

QWidget

QWidgetは、PySide6のすべてのUIコンポーネントの基底クラスです。
独立したウィンドウまたは他のウィジェットの子として使用できます。

インポート

```
from PySide6.QtWidgets import QWidget
```

基本的な使用方法

```
from PySide6.QtWidgets import QWidget, QVBoxLayout

widget = QWidget()
layout = QVBoxLayout()
widget.setLayout(layout)
```

主要なメソッド

レイアウト管理

setLayout(layout)

```
layout = QVBoxLayout()
widget.setLayout(layout)
```

ウィジェットにレイアウトを設定します。

パラメータ:- `layout` (QLayout): 設定するレイアウト

layout()

```
current_layout = widget.layout()
```

現在のレイアウトを取得します。

戻り値: QLayout - 現在のレイアウト（設定されていない場合はNone）

サイズ設定

resize(width, height)

```
widget.resize(400, 300)
```

ウィジェットのサイズを設定します。

パラメータ:- `width` (int): 幅 - `height` (int): 高さ

setFixedSize(width, height)

```
widget.setFixedSize(400, 300)
```

ウィジェットの固定サイズを設定し、リサイズを無効にします。

パラメータ:- `width` (int): 固定幅 - `height` (int): 固定高さ

表示制御

show()

```
widget.show()
```

ウィジェットを表示します。

hide()

```
widget.hide()
```

ウィジェットを非表示にします。

setVisible(visible)

```
widget.setVisible(True) # 表示
widget.setVisible(False) # 非表示
```

ウィジェットの表示/非表示を設定します。

パラメータ:- `visible` (bool): True で表示、False で非表示

有効/無効制御

setEnabled(enabled)

```
widget.setEnabled(False) # 無効化
widget.setEnabled(True) # 有効化
```

ウィジェットの有効/無効を設定します。

パラメータ:- `enabled` (bool): True で有効、False で無効

isEnabled()

```
is_enabled = widget.isEnabled()
```

ウィジェットが有効かどうかを確認します。

戻り値: bool - 有効な場合True

使用例

基本的なウィジェット

```
import sys
from PySide6.QtWidgets import QApplication, QWidget, QVBoxLayout, QLabel, QPushButton

class MyWidget(QWidget):
    def __init__(self):
        super().__init__()
        self.setWindowTitle("My Widget")
        self.resize(300, 200)

        # レイアウトの設定
        layout = QVBoxLayout()
        self.setLayout(layout)

        # 子ウィジェットの追加
        label = QLabel("Hello World!")
        button = QPushButton("Click Me")

        layout.addWidget(label)
        layout.addWidget(button)

if __name__ == "__main__":
    app = QApplication(sys.argv)
    widget = MyWidget()
    widget.show()
    sys.exit(app.exec())
```

コンテナウィジェット

```
from PySide6.QtWidgets import QWidget, QVBoxLayout, QLabel

# 他のウィジェットのコンテナとして使用
container = QWidget()
layout = QVBoxLayout()
container.setLayout(layout)

for i in range(3):
    label = QLabel(f"Label {i+1}")
    layout.addWidget(label)
```

注意事項

- QWidgetは他のすべてのウィジェットクラスの基底クラスです
- レイアウトを設定する前に子ウィジェットを追加しないでください
- 親を持たないQWidgetは独立したウィンドウとして表示されます