

# QGroupBox クラス

## 概要

**QGroupBox** は、関連するウィジェットをグループ化し、視覚的に整理するためのコンテナウィジェットです。タイトル付きの枠線でウィジェットを囲み、ユーザーインターフェースの構造を明確にします。フォーム、設定画面、オプション群の整理に広く使用されます。

## 基本的な使用方法

```
from PySide6.QtWidgets import QApplication, QGroupBox, QVBoxLayout, QCheckBox, QWidget
import sys

app = QApplication(sys.argv)
window = QWidget()
main_layout = QVBoxLayout()

# 基本的なQGroupBoxの作成
group_box = QGroupBox("設定オプション")
group_layout = QVBoxLayout()

# グループ内にウィジェットを追加
group_layout.addWidget(QCheckBox("オプション1"))
group_layout.addWidget(QCheckBox("オプション2"))
group_layout.addWidget(QCheckBox("オプション3"))

group_box.setLayout(group_layout)
main_layout.addWidget(group_box)

window.setLayout(main_layout)
window.show()
sys.exit(app.exec())
```

## 主要なプロパティとメソッド

### タイトルの設定

メソッド	説明	例
<code>setTitle(title)</code>	グループボックスのタイトルを設定	<code>group_box.setTitle("新しいタイトル")</code>
<code>title()</code>	現在のタイトルを取得	<code>title = group_box.title()</code>

### チェック可能なグループボックス

メソッド	説明	例
<code>setCheckable(checkable)</code>	チェック可能にする	<code>group_box.setCheckable(True)</code>
<code>isCheckable()</code>	チェック可能かどうかを確認	<code>checkable = group_box.isCheckable()</code>
<code>setChecked(checked)</code>	チェック状態を設定	<code>group_box.setChecked(True)</code>
<code>isChecked()</code>	チェック状態を取得	<code>checked = group_box.isChecked()</code>

### アライメント

メソッド	説明	例
<code>setAlignment(alignment)</code>	タイトルの配置を設定	<code>group_box.setAlignment(Qt.AlignCenter)</code>
<code>alignment()</code>	現在の配置を取得	<code>align = group_box.alignment()</code>

### フラット表示

メソッド	説明	例
<code>setFlat(flat)</code>	フラット表示モードを設定	<code>group_box.setFlat(True)</code>
<code>isFlat()</code>	フラット表示かどうかを確認	<code>flat = group_box.isFlat()</code>

## 主要なシグナル

シグナル	説明	使用例
<code>toggled(checked)</code>	チェック状態が変更された時	<code>group_box.toggled.connect(on_group_toggled)</code>

シグナル	説明	使用例
clicked( <b>checked</b> )	クリックされた時	group_box.clicked.connect( <b>on_group_clicked</b> )

## 実用的な使用例

### 1. 基本的な設定グループ

```
# 通知設定のグループ
notification_group = QGroupBox("通知設定")
notification_layout = QVBoxLayout()

email_check = QCheckBox("メール通知")
sms_check = QCheckBox("SMS通知")
push_check = QCheckBox("プッシュ通知")

notification_layout.addWidget(email_check)
notification_layout.addWidget(sms_check)
notification_layout.addWidget(push_check)

notification_group.setLayout(notification_layout)
```

### 2. チェック可能なグループボックス

```
# 高度な設定（有効/無効切り替え可能）
advanced_group = QGroupBox("高度な設定")
advanced_group.setCheckable(True)
advanced_group.setChecked(False) # 初期状態では無効

advanced_layout = QVBoxLayout()
debug_check = QCheckBox("デバッグモード")
verbose_check = QCheckBox("詳細ログ")
experimental_check = QCheckBox("実験的機能")

advanced_layout.addWidget(debug_check)
advanced_layout.addWidget(verbose_check)
advanced_layout.addWidget(experimental_check)

advanced_group.setLayout(advanced_layout)

# グループの有効/無効に応じて内部ウィジェットも制御
def on_advanced_toggled(checked):
    # グループが無効の場合、内部のウィジェットも無効になる
    print(f"高度な設定: {'有効' if checked else '無効'}")

advanced_group.toggled.connect(on_advanced_toggled)
```

### 3. 複数のグループを使ったフォーム

```
from PySide6.QtWidgets import QLineEdit, QSpinBox, QComboBox, QHBoxLayout

# ユーザー情報フォーム
main_layout = QVBoxLayout()

# 基本情報グループ
basic_group = QGroupBox("基本情報")
basic_layout = QVBoxLayout()

name_layout = QHBoxLayout()
name_layout.addWidget(QLabel("名前:"))
name_edit = QLineEdit()
name_layout.addWidget(name_edit)
basic_layout.addLayout(name_layout)

age_layout = QHBoxLayout()
age_layout.addWidget(QLabel("年齢:"))
age_spin = QSpinBox()
age_spin.setRange(0, 120)
age_layout.addWidget(age_spin)
basic_layout.addLayout(age_layout)

basic_group.setLayout(basic_layout)

# 連絡先情報グループ
contact_group = QGroupBox("連絡先情報")
contact_layout = QVBoxLayout()

email_layout = QHBoxLayout()
email_layout.addWidget(QLabel("メール:"))
email_edit = QLineEdit()
email_layout.addWidget(email_edit)
contact_layout.addLayout(email_layout)

phone_layout = QHBoxLayout()
phone_layout.addWidget(QLabel("電話:"))
phone_edit = QLineEdit()
phone_layout.addWidget(phone_edit)
contact_layout.addLayout(phone_layout)

contact_group.setLayout(contact_layout)
```

```
# 設定グループ
settings_group = QGroupBox("設定")
settings_group.setCheckable(True)
settings_layout = QVBoxLayout()

theme_layout = QHBoxLayout()
theme_layout.addWidget(QLabel("テーマ:"))
theme_combo = QComboBox()
theme_combo.addItems(["ライト", "ダーク", "自動"])
theme_layout.addWidget(theme_combo)
settings_layout.addLayout(theme_layout)

settings_group.setLayout(settings_layout)

# すべてのグループをメインレイアウトに追加
main_layout.addWidget(basic_group)
main_layout.addWidget(contact_group)
main_layout.addWidget(settings_group)
```

#### 4. ラジオボタンのグループ化

```
from PySide6.QtWidgets import QRadioButton

# 優先度選択グループ
priority_group = QGroupBox("優先度")
priority_layout = QVBoxLayout()

low_radio = QRadioButton("低")
medium_radio = QRadioButton("中")
high_radio = QRadioButton("高")
urgent_radio = QRadioButton("緊急")

# デフォルト選択
medium_radio.setChecked(True)

priority_layout.addWidget(low_radio)
priority_layout.addWidget(medium_radio)
priority_layout.addWidget(high_radio)
priority_layout.addWidget(urgent_radio)

priority_group.setLayout(priority_layout)
```

#### 5. 水平レイアウトでのグループ配置

```
# 複数のグループを水平に配置
horizontal_layout = QHBoxLayout()

# 左側のグループ
left_group = QGroupBox("入力設定")
left_layout = QVBoxLayout()
left_layout.addWidget(QCheckBox("自動保存"))
left_layout.addWidget(QCheckBox("自動補完"))
left_group.setLayout(left_layout)

# 右側のグループ
right_group = QGroupBox("表示設定")
right_layout = QVBoxLayout()
right_layout.addWidget(QCheckBox("行番号表示"))
right_layout.addWidget(QCheckBox("構文ハイライト"))
right_group.setLayout(right_layout)

horizontal_layout.addWidget(left_group)
horizontal_layout.addWidget(right_group)
```

### スタイリングとカスタマイズ

#### CSS スタイルシートの例

```
group_box.setStyleSheet("""
    QGroupBox {
        font-weight: bold;
        border: 2px solid #cccccc;
        border-radius: 8px;
        margin-top: 10px;
        padding-top: 10px;
    }
    QGroupBox::title {
        subcontrol-origin: margin;
        left: 10px;
        padding: 0 5px 0 5px;
        color: #333333;
        background-color: white;
    }
    QGroupBox::indicator {
        width: 13px;
        height: 13px;
    }
    QGroupBox::indicator:unchecked {
        border: 1px solid #cccccc;
        background-color: white;
    }
""")
```

```
    }
    QGroupBox::indicator:checked {
        border: 1px solid #007acc;
        background-color: #007acc;
    }
}
""")
```

## フラット表示のスタイリング

```
# フラット表示 (枠線なし)
flat_group = QGroupBox("フラットグループ")
flat_group.setFlat(True)
flat_group.setStyleSheet("""
    QGroupBox {
        font-weight: bold;
        color: #666666;
        border: none;
        margin-top: 5px;
    }
    QGroupBox::title {
        subcontrol-origin: margin;
        left: 0px;
        padding: 0 0 5px 0;
    }
}
""")
```

## ベストプラクティス

### 1. 論理的なグループ化

```
# 良い例: 関連する機能をグループ化
security_group = QGroupBox("セキュリティ設定")
security_layout = QVBoxLayout()

password_check = QCheckBox("パスワード保護")
encryption_check = QCheckBox("データ暗号化")
two_factor_check = QCheckBox("二要素認証")

security_layout.addWidget(password_check)
security_layout.addWidget(encryption_check)
security_layout.addWidget(two_factor_check)
security_group.setLayout(security_layout)
```

### 2. 適切なタイトルの設定

```
# 良い例: 明確で簡潔なタイトル
network_group = QGroupBox("ネットワーク設定")
appearance_group = QGroupBox("外観とテーマ")

# 避けるべき例: 曖昧なタイトル
misc_group = QGroupBox("その他") # 何の設定か不明
```

### 3. チェック可能グループの適切な使用

```
# 機能全体の有効/無効を制御する場合に使用
backup_group = QGroupBox("自動バックアップ")
backup_group.setCheckable(True)
backup_group.setChecked(True)

backup_layout = QVBoxLayout()
backup_layout.addWidget(QCheckBox("毎日バックアップ"))
backup_layout.addWidget(QCheckBox("クラウド同期"))

backup_group.setLayout(backup_layout)

# グループが無効になると、内部のウィジェットも自動的に無効になる
```

### 4. レスポンシブなレイアウト

```
# ウィンドウサイズに応じてグループの配置を調整
def create_responsive_groups():
    main_layout = QVBoxLayout()

    # 小さな画面では縦に配置
    if window_width < 800:
        main_layout.addWidget(group1)
        main_layout.addWidget(group2)
    else:
        # 大きな画面では横に配置
        horizontal_layout = QHBoxLayout()
        horizontal_layout.addWidget(group1)
        horizontal_layout.addWidget(group2)
        main_layout.addLayout(horizontal_layout)
```

```
return main_layout
```

## 注意事項

- ユーザビリティ:** グループのタイトルは機能を明確に表現する
- 視覚的階層:** 重要度に応じてグループの配置を決める
- アクセシビリティ:** チェック可能グループの状態を明確に示す
- パフォーマンス:** 深いネストは避け、シンプルな構造を保つ

## 関連するクラス

- QFrame:** フレーム表示の基底クラス
- QWidget:** すべてのウィジェットの基底クラス
- QVBoxLayout/QHBoxLayout:** レイアウト管理
- QButtonGroup:** ボタンのグループ管理
- QTabWidget:** タブ形式のグループ化

## 参考リンク

- [Qt公式ドキュメント - QGroupBox](#)
- [PySide6公式ドキュメント](#)