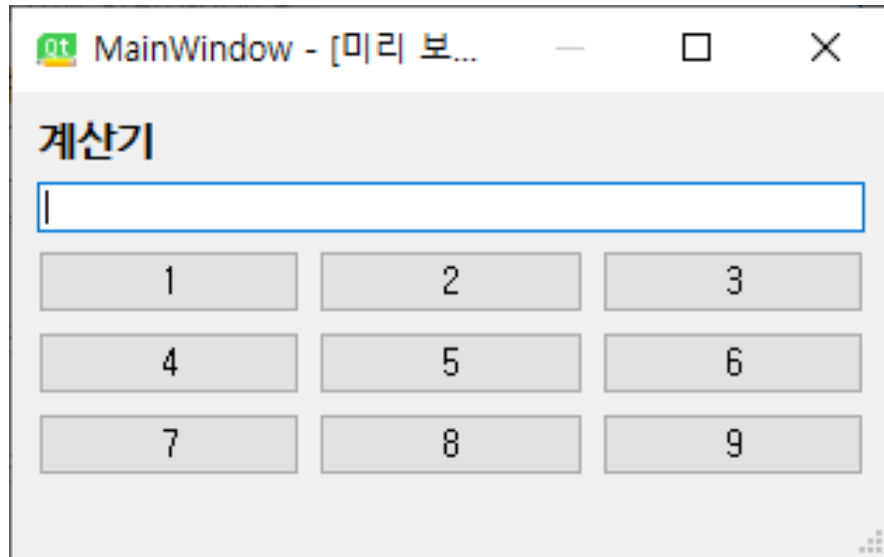


QtDesigner

PyQT를 활용한 GUI 프로그래밍

GUI applications (1/2)

- Hand Coding
 - 파이썬 코드로 GUI를 제어할 수 있음

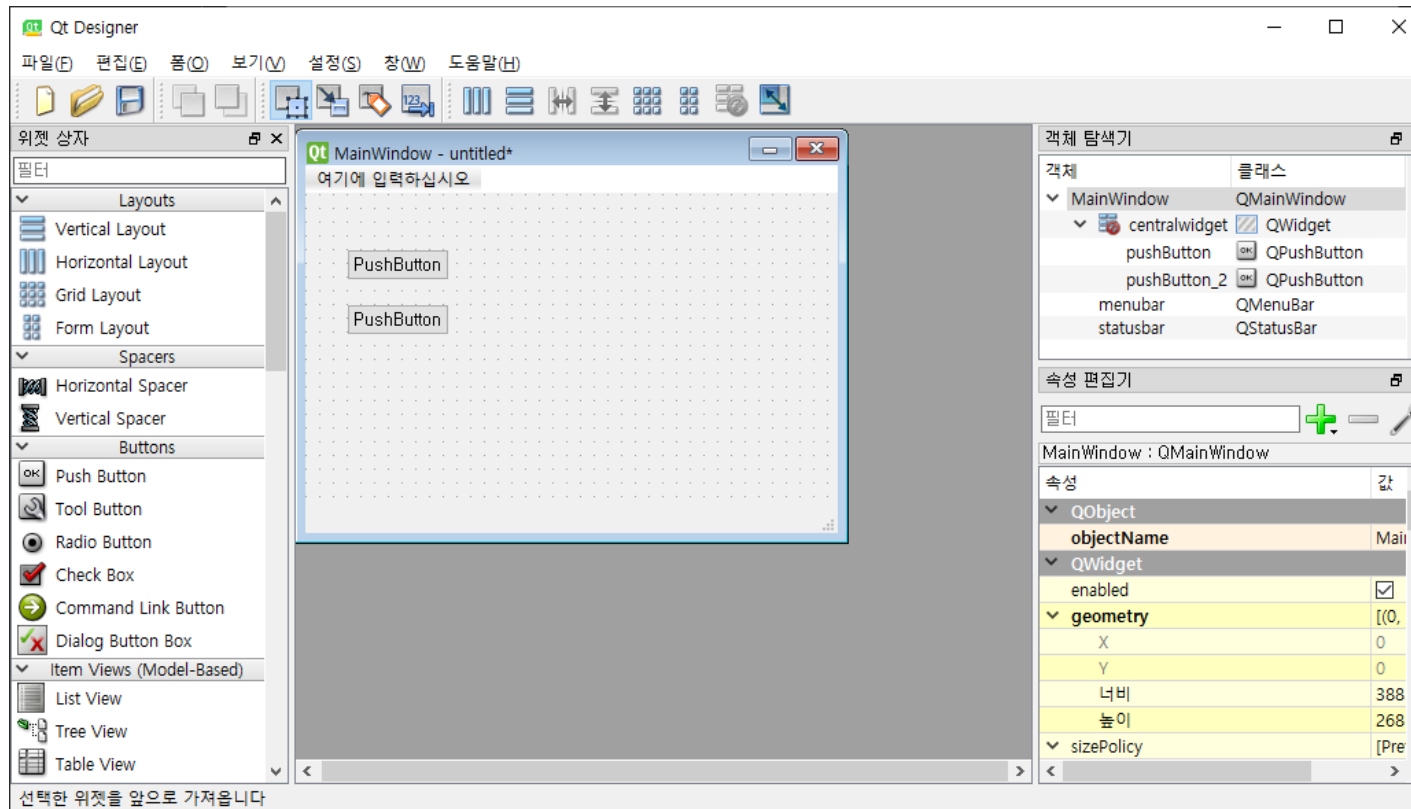


```
self.btn0 = QPushButton("1", self)
self.btn0.move(10, 10)
self.btn1 = QPushButton("2", self)
self.btn1.move(10, 50)
self.btn2 = QPushButton("3", self)
self.btn2.move(10, 90)
```

....

GUI applications (2/2)

- 그리면서 GUI를 배치하는 Qt Designer



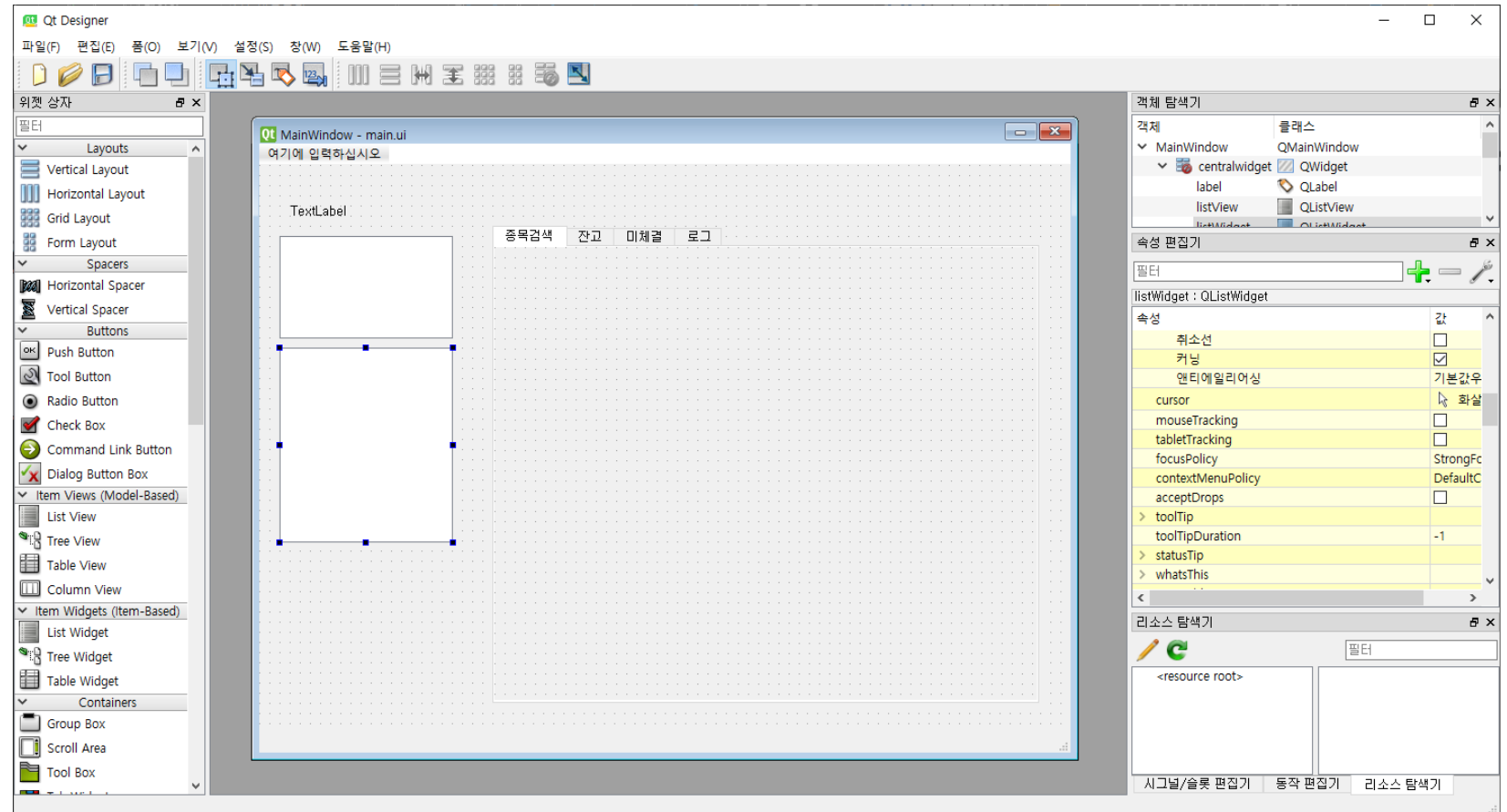
GUI 개발 방법 정리

- Hand Coding과 Qt Designer

구 분	Qt Designer	Hand Coding
생산성	High	Low
GUI와 로직의 분리	High	Low
GUI 컴포넌트의 제어	Low	High
위젯의 동적인 수정	Low	High
Custom Widget	Low	High

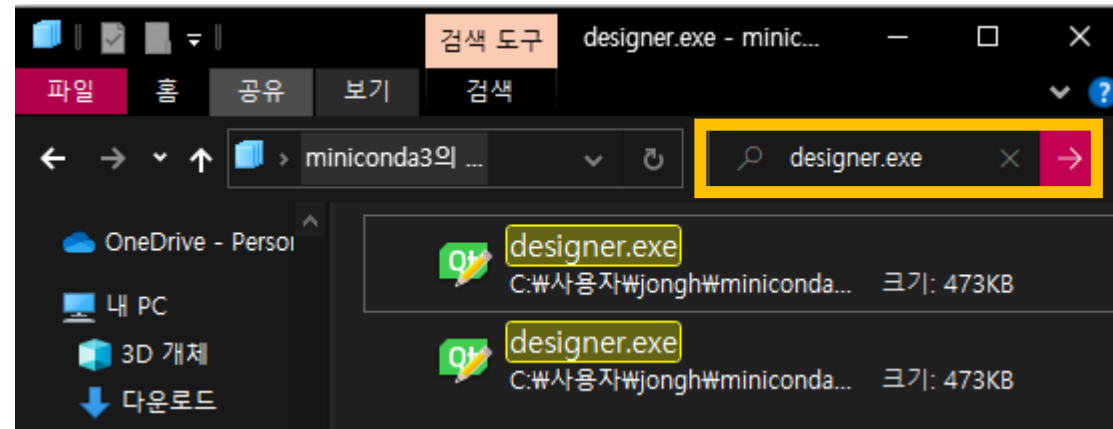
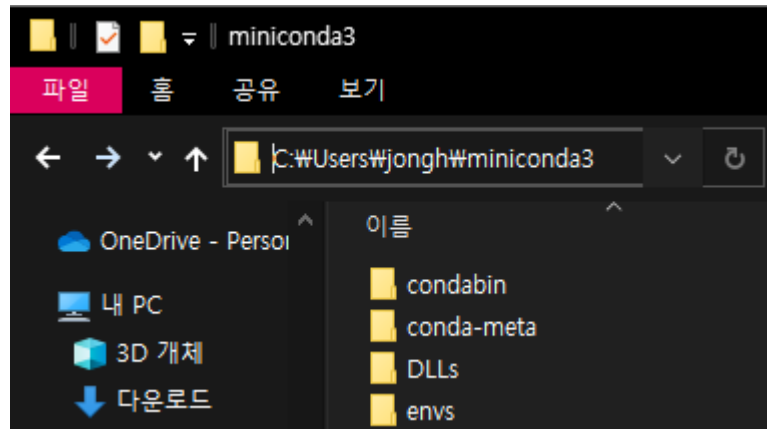
QT Designer란?

- 위젯의 배치를 도와주는 툴
 - 그리듯이 위젯을 배치



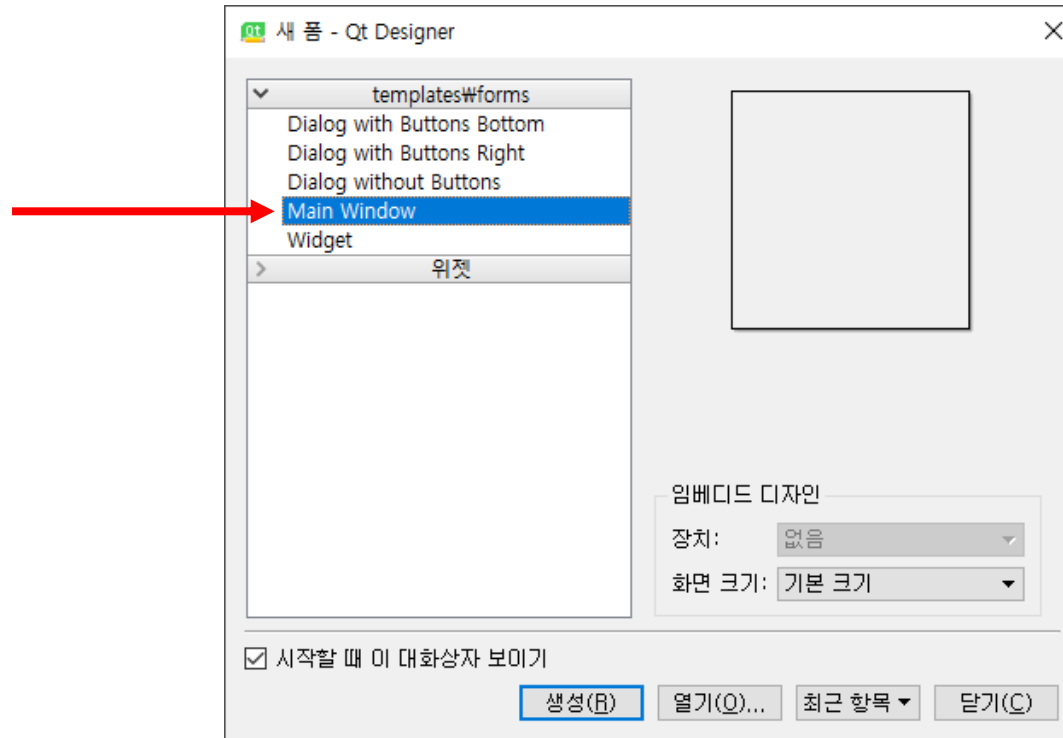
QT Designer 실행 파일

- 아나콘다 설치 경로에서 designer.exe 검색
 - 검색되지 않는 경우 pyside2 설치 후 검색



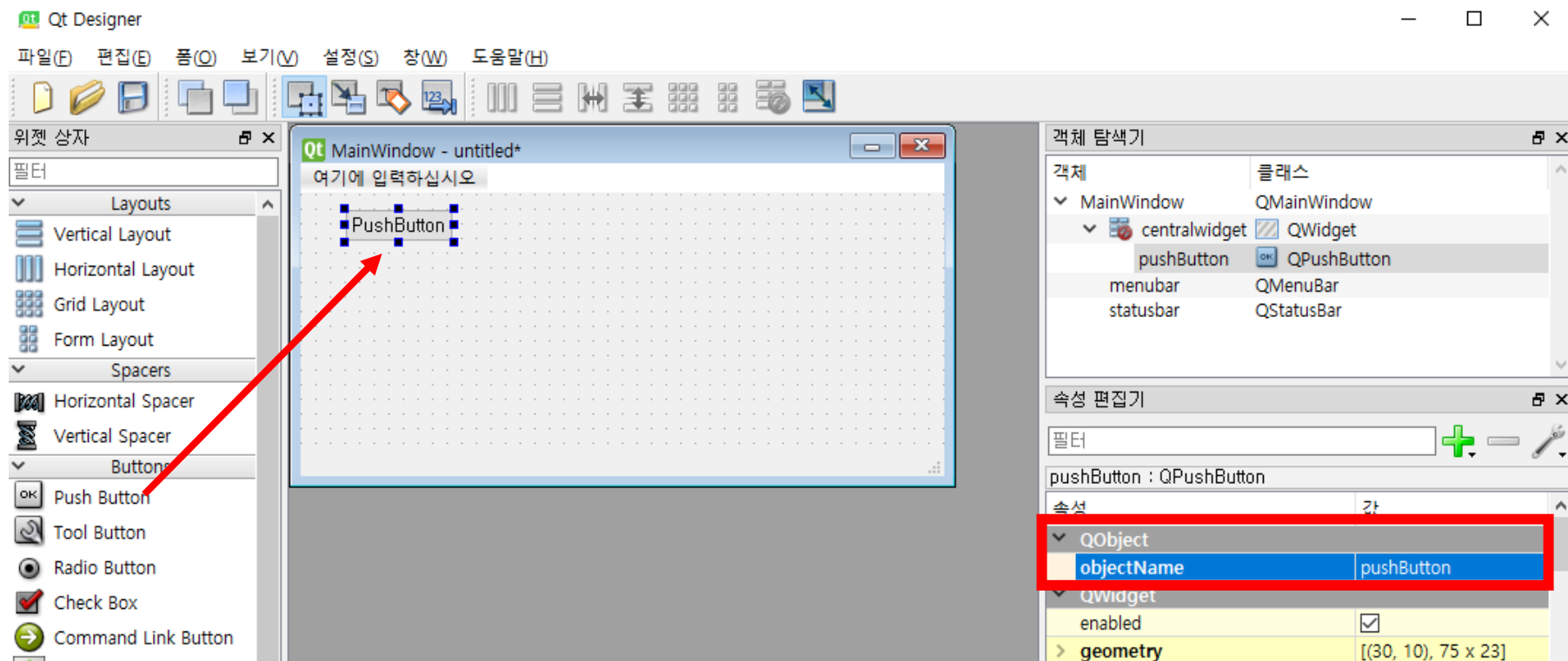
Main Window

- PyQt의 최상위 위젯



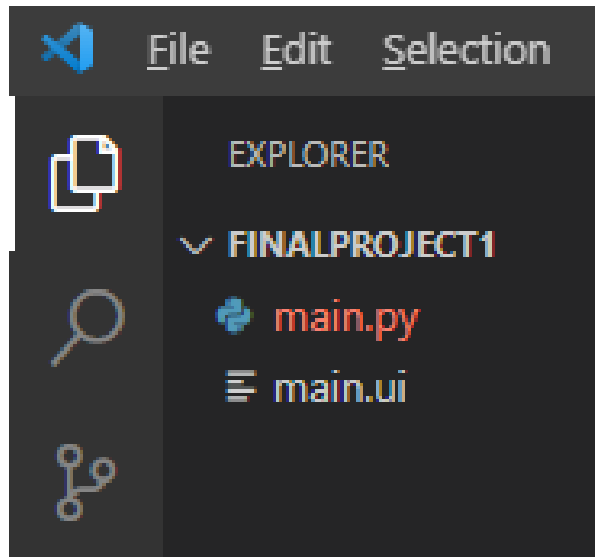
위젯 배치

- 드래그&드롭으로 직관적인 사용



Python과 UI 연결 (1/3)

- Designer로 작성한 파일을 python 파일과 같은 경로에 저장
 - 다른 위치에 파일을 저장해도 되지만 편의상 지정함



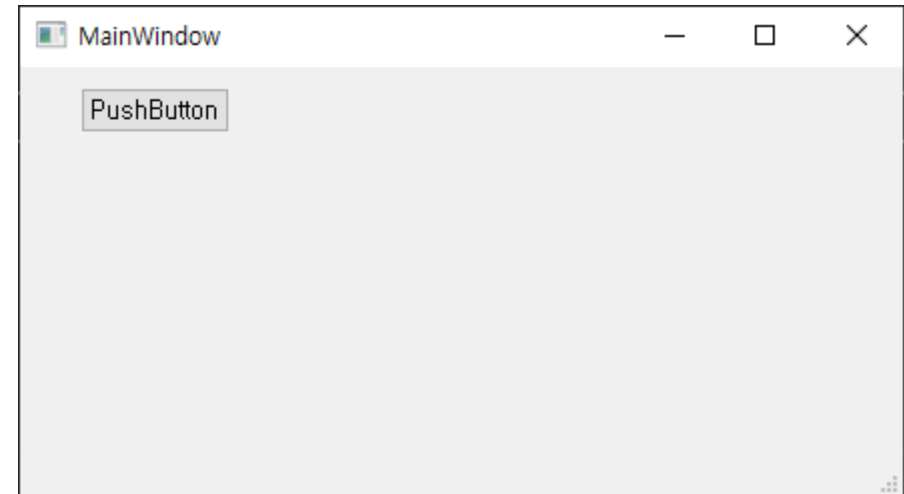
Python과 UI 연결 (2/3)

- 디자이너와 연결하는 UIC 모듈

```
from PyQt5.QtWidgets import QMainWindow, QApplication  
from PyQt5 import uic
```

```
class MyWin(QMainWindow):  
    def __init__(self):  
        super().__init__()  
        uic.loadUi("test.ui", self)
```

```
if __name__ == "__main__":  
    app = QApplication([])  
    m = MyWin()  
    m.show()  
    app.exec_()
```



Python과 UI 연결 (3/3)

- 이벤트는 파이썬에서 연결하기

```
from PyQt5.QtWidgets import QMainWindow,  
QApplication  
from PyQt5 import uic
```

```
class MyWin(QMainWindow):  
    def __init__(self):  
        super().__init__()  
        uic.loadUi("test.ui", self)
```

```
self.pushButton.clicked.connect(self.func)
```

```
def func(self):  
    print("click 됨")
```