

QPushButton

QPushButtonは、ユーザーがクリックできるボタンウィジェットです。シグナルとスロットメカニズムを使用してイベント処理を行います。

インポート

```
from PySide6.QtWidgets import QPushButton
```

基本的な使用方法

```
from PySide6.QtWidgets import QPushButton

button = QPushButton("クリックしてください")
button.clicked.connect(some_function)
```

主要なメソッド

テキスト設定

setText(text)

```
button.setText("新しいボタン")
```

ボタンに表示するテキストを設定します。

パラメータ: - **text** (str): ボタンに表示するテキスト

text()

```
current_text = button.text()
```

現在のボタンテキストを取得します。

戻り値: str - 現在表示されているテキスト

イベント処理

clicked.connect(slot)

```
def on_button_click():
    print("ボタンがクリックされました")

button.clicked.connect(on_button_click)
```

ボタンクリック時に呼び出される関数を接続します。

パラメータ: - **slot** (callable): クリック時に呼び出される関数

clicked.disconnect()

```
button.clicked.disconnect()
```

すべてのクリックイベント接続を解除します。

サイズとレイアウト

setFixedSize(width, height)

```
button.setFixedSize(100, 30)
```

ボタンの固定サイズを設定します。

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

setSizePolicy(policy)

```
from PySide6.QtWidgets import QSizePolicy
button.setSizePolicy(QSizePolicy.Policy.Expanding, QSizePolicy.Policy.Fixed)
```

ボタンのサイズポリシーを設定します。

スタイル設定

setStyleSheet(styleSheet)

```
button.setStyleSheet("""
    QPushButton {
        background-color: #3498db;
        color: white;
        border: none;
        padding: 10px;
        border-radius: 5px;
    }
    QPushButton:hover {
        background-color: #2980b9;
    }
""")
```

CSSライクなスタイルシートを設定します。

パラメータ: - `styleSheet` (str): スタイルシート文字列

setFont(font)

```
from PySide6.QtGui import QFont
font = QFont("Arial", 12, QFont.Weight.Bold)
button.setFont(font)
```

ボタンのフォントを設定します。

パラメータ: - `font` (QFont): 設定するフォント

状態制御

setEnabled(enabled)

```
button.setEnabled(False) # ボタンを無効化
button.setEnabled(True)  # ボタンを有効化
```

ボタンの有効/無効を設定します。

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

setCheckable(checkable)

```
button.setCheckable(True)
```

ボタンをトグルボタンとして使用できるようにします。

パラメータ: - `checkable` (bool): True でチェック可能

isChecked()

```
is_pressed = button.isChecked()
```

チェック可能なボタンの状態を取得します。

戻り値: bool - チェックされている場合True

使用例

基本的なボタン

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

class MainWindow(QWidget):
    def __init__(self):
        super().__init__()
        layout = QVBoxLayout()
        self.setLayout(layout)

        self.label = QLabel("ボタンをクリックしてください")
        layout.addWidget(self.label)

        button = QPushButton("クリック")
        button.clicked.connect(self.on_button_click)
        layout.addWidget(button)

    def on_button_click(self):
        self.label.setText("ボタンがクリックされました！")

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

複数のボタン

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

class MultiButtonWidget(QWidget):
    def __init__(self):
        super().__init__()
        layout = QVBoxLayout()
        self.setLayout(layout)

        self.status_label = QLabel("何も押されていません")
        layout.addWidget(self.status_label)

        # 複数のボタンを作成
        buttons = ["ボタン1", "ボタン2", "ボタン3"]
        for button_text in buttons:
            button = QPushButton(button_text)
            button.clicked.connect(lambda checked, text=button_text: self.on_button_click(text))
            layout.addWidget(button)

    def on_button_click(self, button_name):
        self.status_label.setText(f"{button_name} がクリックされました")
```

スタイル付きボタン

```
from PySide6.QtWidgets import QPushButton
from PySide6.QtGui import QFont

def create_styled_button(text, color="#3498db"):
    button = QPushButton(text)
    button.setStyleSheet(f"""
        QPushButton {{
            background-color: {color};
            color: white;
            border: none;
            padding: 12px 24px;
            border-radius: 6px;
            font-size: 14px;
            font-weight: bold;
        }}
        QPushButton:hover {{
            background-color: #2980b9;
        }}
        QPushButton:pressed {{
            background-color: #21618c;
        }}
        QPushButton:disabled {{
            background-color: #bdc3c7;
            color: #7f8c8d;
        }}
    """)
    return button

# 使用例
save_button = create_styled_button("保存", "#27ae60")
delete_button = create_styled_button("削除", "#e74c3c")
cancel_button = create_styled_button("キャンセル", "#95a5a6")
```

トグルボタン

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

class ToggleButtonExample(QWidget):
    def __init__(self):
        super().__init__()
        layout = QVBoxLayout()
        self.setLayout(layout)

        self.status_label = QLabel("オフ")
        layout.addWidget(self.status_label)

        self.toggle_button = QPushButton("ON/OFF")
        self.toggle_button.setCheckable(True)
        self.toggle_button.clicked.connect(self.on_toggle)
        layout.addWidget(self.toggle_button)

    def on_toggle(self):
        if self.toggle_button.isChecked():
            self.status_label.setText("オン")
            self.toggle_button.setText("オフにする")
        else:
            self.status_label.setText("オフ")
            self.toggle_button.setText("オンにする")
```

よく使用されるスタイルシートプロパティ

```
/* 基本スタイル */
QPushButton {
    background-color: #3498db;
    color: white;
    border: 2px solid #2980b9;
    padding: 8px 16px;
    border-radius: 4px;
}

/* ホバー状態 */
QPushButton:hover {
    background-color: #2980b9;
}

/* 押下状態 */
QPushButton:pressed {
    background-color: #21618c;
}

/* 無効状態 */
QPushButton:disabled {
    background-color: #bdc3c7;
    color: #7f8c8d;
}

/* チェック状態 (トグルボタン) */
QPushButton:checked {
    background-color: #27ae60;
}
```

注意事項

- `clicked.connect()` で関数を接続する際、引数が必要な場合は `lambda` を使用してください
- スタイルシートでは疑似状態 (`:hover`, `:pressed` など) を活用して見た目を向上させることができます
- トグルボタンを使用する場合は `setCheckable(True)` を忘れずに設定してください