



Layout-Management in PyQt5

Layout-Management in PyQt5

Layout-Management in PyQt5 ist ein entscheidender Aspekt bei der Gestaltung von Benutzeroberflächen (GUIs). Es hilft dabei, Widgets in einer strukturierten und visuell ansprechenden Weise anzuordnen.

PyQt5 bietet mehrere Layout-Manager, um die Platzierung von Widgets innerhalb eines Fensters zu verwalten. Die am häufigsten verwendeten Layout-Manager sind:

QVBoxLayout: Dieses Layout ordnet Widgets **vertikal** an (von oben nach unten).

QHBoxLayout: Dieses Layout ordnet Widgets **horizontal** an (von links nach rechts).

QGridLayout: Dieses Layout ordnet Widgets in einem **Raster** (Zeilen und Spalten) an.

QFormLayout: Dieses Layout ordnet Widgets in einem **formularartigen Layout** mit Beschriftungen und Eingabefeldern an.



Layout-Management in PyQt5

QVBoxLayout arrangiert Widgets in einer vertikalen Spalte. Die Syntax lautet wie folgt:

```
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget, QVBoxLayout, QPushButton
```

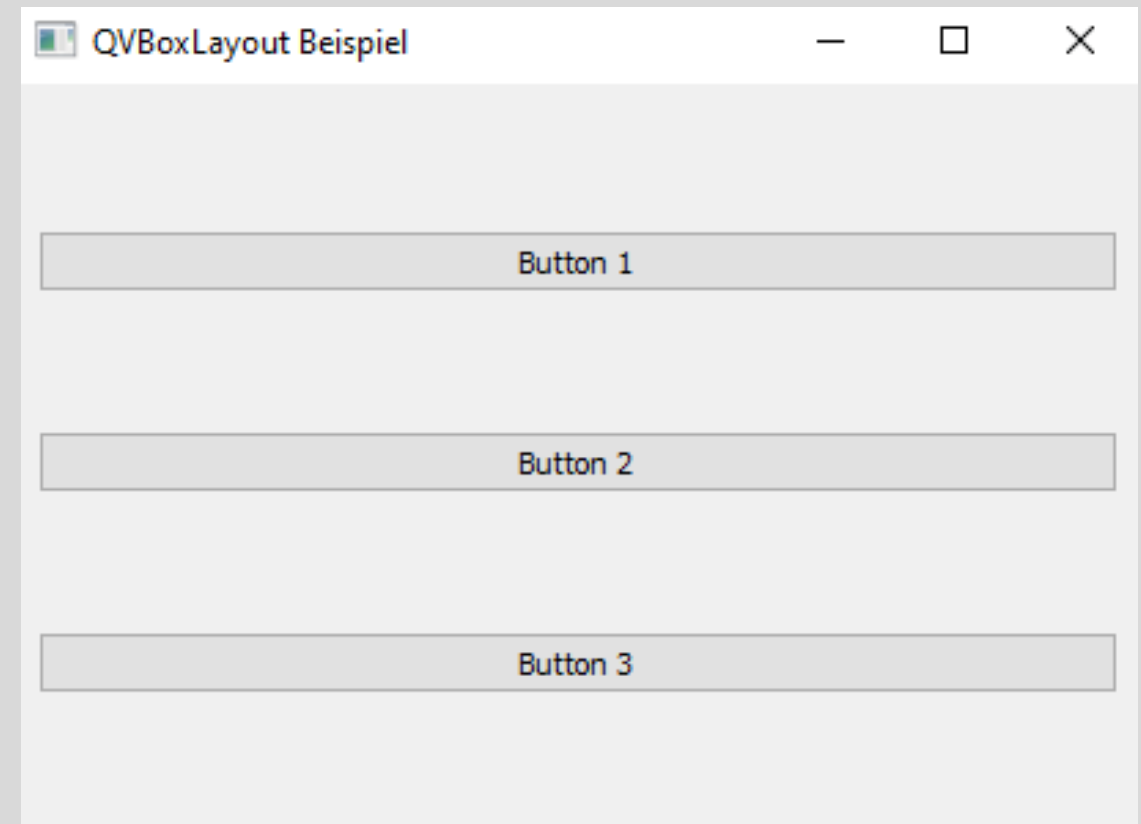
```
.....
# Erstellen eines zentralen Widgets und Layouts.
central_widget = QWidget()
self.setCentralWidget(central_widget)

# Erstellen eines vertikalen Layouts
vbox = QVBoxLayout()

# Erstellen von Schaltflächen
btn1 = QPushButton('Button 1')
btn2 = QPushButton('Button 2')
btn3 = QPushButton('Button 3')

# Hinzufügen der Schaltflächen zum Layout
vbox.addWidget(btn1)
vbox.addWidget(btn2)
vbox.addWidget(btn3)

# Setzen des Layouts für das zentrale Widget
central_widget.setLayout(vbox)
.....
```





Layout-Management in PyQt5

QHBoxLayout arrangiert Widgets in einer **horizontalen** Reihe. Die Syntax lautet wie folgt:

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

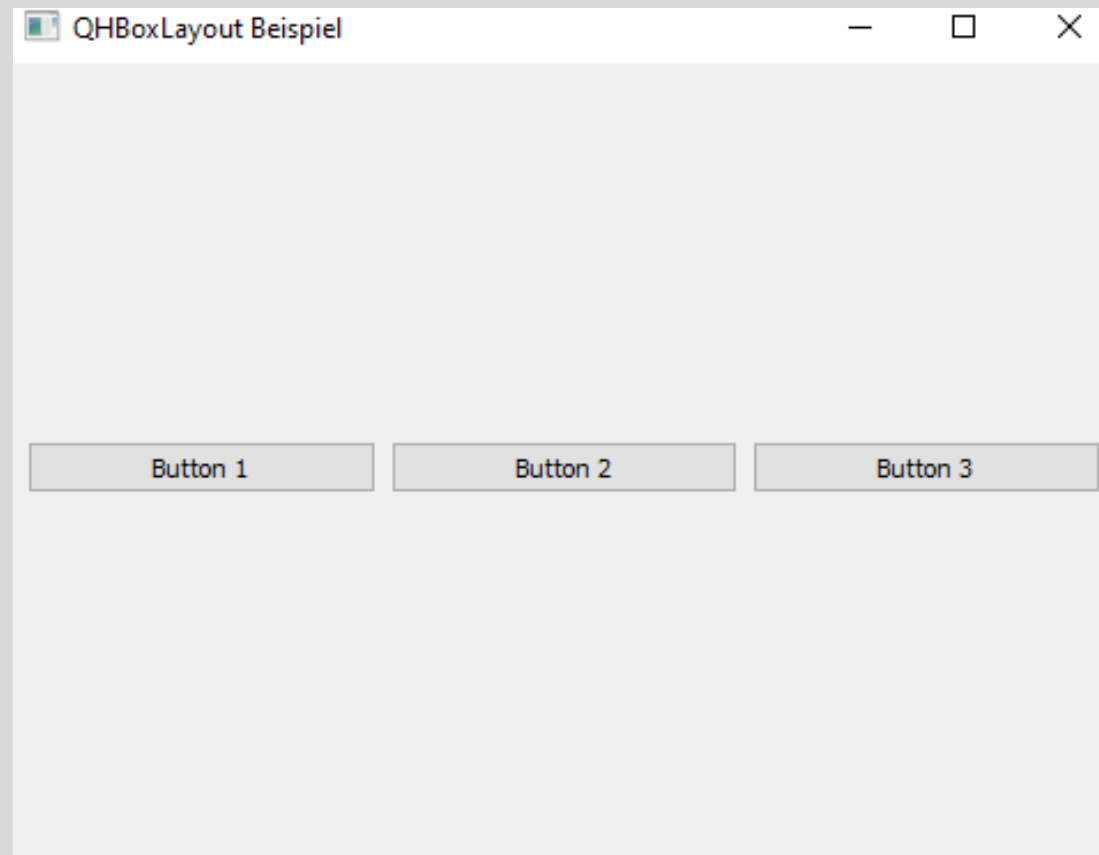
```
.....
.....
# Erstellen eines zentralen Widgets und Layouts.
central_widget = QWidget()
self.setCentralWidget(central_widget)

# Erstellen eines horizontalen Layouts
hbox = QHBoxLayout()

# Erstellen von Schaltflächen
btn1 = QPushButton('Button 1')
btn2 = QPushButton('Button 2')
btn3 = QPushButton('Button 3')

# Hinzufügen der Schaltflächen zum Layout
hbox.addWidget(btn1)
hbox.addWidget(btn2)
hbox.addWidget(btn3)

# Setzen des Layouts für das zentrale Widget
central_widget.setLayout(hbox)
.....
.....
```





Layout-Management in PyQt5

QGridLayout arrangiert Widgets in einem **Raster (Grid)** aus Zeilen und Spalten. Die Syntax lautet wie folgt:

```
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget, QGridLayout, QPushButton
```

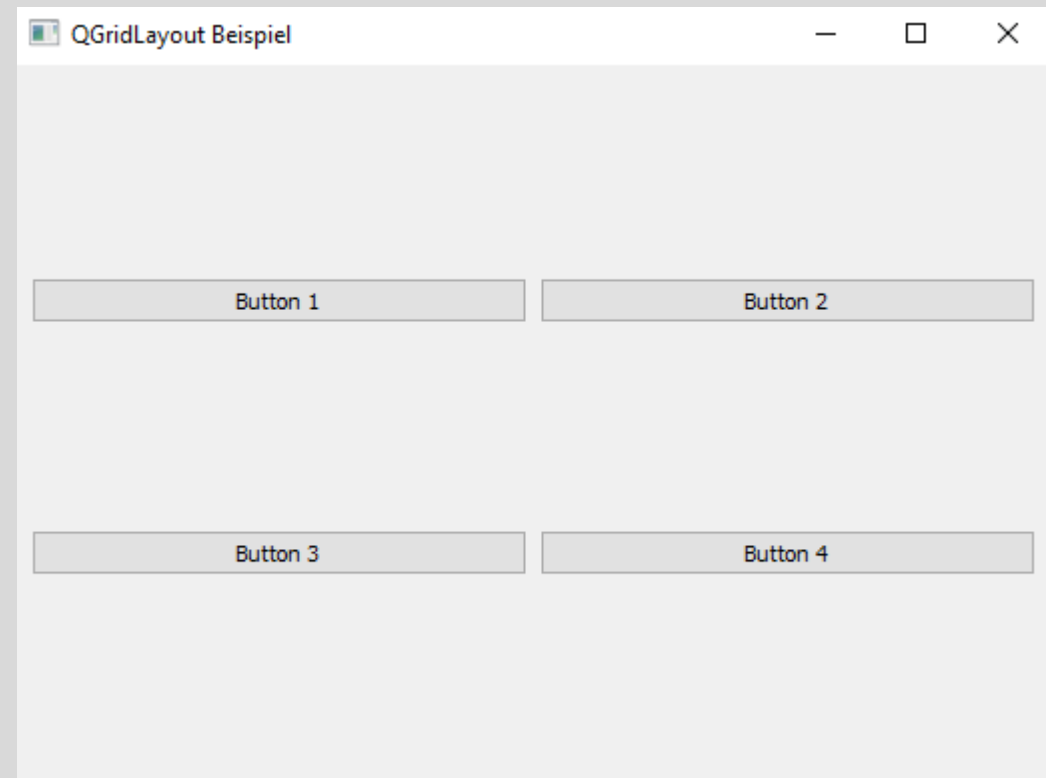
```
.....
# Erstellen eines zentralen Widgets und Layouts.
central_widget = QWidget()
self.setCentralWidget(central_widget)

# Erstellen eines Rasterlayouts
grid = QGridLayout()

# Erstellen von Schaltflächen
btn1 = QPushButton('Button 1')
btn2 = QPushButton('Button 2')
btn3 = QPushButton('Button 3')
btn4 = QPushButton('Button 4')

# Hinzufügen der Schaltflächen zum Rasterlayout (Zeile, Spalte)
grid.addWidget(btn1, 0, 0)
grid.addWidget(btn2, 0, 1)
grid.addWidget(btn3, 1, 0)
grid.addWidget(btn4, 1, 1)

# Setzen des Layouts für das zentrale Widget
central_widget.setLayout(grid)
.....
```





Layout-Management in PyQt5

QFormLayout wird verwendet, um **Formulare** mit Labels und Eingabefeldern anzuordnen. Die Syntax lautet wie folgt:

```
import sys
from PyQt5.QtWidgets import QApplication, QMainWindow, QWidget, QFormLayout, QLabel, QLineEdit
```

```
.....
# Erstellen eines zentralen Widgets und Layouts
```

```
central_widget = QWidget()
self.setCentralWidget(central_widget)
```

```
# Erstellen eines Formularlayouts
```

```
form = QFormLayout()
```

```
# Erstellen von Beschriftungen und Eingabefeldern
```

```
label1 = QLabel('Name:')
```

```
lineEdit1 = QLineEdit()
```

```
label2 = QLabel('Alter:')
```

```
lineEdit2 = QLineEdit()
```

```
# Hinzufügen von Beschriftungen und Eingabefeldern zum Formularlayout
```

```
form.addRow(label1, lineEdit1)
```

```
form.addRow(label2, lineEdit2)
```

```
# Setzen des Layouts für das zentrale Widget
```

```
central_widget.setLayout(form)
```

QFormLayout Beispiel

Name:

Alter:



Layout-Management in PyQt5

Oftmals müssen Sie verschiedene Layouts kombinieren, um das gewünschte Design zu erreichen. Hier ist ein Beispiel, das `QVBoxLayout` und `QHBoxLayout` kombiniert:

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

.....

# Erstellen eines zentralen Widgets und Layouts
central_widget = QWidget()
self.setCentralWidget(central_widget)

# Hauptlayout (vertikal)
main_layout = QVBoxLayout()

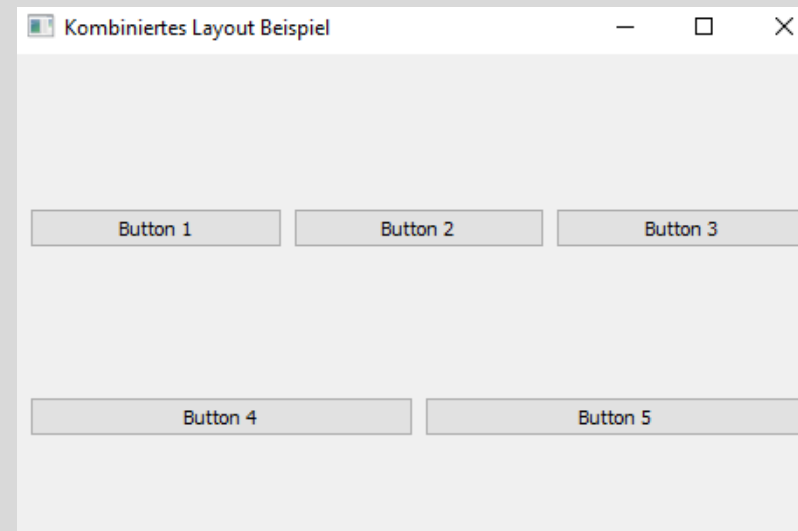
# Oberes Layout (horizontal)
top_layout = QHBoxLayout()

# Unteres Layout (horizontal)
bottom_layout = QHBoxLayout()

# Erstellen von Schaltflächen, Hinzufügen der Schaltflächen (Button 1, 2, 3) zum oberen Layout, Hinzufügen der Schaltflächen (Button 4, 5) zum unteren Layout

# Hinzufügen der horizontalen Layouts zum Vertikalen Hauptlayout
main_layout.addLayout(top_layout)
main_layout.addLayout(bottom_layout)

# Setzen des Layouts für das zentrale Widget
central_widget.setLayout(main_layout)
```





Layout-Management in PyQt5

Mehr erkunden :

<https://www.pythonguis.com/tutorials/pyqt-layouts/>

<https://realpython.com/python-pyqt-layout/>

https://www.tutorialspoint.com/pyqt5/pyqt5_layout_management.htm