

# QTextEdit

---

QTextEdit 클래스는 플레인 텍스트 (plain text)와 리치 텍스트 (rich text)를 모두 편집하고 표시할 수 있는 편집기를 제공합니다.

## 예제

---

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QLabel, QTextEdit, QVBoxLayout

class MyApp(QWidget):

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

    def initUI(self):
        self.lbl1 = QLabel('Enter your sentence:')
        self.te = QTextEdit()
        self.te.setAcceptRichText(False)
        self.lbl2 = QLabel('The number of words is 0')

        self.te.textChanged.connect(self.text_changed)

        vbox = QVBoxLayout()
        vbox.addWidget(self.lbl1)
        vbox.addWidget(self.te)
        vbox.addWidget(self.lbl2)
        vbox.addStretch()

        self.setLayout(vbox)

        self.setWindowTitle('QTextEdit')
        self.setGeometry(300, 300, 300, 200)
        self.show()

    def text_changed(self):
        text = self.te.toPlainText()
        self.lbl2.setText('The number of words is ' + str(len(text.split())))
```

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

텍스트 편집기에 텍스트를 입력하면, 아래에 단어수를 표시합니다.

## 설명

---

```
self.lbl1 = QLabel('Enter your sentence:')  
self.te = QTextEdit()  
self.te.setAcceptRichText(False)  
self.lbl2 = QLabel('The number of words is 0')
```

QTextEdit() 클래스를 이용해서 텍스트 편집기 하나를 만들었습니다.

setAcceptRichText를 False로 하면, 모두 플레인 텍스트로 인식합니다.

아래의 라벨은 단어수를 표시합니다.

```
self.te.textChanged.connect(self.text_changed)
```

텍스트 편집기의 텍스트가 수정될 때마다 text\_changed 메서드가 호출됩니다.

```
self.clear_btn = QPushButton('Clear')  
self.clear_btn.pressed.connect(self.clear_text)
```

clear\_btn을 클릭하면, clear\_text 메서드가 호출됩니다.

```
vbox = QVBoxLayout()  
vbox.addWidget(self.lbl1)  
vbox.addWidget(self.te)  
vbox.addWidget(self.lbl2)  
vbox.addStretch()  
  
self.setLayout(vbox)
```

수직 박스 레이아웃을 이용해서, 두 개의 라벨과 하나의 텍스트 편집기를 수직 방향으로 배치합니다.

```
def text_changed(self):  
  
    text = self.te.toPlainText()  
    self.lbl2.setText('The number of words is ' + str(len(text.split())))
```

text\_changed 메서드가 호출되면, toPlainText() 메서드를 이용해서 텍스트 편집기 (self.te)에 있던 텍스트를 text 변수에 저장합니다.

split()은 문자열의 단어들을 리스트 형태로 바꿔줍니다.

len(text.split())은 text의 단어수입니다.

setText()를 이용해서 두 번째 라벨에 단어수를 표시합니다.

## 결과

---

