

Second Part

لایه‌بندی (Layouts) در PyQt6

در این فصل با سه مدیر لایه اصلی در PyQt6 آشنا می‌شویم: چینش عمودی/افقی با `QVBoxLayout` و `QHBoxLayout`، شبکه‌ای با `QGridLayout`، و چگونگی تو در تو کردن آن‌ها. همچنین روش‌های تنظیم فاصله (`spacing`)، حاشیه (`margins`) و تراز (`alignment`) را بررسی می‌کنیم تا رابط‌های کاربری قابل‌اعتماد و واکنش‌گرا بسازیم.

در ادامه:

- چرا باید از Layout استفاده کرد، نه موقعیت‌دهی مطلق
- معرفی `QVBoxLayout` و `QHBoxLayout` با مثال
- معرفی `QGridLayout` و قابلیت `spanning`
- تو در تو کردن Layoutها (Nesting)
- کنترل `spacing` و `margins`
- تنظیم `alignment` برای ویجت‌ها و خود Layout

چرا Layout به جای موقعیت‌دهی مطلق؟

در موقعیت‌دهی مطلق، تغییر اندازه پنجره یا فونت کاربر موجب بر هم خوردن چینش ویجت‌ها می‌شود؛ در حالی که Layout Manager براساس اندازه‌ی در دسترس، ویجت‌ها را به‌طور خودکار مرتب می‌کند و رابط شما را انعطاف‌پذیر و قابل انتقال می‌سازد. [ZetCode](#). استفاده از Layouts همچنین نگهداری کد را ساده‌تر کرده و از محاسبات دستی برای مختصات اجتناب می‌کند [Python Tutorials – Real Python](#).

۱. QVBoxLayout و QHBoxLayout

ویژگی‌ها

- `addWidget(widget, stretch=0, alignment=...)`

- `setSpacing(int)` فاصله بین آیتم‌ها Python GUIs
- `setContentsMargins(left, top, right, bottom)` حاشیه‌ی داخلی Qt Forum
- `setAlignment(widget or layout, Qt.Align...)` برای تراز دادن Stack Overflow

مثال: فرم ساده با دو دکمه در افقی و لیبل در بالا

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QVBoxLayout, QHBoxLayout
from PyQt6.QtCore import Qt

app = QApplication(sys.argv)
window = QWidget()
window.setWindowTitle('VBox & HBox Layout') # Window title

# Main vertical layout
main_layout = QVBoxLayout() # Create a vertical box layout
main_layout.setSpacing(15) # Set spacing between items
main_layout.setContentsMargins(20, 20, 20, 20) # Set margins: left, top,

# Add a centered label at top
label = QLabel('Choose an option:') # Create a label
label.setAlignment(Qt.AlignmentFlag.AlignCenter) # Center-align
main_layout.addWidget(label)

# Horizontal layout for buttons
btn_layout = QHBoxLayout() # Create a horizontal box layout
btn_layout.setSpacing(10) # Spacing between buttons :c

btn1 = QPushButton('Start') # Button 1
btn2 = QPushButton('Stop') # Button 2
btn_layout.addWidget(btn1) # Add to hbox
btn_layout.addWidget(btn2) # Add to hbox

# Add the horizontal layout into the main vertical layout
main_layout.addLayout(btn_layout) # Nesting layouts :contentRe

window.setLayout(main_layout) # Apply main layout to window
```

```
33 | window.show()
    | sys.exit(app.exec())
```

۲. QGridLayout

QGridLayout ویجت‌ها را در سلول‌های ردیف و ستون مرتب می‌کند و امکان ادغام خانه‌ها (spanning) را می‌دهد [Python TutorialTutorialsPoint](#).

متدهای کلیدی

- `addWidget(widget, row, column, rowspan=1, colspan=1)`
- `setColumnStretch(col, stretch)` و `setRowStretch(row, stretch)` برای انعطاف‌پذیری
- `setContentsMargins()` و `setSpacing()` مشابه `BoxLayout`

مثال: ماتریس دکمه‌ها با spanning

```
import sys
from PyQt6.QtWidgets import QApplication, QWidget, QGridLayout, QPushButton

app = QApplication(sys.argv)
window = QWidget()
window.setWindowTitle('Grid Layout Example')

grid = QGridLayout()                                # Create grid layout
grid.setSpacing(5)                                # Set spacing :contentReferenc

# Create buttons in a 3x3 grid
for i in range(3):
    for j in range(3):
        btn = QPushButton(f'{i},{j}')
        grid.addWidget(btn, i, j)

# Add a wide button spanning two columns at bottom
wide_btn = QPushButton('Wide Button')
grid.addWidget(wide_btn, 3, 0, 1, 2)                # row=3, col=0, rowspan=1,
```

```

21 | window.setLayout(grid)
22 | window.show()
23 | sys.exit(app.exec())

```

۳. تو در تو کردن Layout ها (Nesting)

با nesting می‌توان ساختارهای پیچیده ساخت: مثلاً یک QHBoxLayout داخل یک QVBoxLayout یا بالعکس Python GUIs. این تکنیک برای کنترل بخش‌های مجزا در پنجره بسیار مفید است.

۴. تنظیم فاصله و حاشیه

- layout.setSpacing(n) فاصله بین آیتم‌های داخل layout را تعیین می‌کند.
- layout.setContentsMargins(left, top, right, bottom) حاشیه‌ی داخلی اطراف کل آیتم‌ها را تنظیم می‌کند Qt Forum. بدون این تنظیمات، ویجت‌ها چسبیده به هم یا به دیواره پنجره خواهند بود.

۵. تراز (Alignment)

- layout.setAlignment(widget, Qt.AlignmentFlag.AlignCenter|Left|Right|Top|Bottom)
- یا layout.setAlignment(Qt.AlignHCenter) برای تراز تمام آیتم‌ها Stack Overflow. تراز مناسب

ترتیب تعیین می‌شود:

تمرین‌های پیشنهادی

۱. یک پنجره بساز که در سمت چپ یک لیست (QListWidget) و در سمت راست یک فرم (QFormLayout) داشته باشد.
۲. در یک QGridLayout سه فیلد ورودی و برچسب کنار هم قرار بده و دکمه "Submit" را در ستون وسط و ردیف پایین span کن.
۳. با setContentsMargins(0,0,0,0) و setSpacing(0) یک چیدمان فشرده بساز و تفاوت را با حالت پیش‌فرض مقایسه کن.