

# 박스 레이아웃

---

- 박스 레이아웃 클래스를 이용하면 훨씬 유연하고 실용적인 레이아웃을 할 수 있습니다. ([QBoxLayout 공식 문서](#) 참고)
- QHBoxLayout, QVBoxLayout은 여러 위젯을 수평으로 정렬하는 레이아웃 클래스입니다.
- QHBoxLayout, QVBoxLayout 생성자는 수평, 수직의 박스를 하나 만드는데, 다른 레이아웃 박스를 넣을 수도 있고 위젯을 배치할 수도 있습니다.
- 예제 코드에서 위젯의 가운데 아래 부분에 두 개의 버튼을 배치하기 위해 수평, 수직의 박스를 하나씩 사용합니다.
- 필요한 공간을 만들기 위해 addStretch() 메서드를 사용하고, 'stretch factor'를 조절해 보겠습니다.

```
import sys
from PyQt5.QtWidgets import QApplication, QWidget, QPushButton, QHBoxLayout

class MyApp(QWidget):

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

    def initUI(self):
        okButton = QPushButton('OK')
        cancelButton = QPushButton('Cancel')

        hbox = QHBoxLayout()
        hbox.addStretch(1)
        hbox.addWidget(okButton)
        hbox.addWidget(cancelButton)
        hbox.addStretch(1)

        vbox = QVBoxLayout()
        vbox.addStretch(3)
        vbox.addLayout(hbox)
        vbox.addStretch(1)

        self.setLayout(vbox)
```

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

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

## 설명

---

```
okButton = QPushButton('OK')
cancelButton = QPushButton('Cancel')
```

- 두 개의 버튼을 만들었습니다.

```
hbox = QHBoxLayout()
hbox.addStretch(1)
hbox.addWidget(okButton)
hbox.addWidget(cancelButton)
hbox.addStretch(1)
```

- 수평 박스를 하나 만들고, 두 개의 버튼과 양쪽에 빈 공간을 추가합니다.
- 이 `addStretch()` 메서드는 신축성있는 빈 공간을 제공합니다.
- 두 버튼 양쪽의 stretch factor가 1로 같기 때문에 이 두 빈 공간의 크기는 창의 크기가 변화해도 항상 같습니다.

```
vbox = QVBoxLayout()
vbox.addStretch(3)
vbox.addLayout(hbox)
vbox.addStretch(1)
```

- 다음으로 수평 박스(hbox)를 수직 박스(vbox)에 넣어줍니다.
- 수직 박스의 stretch factor는 수평 박스를 아래쪽으로 밀어내서 두 개의 버튼을 창의 아래쪽에 위치하도록 합니다.
- 이 때에도 수평 박스 위와 아래의 빈 공간의 크기는 항상 3:1을 유지합니다. stretch factor를 다양하게 바꿔보면, 의미를 잘 이해할 수 있습니다.