

# 버튼과 레이블

PyQT를 활용한 GUI 프로그래밍

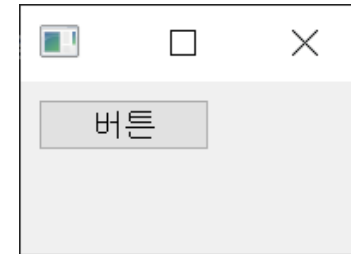
# 버튼 만들기

- 버튼을 표현하는 QPushButton
  - parent에는 위젯이 연결될 위치를 지정

```
import sys
from PyQt5.QtWidgets import *

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

        button = QPushButton(text="버튼", parent=self)
        button.move(10, 10)
```



# 버튼 크기 변경하기

- QPushButton 객체의 resize 메서드
  - 너비 (width)와 높이 (height)를 차례로 입력

```
import sys
from PyQt5.QtWidgets import *

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

        button = QPushButton(text="버튼", parent=self)
        button.resize(150, 30)
        button.move(10, 10)
```

# 버튼 클릭 이벤트 처리 (1/2)

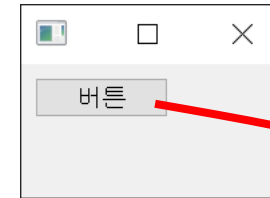
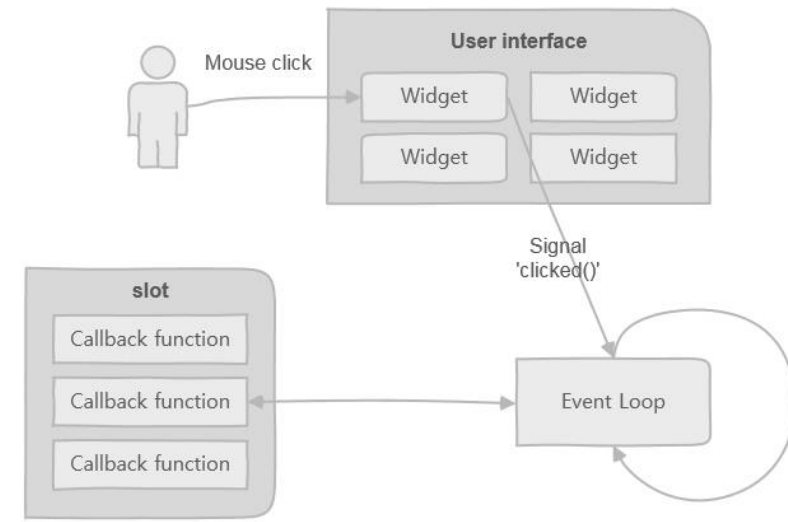
- 버튼 클릭시 호출할 함수(메소드) 연결
  - `btn.clicked.connect(함수)`

```
import sys
from PyQt5.QtWidgets import *

def test():
    print("button clicked")

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

        button = QPushButton(text="버튼", parent=self)
        button.move(10, 10)
        button.clicked.connect(test)
```



```
button clicked
button clicked
button clicked
button clicked
```

# 버튼 클릭 이벤트 처리 (2/2)

- 함수를 클래스의 메소드로 변경 (같은 결과)

```
import sys
from PyQt5.QtWidgets import *

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

        button = QPushButton(text="버튼", parent=self)
        button.move(10, 10)
        button.clicked.connect(self.test)

    def test(self):
        print("button clicked")
```

# 연습 문제 - 1

- 그림과 같이 버튼을 배치하라.



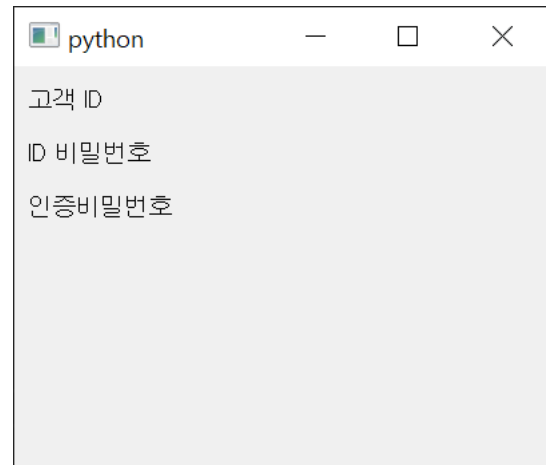
# 레이블 위젯 생성

- QLabel은 간단한 텍스트를 출력할 때 사용

```
class MyWindow(QMainWindow):  
    def __init__(self):  
        super().__init__()  
        self.setGeometry(300, 300, 400, 300)  
        label1 = QLabel(text="고객 ID", parent=self)  
        label1.resize(150, 30)  
        label1.move(10, 10)  
  
        label2 = QLabel(text="ID 비밀번호", parent=self)  
        label2.resize(150, 30)  
        label2.move(10, 50)  
  
        label3 = QLabel(text="인증비밀번호", parent=self)  
        label3.resize(150, 30)  
        label3.move(10, 90)
```



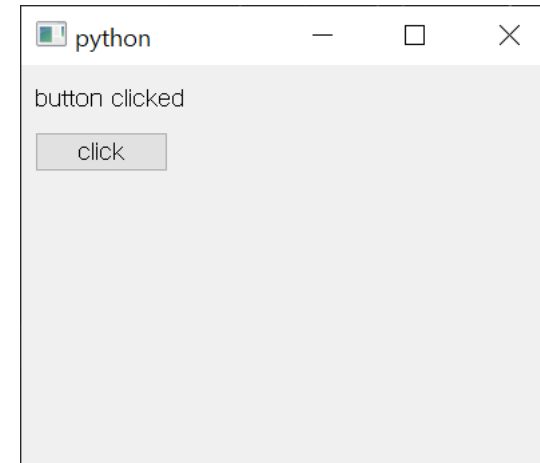
A login form with three input fields and three buttons. The first input field is labeled '고객 ID' and contains five asterisks. The second input field is labeled 'ID 비밀번호' and is empty. The third input field is labeled '인증비밀번호' and is empty. Below the input fields are three buttons: '로그인' (Login) in red, '접속환경설정' (Connection Environment Settings) in dark gray, and '종료' (End) in dark gray.



# 레이블 메소드

- 버튼 클릭 시 메시지를 QLabel에 출력

```
class MyWindow(QMainWindow):  
    def __init__(self):  
        super().__init__()  
        self.setGeometry(300, 300, 400, 300)  
  
        self.label1 = QLabel(self)  
        self.label1.resize(300, 30)  
        self.label1.move(10, 10)  
  
        self.btn1 = QPushButton("click", self)  
        self.btn1.move(10, 50)  
        self.btn1.clicked.connect(self.btn1_clicked)  
  
    def btn1_clicked(self):  
        self.label1.setText("button clicked")
```



다른 메소드에서 생성자에서 생성한 QLabel 객체에 접근해야 하므로 변수 이름 앞에 self를 붙여야 함.



# 연습 문제 - 2

- QLabel 객체에 숫자 100을 출력하라.

# 연습 문제 - 3

- QLabel 객체에 1부터 10까지 순차 증가하는 숫자를 출력하라.