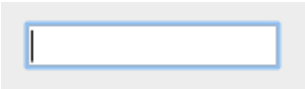


# QLineEdit



QLineEdit은 한 줄의 문자열을 입력하고 수정할 수 있도록 하는 위젯입니다.

echoMode()를 설정함으로써 ‘쓰기 전용’ 영역으로 사용할 수 있습니다. 이 기능은 비밀번호와 같은 입력을 받을 때 유용하게 사용됩니다.

setEchoMode() 메서드로 이러한 모드를 설정할 수 있으며, 입력값과 기능은 아래 표와 같습니다.

Normal 모드를 가장 흔하게 사용하며, 기본 설정값이기도 합니다. (예시: `setEchoMode(QLineEdit.Normal)` 또는 `setEchoMode(0)`)

상수	값	설명
<code>QLineEdit.Normal</code>	0	입력된 문자를 표시합니다. (기본값)
<code>QLineEdit.NoEcho</code>	1	문자열을 표시하지 않습니다. 이 설정은 비밀번호의 글자수도 공개하지 않을 때 유용합니다.
<code>QLineEdit.Password</code>	2	입력된 문자 대신 비밀번호 가림용 문자를 표시합니다.
<code>QLineEdit.PasswordEchoOnEdit</code>	3	입력할 때만 문자를 표시하고, 수정 중에는 다른 문자를 표시합니다.

maxLength() 메서드로 입력되는 텍스트의 길이를 제한할 수 있고, setValidator() 메서드로 입력되는 텍스트의 종류를 제한할 수도 있습니다.

setText() 또는 insert() 메서드로, 텍스트를 편집할 수 있고, text() 메서드로 입력된 텍스트를 가져올 수 있습니다. 만약 echoMode에 의해 입력되는 텍스트와 표시되는 텍스트가 다르다면, displayText() 메서드로 표시되는 텍스트를 가져올 수도 있습니다.

setSelection(), selectAll() 메서드로 텍스트를 선택하거나, cut(), copy(), paste() 메서드를 통해 잘라내기, 복사하기, 붙여넣기 등의 동작을 수행할 수 있습니다. 또한 setAlignment() 메서드로 텍스트의 정렬을 설정할 수 있습니다.

텍스트가 변경되거나 커서가 움직일 때, `textChanged()`, `cursorPositionChanged()`와 같은 시그널이 발생합니다.

자주 사용되는 시그널은 아래와 같습니다.

시그널	설명
<code>cursorPositionChanged()</code>	커서가 움직일 때 발생하는 신호를 발생합니다.
<code>editingFinished()</code>	편집이 끝났을 때 (Return/Enter 버튼이 눌릴 때) 신호를 발생합니다.
<code>returnPressed()</code>	Return/Enter 버튼이 눌릴 때 신호를 발생합니다.
<code>selectionChanged()</code>	선택 영역이 바뀔 때 신호를 발생합니다.
<code>textChanged()</code>	텍스트가 변경될 때 신호를 발생합니다.
<code>textEdited()</code>	텍스트가 편집될 때 신호를 발생합니다.

## 예제

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QLineEdit

class MyApp(QWidget):

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

    def initUI(self):
        self.lbl = QLabel(self)
        self.lbl.move(60, 40)

        qle = QLineEdit(self)
        qle.move(60, 100)
        qle.textChanged[str].connect(self.onChanged)

        self.setWindowTitle('QLineEdit')
        self.setGeometry(300, 300, 300, 200)
        self.show()
```

```
def onChanged(self, text):
    self.lbl.setText(text)
    self.lbl.adjustSize()

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

위젯에 하나의 라벨과 하나의 QLineEdit 위젯이 포함됩니다.

QLineEdit 위젯에서 입력되고 수정되는 텍스트가 라벨에 즉시 표시됩니다.

## 설명

---

```
qle = QLineEdit(self)
```

QLineEdit 위젯을 만들었습니다.

```
qle.textChanged[str].connect(self.onChanged)
```

qle의 텍스트가 바뀌면, onChanged() 메서드를 호출합니다.

```
def onChanged(self, text):

    self.lbl.setText(text)
    self.lbl.adjustSize()
```

onChanged() 메서드 안에서, 입력된 'text'를 라벨 위젯(lbl)의 텍스트로 설정하도록 합니다.

또한 adjustSize() 메서드로 텍스트의 길이에 따라 라벨의 길이를 조절해주도록 합니다.

## 결과

---

