

QSlider & QDial



QSlider는 수평 또는 수직 방향의 슬라이더를 제공합니다.

슬라이더는 제한된 범위 안에서 값을 조절하는 위젯입니다.



슬라이더의 틱(tick)의 간격을 조절하기 위해서는 `setTickInterval()` 메서드, 틱(tick)의 위치를 조절하기 위해서는 `setTickPosition()` 메서드를 사용합니다.

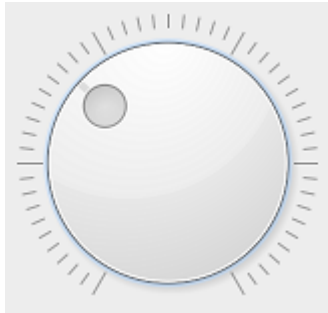
`setTickInterval()` 메서드의 입력값은 픽셀이 아니라 값을 의미합니다.

`setTickPosition()` 메서드의 입력값과 기능은 아래의 표와 같습니다. (예시:
`setTickPosition(QSlider.NoTicks)` 또는 `setTickPosition(0)`)

상수	값	설명
<code>QSlider.NoTicks</code>	0	틱을 표시하지 않습니다.
<code>QSlider.TicksAbove</code>	1	틱을 (수평) 슬라이더 위쪽에 표시합니다.
<code>QSlider.TicksBelow</code>	2	틱을 (수평) 슬라이더 아래쪽에 표시합니다.
<code>QSlider.TicksBothSides</code>	3	틱을 (수평) 슬라이더 양쪽에 표시합니다.
<code>QSlider.TicksLeft</code>	<code>TicksAbove</code>	틱을 (수직) 슬라이더 왼쪽에 표시합니다.
<code>QSlider.TicksRight</code>	<code>TicksBelow</code>	틱을 (수직) 슬라이더 오른쪽에 표시합니다.



QDial은 슬라이더를 둥근 형태로 표현한 다이얼 위젯이며, 기본적으로 같은 시그널과 슬롯, 메서드들을 공유합니다. ([QDial 공식 문서](#) 참고)



위의 그림처럼 다이얼 위젯에 노치(notch)를 표시하기 위해서는 `setNotchesVisible()` 메서드를 사용합니다. `True`로 설정하면 둥근 다이얼을 따라서 노치들이 표시됩니다. 기본적으로 노치는 표시되지 않도록 설정되어 있습니다.

QSlider과 QDial 위젯에서 가장 자주 쓰이는 시그널은 아래와 같습니다. 예제에서는 `valueChanged` 시그널을 사용해보겠습니다.

시그널	설명
<code>valueChanged()</code>	슬라이더의 값이 변할 때 발생합니다.
<code>sliderPressed()</code>	사용자가 슬라이더를 움직이기 시작할 때 발생합니다.
<code>sliderMoved()</code>	사용자가 슬라이더를 움직이면 발생합니다.
<code>sliderReleased()</code>	사용자가 슬라이더를 놓을 때 발생합니다.

예제

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QSlider, QDial, QPushButton
from PyQt5.QtCore import Qt

class MyApp(QWidget):

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

    def initUI(self):
        self.slider = QSlider(Qt.Horizontal, self)
```

```

self.slider.move(30, 30)
self.slider.setRange(0, 50)
self.slider.setSingleStep(2)

self.dial = QDial(self)
self.dial.move(30, 50)
self.dial.setRange(0, 50)

btn = QPushButton('Default', self)
btn.move(35, 160)

self.slider.valueChanged.connect(self.dial.setValue)
self.dial.valueChanged.connect(self.slider.setValue)
btn.clicked.connect(self.button_clicked)

self.setWindowTitle('QSlider and QDial')
self.setGeometry(300, 300, 400, 200)
self.show()

def button_clicked(self):
    self.slider.setValue(0)
    self.dial.setValue(0)

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

```

슬라이더, 다이얼 위젯과 푸시 버튼을 하나 만들었습니다.

슬라이더와 다이얼은 서로 연결되어 있어서 항상 같은 값을 가지고, 푸시 버튼을 누르면 두 위젯의 값이 모두 0이 됩니다.

설명

```
self.slider = QSlider(Qt.Horizontal, self)
```

QSlider 위젯을 하나 만들었습니다. Qt.Horizontal 또는 Qt.Vertical을 입력해줌으로써 수평 또는 수직 방향을 설정합니다.

```
self.slider.setRange(0, 50)
self.slider.setSingleStep(2)
```

setRange() 메서드로 값의 범위를 0에서 50까지로 설정합니다. setSingleStep() 메서드는 조절 가능한 최소 단위를 설정합니다.

```
self.dial = QDial(self)
```

QDial 위젯을 하나 만들었습니다.

```
self.dial.setRange(0, 50)
```

슬라이더와 마찬가지로 setRange() 메서드로 범위를 정해줍니다.

```
self.slider.valueChanged.connect(self.dial.setValue)
self.dial.valueChanged.connect(self.slider.setValue)
```

슬라이더와 다이얼의 값이 변할 때 발생하는 시그널을 각각 다이얼과 슬라이더의 값을 조절해주는 메서드 (setValue)에 서로 연결함으로써 두 위젯의 값이 언제나 일치하도록 해줍니다.

```
btn.clicked.connect(self.button_clicked)
```

‘Default’ 푸시 버튼을 클릭하면 발생하는 시그널을 button_clicked 메서드에 연결합니다.

```
def button_clicked(self):
    self.slider.setValue(0)
    self.dial.setValue(0)
```

button_clicked() 메서드는 슬라이더와 다이얼의 값을 모두 0으로 조절합니다.

따라서 ‘Default’ 푸시 버튼을 클릭하면, 슬라이더와 다이얼의 값이 0으로 초기화됩니다.

결과

