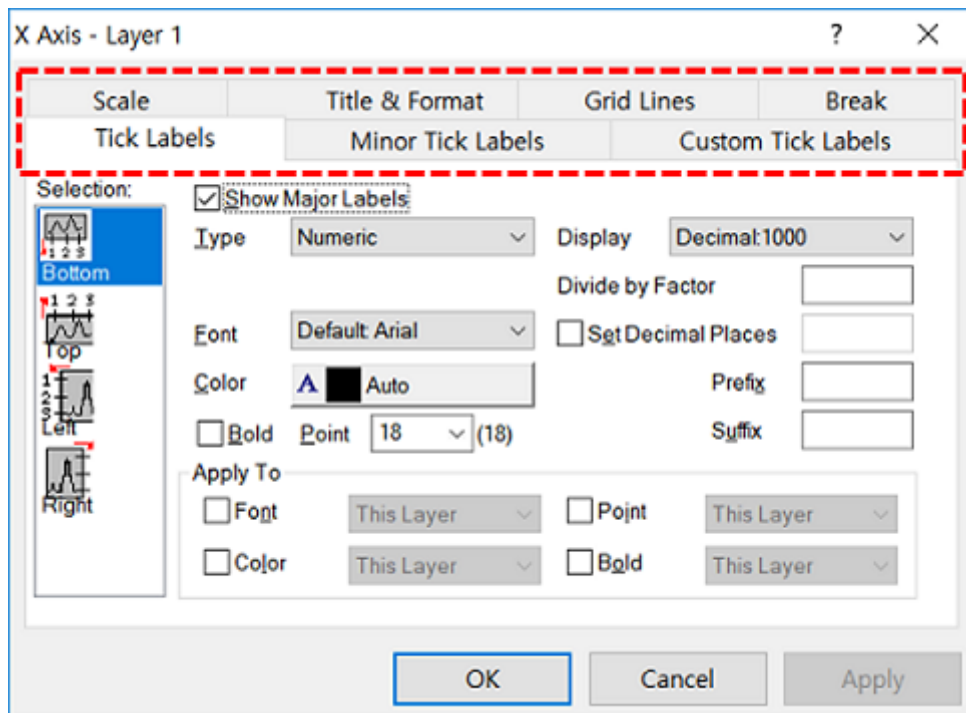


# QTabWidget



GUI 프로그램을 사용하다보면 위의 그림과 같이 탭(Tab)이 있는 창을 볼 수 있습니다.

이러한 탭은 프로그램 안의 구성요소들이 **많은 면적을 차지하지 않으면서**, 그것들을 **카테고리에 따라 분류**할 수 있기 때문에 유용하게 사용될 수 있습니다.

간단한 예제를 통해 두 개의 탭을 갖는 위젯을 하나 만들어보겠습니다.

## 예제

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QTabWidget, QVBoxLayout

class MyApp(QWidget):

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

    def initUI(self):
        tab1 = QWidget()
        tab2 = QWidget()
```

```

        tabs = QTabWidget()
        tabs.addTab(tab1, 'Tab1')
        tabs.addTab(tab2, 'Tab2')

        vbox = QVBoxLayout()
        vbox.addWidget(tabs)

        self.setLayout(vbox)

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

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

```

두 개의 탭을 갖는 작은 위젯이 하나 만들어집니다.

## 설명

---

```

tab1 = QWidget()
tab2 = QWidget()

```

각 탭에 위치할 두 개의 위젯을 만들었습니다.

```

tabs = QTabWidget()
tabs.addTab(tab1, 'Tab1')
tabs.addTab(tab2, 'Tab2')

```

QTabWidget()을 이용해서 탭을 만들어주고, addTab()을 이용해서 'Tab1'과 'Tab2'를 tabs에 추가해줍니다.

```

vbox = QVBoxLayout()
vbox.addWidget(tabs)

self.setLayout(vbox)

```

수직 박스 레이아웃을 하나 만들어서 탭 위젯 (tabs)을 넣어줍니다.

그리고 수직 박스(vbox)를 위젯의 레이아웃으로 설정합니다.

## 결과

---

