.0	67568000.0	68476000.0	5919.611842
2021-05-12 00:00:00	68539000.0	70357000.0	68201000.0	68757000.0	4090.260430
2021-05-13 23:00:00	68782000.0	69058000.0	58500000.0	63242000.0	9393.141704

```
[2605 rows x 5 columns]  
>>>
```


PyQt5 프로그램 실습

```
import pybithumb
```

```
btc = pybithumb.get_ohlcv("BTC")  
close = btc['close']  
print(close)
```

```
>>> btc = pybithumb.get_ohlcv("BTC")  
>>> close = btc['close']  
>>> print(close)  
time  
2013-12-27 00:00:00      755000.0  
2013-12-28 00:00:00      750000.0  
2013-12-29 00:00:00      739000.0  
2013-12-30 00:00:00      768000.0  
2013-12-31 00:00:00      768000.0  
...  
2021-05-09 00:00:00      70670000.0  
2021-05-10 00:00:00      70858000.0  
2021-05-11 00:00:00      68476000.0  
2021-05-12 00:00:00      68757000.0  
2021-05-13 23:00:00      63451000.0  
Name: close, Length: 2605, dtype: float64  
>>> 
```

PyQt5 프로그램 실습

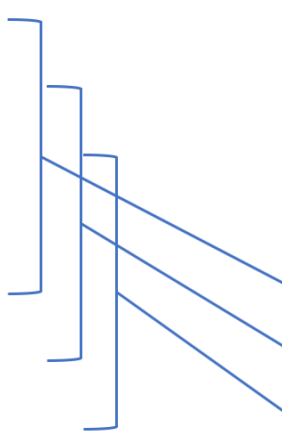
```
import pybithumb
```

```
btc = pybithumb.get_ohlcv("BTC")  
close = btc['close']
```

```
window = close.rolling(5)  
ma5 = window.mean()
```

```
print(ma5)
```

close		ma5	
2018-11-12	7254000.0		7263200.0
2018-11-13	7234000.0		7241800.0
2018-11-14	7149000.0		7226800.0
2018-11-15	6271000.0		7027800.0
2018-11-16	6443000.0		6870200.0
2018-11-17	6377000.0		6694800.0
2018-11-18	6408000.0		6529600.0



PyQt5 프로그램 실습

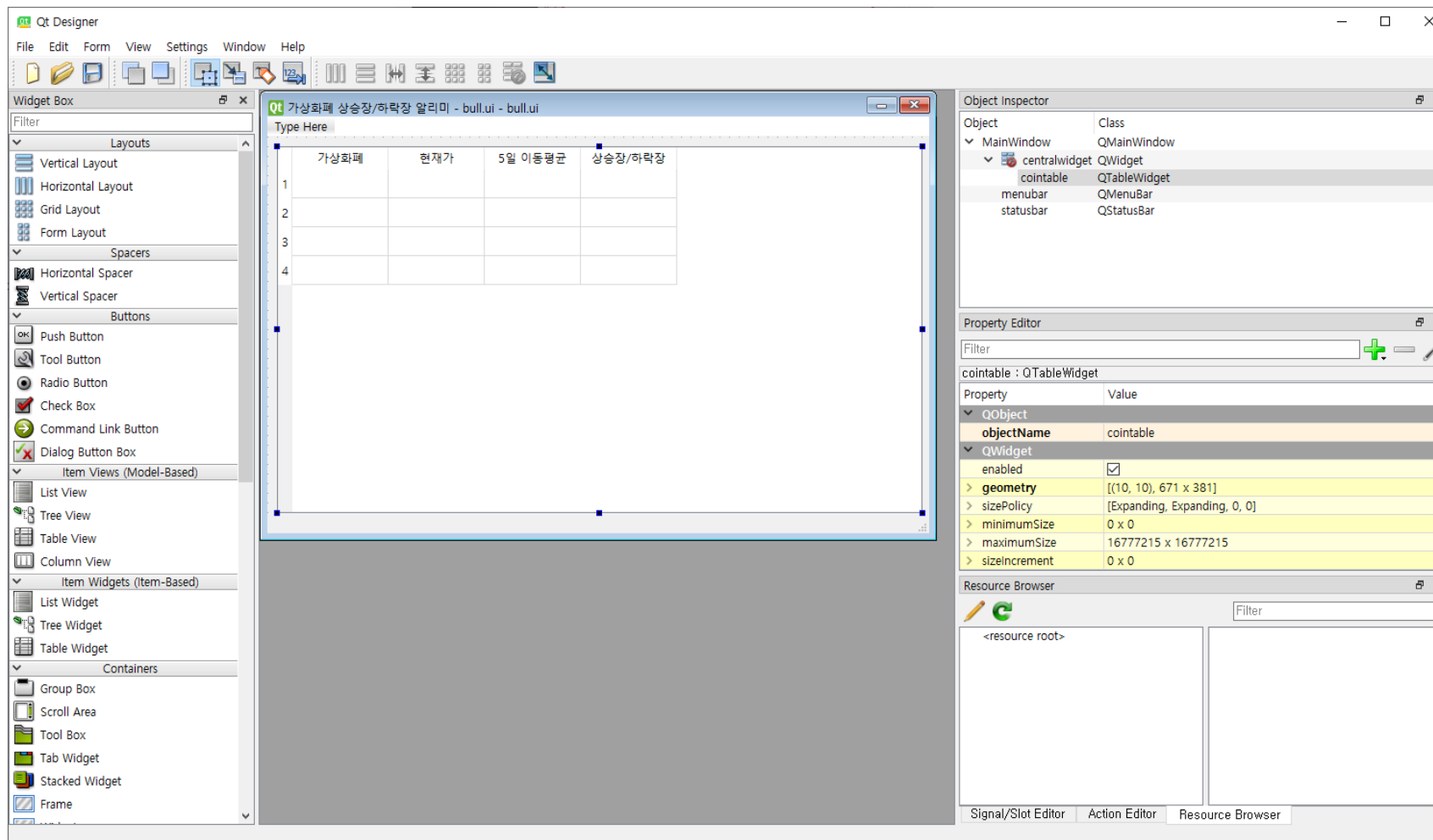
```
import pybithumb
```

```
df = pybithumb.get_ohlcv("BTC")  
ma5 = df['close'].rolling(window=5).mean()  
last_ma5 = ma5[-2]
```

```
price = pybithumb.get_current_price("BTC")
```

```
if price > last_ma5:  
    print("상승장")  
else:  
    print("하락장")
```

PyQt5 프로그램 실습



PyQt5 프로그램 실습

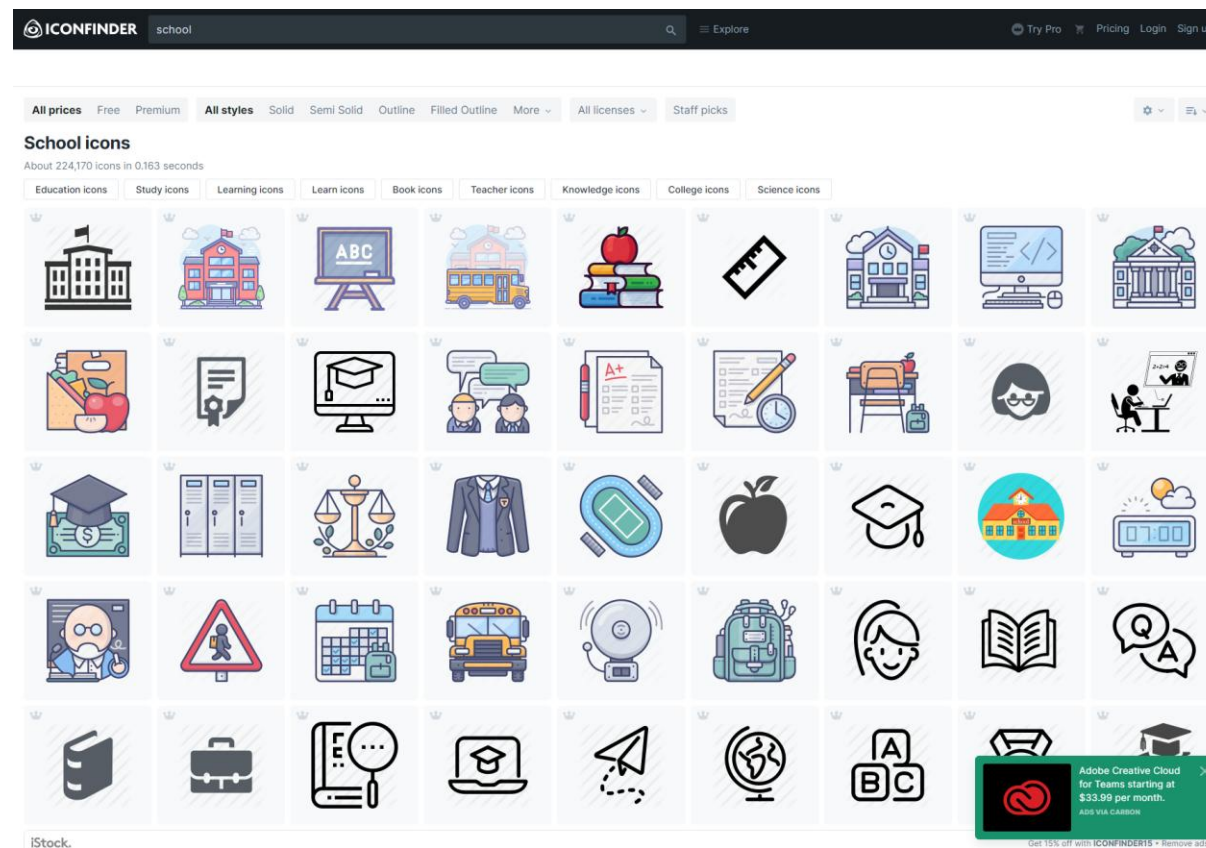


Search for icons, e.g. computer, facebook, folder...

Popular searches: [Instagram](#), [Facebook](#), [Arrow](#), [Phone](#), [Youtube](#), [Search](#), [User](#)

Search through 5,518,271 SVG or PNG icons

<https://www.iconfinder.com/>



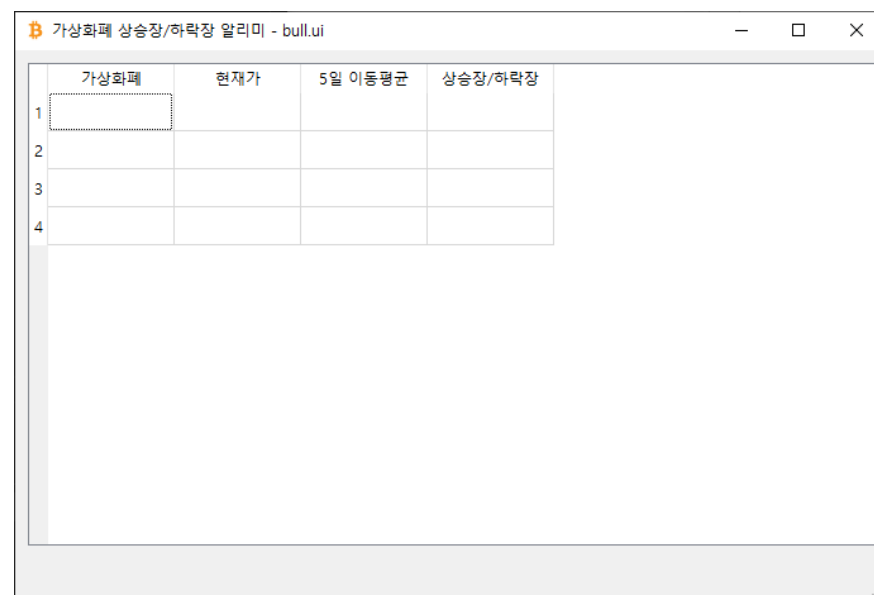
PyQt5 프로그램 실습

```
import sys
import pybithumb
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtCore import *

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

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

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



	종목명	현재가	5일 이동평균	상승/하락
삼성전자(005930)	(0,0)	(0,1)		
LG(003550)	(1,0)			
한화(000880)				
두산(000150)				
카카오(035720)				

PyQt5 프로그램 실습

```
tickers = ["BTC", "ETH", "DOGE", "XRP"]
```

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

```
class MyWindow(QMainWindow, form_class):
```

```
    def __init__(self):
```

```
        super().__init__()
```

```
        self.setupUi(self)
```

```
        timer = QTimer(self)
```

```
        timer.start(3000)
```

```
        timer.timeout.connect(self.timeout)
```

```
    def timeout(self):
```

```
        print("timer starting...")
```

```
D:\python_exam\coin>python bull.py
D:\python_exam\coin>python bull.py
D:\python_exam\coin>python bull.py
D:\python_exam\coin>python bull.py
File "D:\python_exam\coin\bull.py",
  app = QApplication(sys.argv)
IndentationError: expected an indenta
D:\python_exam\coin>python bull.py
D:\python_exam\coin>python bull.py
timer starting...
timer starting...
timer starting...
D:\python_exam\coin>python bull.py
timer starting...
timer starting...
```

가상화폐 상승장/하락장 알리미 - bull.ui			
가상화폐	현재가	5일 이동평균	상승장/하락장
1			
2			
3			
4			

PyQt5 프로그램 실습

```
def timeout(self):  
    # print("timer starting...")  
    for i, ticker in enumerate(tickers):  
        item = QTableWidgetItem(ticker)  
        self.cointable.setItem(i, 0, item)
```

IndentationError: expected an indented block

가상화폐 상승장/하락장 알리미 - bull.ui

	가상화폐	현재가	5일 이동평균	상승장/하락장
1	BTC			
2	ETH			
3	DOGE			
4	XRP			

PyQt5 프로그램 실습

```
timer.timeout.connect(self.timeout)
```

```
def get_market_infos(self, ticker):  
    df = pybithumb.get_ohlcvticker)  
    ma5 = df['close'].rolling(window=5).mean()  
    last_ma5 = ma5[-2]  
    price = pybithumb.get_current_price(ticker)
```

```
state = "  
if price > last_ma5:  
    state = "상승장"  
else:  
    state = "하락장"
```

```
return price, last_ma5, state
```

PyQt5 프로그램 실습

```
def timeout(self):  
    for i, ticker in enumerate(tickers):  
        item = QTableWidgetItem(ticker)  
        self.cointable.setItem(i, 0, item)  
  
        price, last_ma5, state = self.get_market_infos(ticker)  
        self.cointable.setItem(i, 1, QTableWidgetItem(str(price)))  
        self.cointable.setItem(i, 2, QTableWidgetItem(str(last_ma5)))  
        self.cointable.setItem(i, 3, QTableWidgetItem(str(state)))
```

PyQt5 프로그램 실습



<주의사항>

3.8 버전이하는 API 설치

3.9 버전에서는 기본 내장되어 있음.

<모듈설치하기>

- pip install pyinstaller
- pip install -U pyinstaller

PyQt5 프로그램 실습

경로가 있는 위치로 이동

➤ `python -m PyInstaller -w -F 파일이름.py`

를 입력하면 자동으로 컴파일 하여 실행파일을 만든다.

<옵션설명>








-m : 파이썬 내부에 있는 명령어 실행

-w : 콘솔창에 메시지 표시 금지

-F : 파이썬 모든 라이브러리를 실행파일에 모두 넣는다.

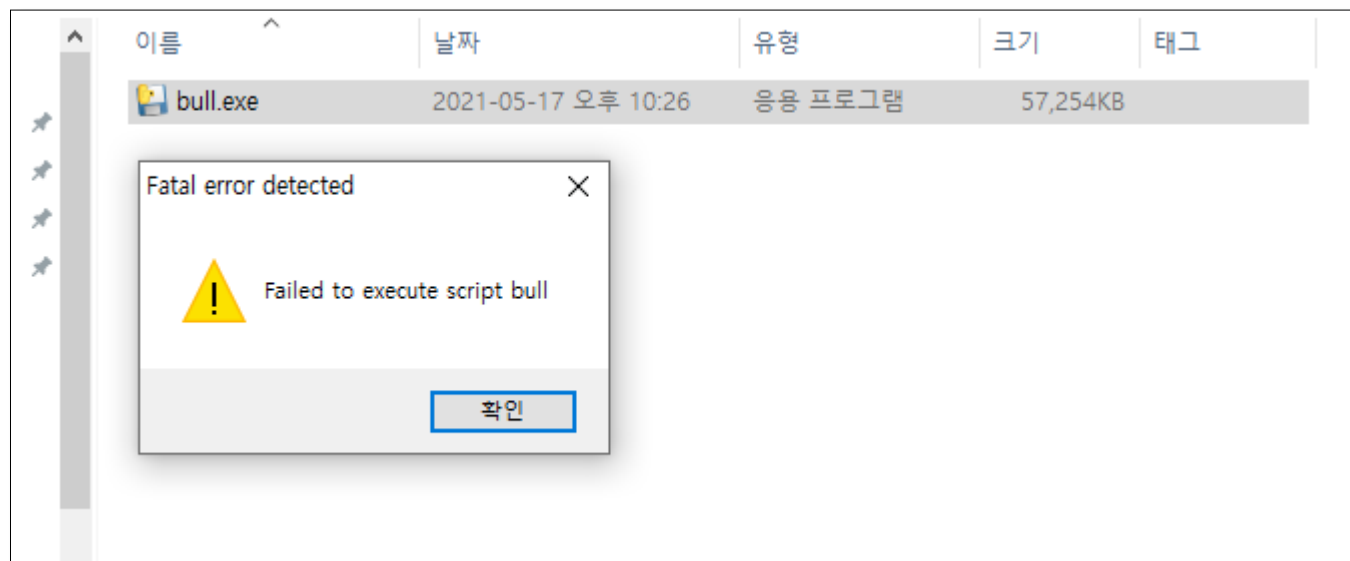
PyQt5 프로그램 실습

경로확인

	__pycache__	← 컴파일시 사용한 캐피파일	2021-05-17 오후 10:26	파일 폴더	
	build	← 컴파일 완료된 라이브러리, 정보파일	2021-05-17 오후 10:25	파일 폴더	
	dist	← 실행파일	2021-05-17 오후 10:33	파일 폴더	
	bull.py		2021-05-17 오후 9:59	Python File	1KB
	bull.spec	← 컴파일시 정보를 저장한 파일	2021-05-17 오후 10:25	SPEC 파일	1KB
	bull.ui		2021-05-17 오후 9:34	UI 파일	2KB
	coin.png		2021-05-17 오후 9:33	PNG 파일	11KB

PyQt5 프로그램 실습

실행확인



에러발생 사유
외부에서 사용한 파일들을 모두 복사하여 주어야 함.
(ex: ui, icon등)

PyQt5 프로그램 실습

다시 실행확인



The screenshot shows a file explorer window with the following files:

이름	날짜	유형	크기	태그
bull.exe	2021-05-17 오후 10:47	응용 프로그램	57,257KB	
bull.ui	2021-05-17 오후 9:34	UI 파일	2KB	
coin.png	2021-05-17 오후 9:33	PNG 파일	11KB	

Below the file explorer, a PyQt5 application window titled "가상화폐 상승장/하락장 알리미 - bull.ui" is shown. It contains a table with the following data:

	가상화폐	현재가	5일 이동평균	상승장/하락장
1	BTC	54244000.0	62770000.0	하락장
2	ETH	4151000.0	4878200.0	하락장
3	DOGE	600.7	nan	하락장
4	XRP	1821.0	1764.6	상승장

PyQt5 프로그램 실습

시그널(Signal)과 슬롯(Slot)

PyQt5 프로그램 실습

```
import sys
import pybithumb
from PyQt5.QtWidgets import *
from PyQt5 import uic
from PyQt5.QtCore import *

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

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

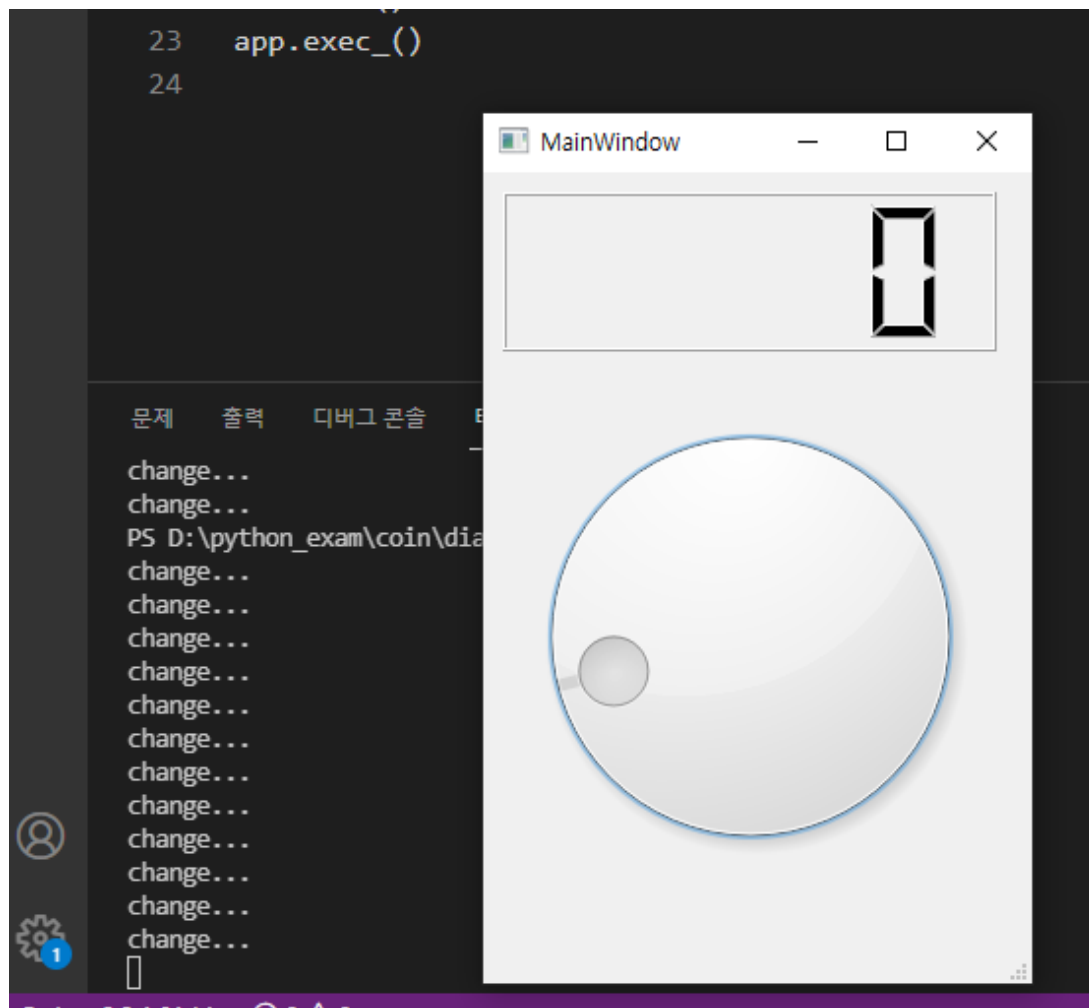
app = QApplication(sys.argv)
win = MyWindow()
win.show()
app.exec_()
```



PyQt5 프로그램 실습

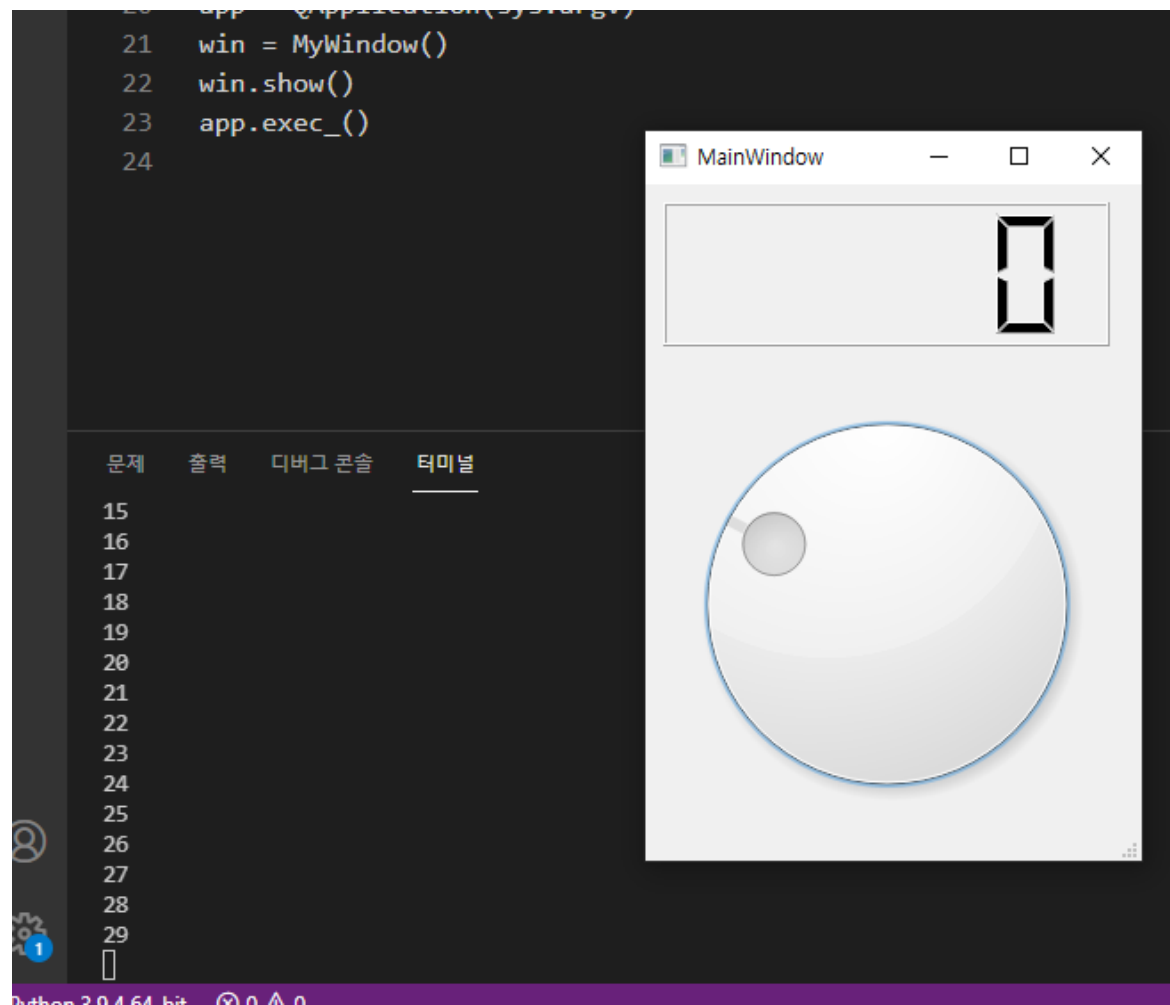
```
class MyWindow(QMainWindow, form_class):  
    def __init__(self):  
        super().__init__()  
        self.setupUi(self)  
  
        self.dial.valueChanged.connect(self.dial_changed)  
  
    def dial_changed(self):  
        print("change...")
```

PyQt5 프로그램 실습



PyQt5 프로그램 실습

```
def dial_changed(self):  
    print(self.dial.value())
```



PyQt5 프로그램 실습

```
self.dial.valueChanged.connect(self.dial_changed)
```

```
def dial_changed(self):  
    #print(self.dial.value())  
    self.lcd.display(self.dial.value())
```

- ⇒ 위의 방식은 이벤트를 만들어서 연결하는 방식임.
- ⇒ 시그널과 슬롯연결방식은 아님.



PyQt5 프로그램 실습

```
self.dial.valueChanged.connect(self.dial_changed)
```

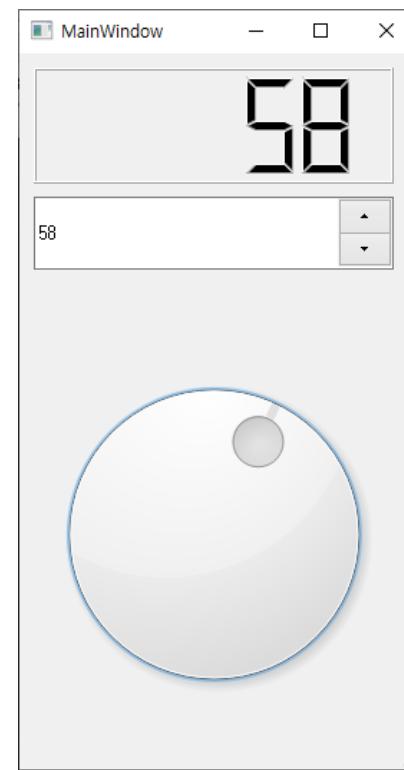
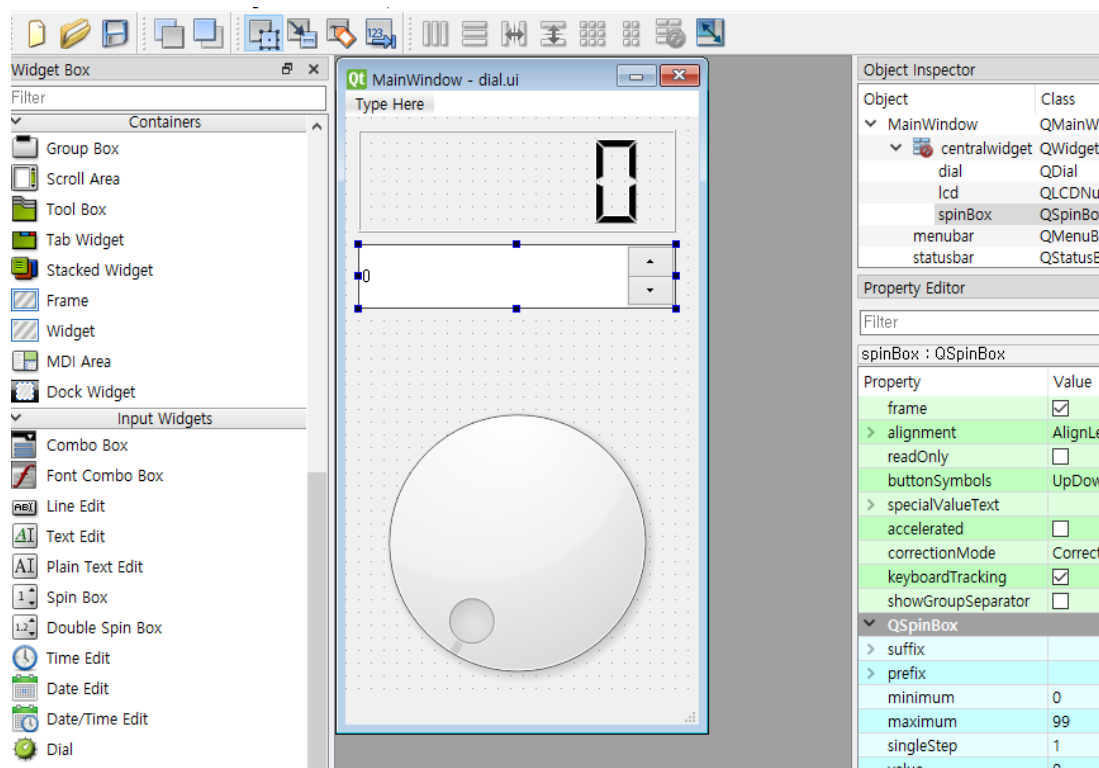
```
def dial_changed(self):  
    #print(self.dial.value())  
    # self.lcd.display(self.dial.value())
```

```
self.dial.valueChanged.connect(self.lcd.display)
```



PyQt5 프로그램 실습

```
self.dial.valueChanged.connect(self.lcd.display)  
self.dial.valueChanged.connect(self.spinBox.setValue)
```



2025년 서울 민간기업 맞춤형 매력일자리 사업

파이썬 GUI 프로그램 개발(PyQt5)

- 2025.05.02 -