

# QCheckBox クラス

## 概要

**QCheckBox** は、ユーザーがオプションを選択（チェック）または選択解除（アンチェック）できるウィジェットです。複数の選択肢から複数のアイテムを選択する際に使用され、フォーム、設定画面、オプション選択などで広く活用されます。

## 基本的な使用方法

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

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

# 基本的なQCheckBoxの作成
checkbox = QCheckBox("同意する")
layout.addWidget(checkbox)

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

## 主要なプロパティとメソッド

### チェック状態の操作

メソッド	説明	例
<code>setChecked(checked)</code>	チェック状態を設定	<code>checkbox.setChecked(True)</code>
<code>isChecked()</code>	チェック状態を取得	<code>is_checked = checkbox.isChecked()</code>
<code>toggle()</code>	チェック状態を切り替え	<code>checkbox.toggle()</code>
<code>setCheckState(state)</code>	詳細なチェック状態を設定	<code>checkbox.setCheckState(Qt.PartiallyChecked)</code>
<code>checkState()</code>	詳細なチェック状態を取得	<code>state = checkbox.checkState()</code>

### テキストとアイコン

メソッド	説明	例
<code>setText(text)</code>	ラベルテキストを設定	<code>checkbox.setText("新しいオプション")</code>
<code>text()</code>	ラベルテキストを取得	<code>text = checkbox.text()</code>
<code>setIcon(icon)</code>	アイコンを設定	<code>checkbox.setIcon(QIcon("icon.png"))</code>
<code>icon()</code>	アイコンを取得	<code>icon = checkbox.icon()</code>

### 三状態チェックボックス

メソッド	説明	例
<code>setTristate(tristate)</code>	三状態モードを有効/無効	<code>checkbox.setTristate(True)</code>
<code>isTristate()</code>	三状態モードかどうかを確認	<code>is_tri = checkbox.isTristate()</code>

## チェック状態

QCheckBoxは以下の状態を持ちます：

状態	説明	値
<code>Qt.Unchecked</code>	チェックなし	0
<code>Qt.PartiallyChecked</code>	部分的にチェック（三状態モード）	1
<code>Qt.Checked</code>	チェック済み	2

## 主要なシグナル

シグナル	説明	使用例
<code>toggled(<i>checked</i>)</code>	チェック状態が変更された時	<code>checkbox.toggled.connect(on_toggled)</code>
<code>stateChanged(<i>state</i>)</code>	状態が変更された時（三状態対応）	<code>checkbox.stateChanged.connect(on_state_changed)</code>
<code>clicked(<i>checked</i>)</code>	クリックされた時	<code>checkbox.clicked.connect(on_clicked)</code>

## 実用的な使用例

### 1. 基本的なオプション選択

```
# 複数のオプション
options = ["メール通知", "SMS通知", "プッシュ通知"]
checkboxes = []

for option in options:
    checkbox = QCheckBox(option)
    checkboxes.append(checkbox)
    layout.addWidget(checkbox)
```

### 2. 利用規約同意チェック

```
terms_checkbox = QCheckBox("利用規約に同意する")
terms_checkbox.toggled.connect(lambda checked: submit_button.setEnabled(checked))
```

### 3. 三状態チェックボックス（親子関係）

```
parent_checkbox = QCheckBox("すべて選択")
parent_checkbox.setTristate(True)

child_checkboxes = [
    QCheckBox("オプション1"),
    QCheckBox("オプション2"),
    QCheckBox("オプション3")
]

def update_parent_state():
    checked_count = sum(1 for cb in child_checkboxes if cb.isChecked())
    if checked_count == 0:
        parent_checkbox.setCheckState(Qt.Unchecked)
    elif checked_count == len(child_checkboxes):
        parent_checkbox.setCheckState(Qt.Checked)
    else:
        parent_checkbox.setCheckState(Qt.PartiallyChecked)

for checkbox in child_checkboxes:
    checkbox.toggled.connect(update_parent_state)
```

### 4. 条件付きオプション

```
enable_feature = QCheckBox("高度な機能を有効にする")
advanced_options = QCheckBox("詳細オプション")

# 依存関係の設定
enable_feature.toggled.connect(advanced_options.setEnabled)
advanced_options.setEnabled(False) # 初期状態では無効
```

## QRadioButton との比較

QCheckBoxとよく似たウィジェットにQRadioButtonがあります：

特徴	QCheckBox	QRadioButton
選択方式	複数選択可能	単一選択（グループ内）
用途	オプション設定、機能有効/無効	択一選択
表示	四角いチェックマーク	丸いラジオボタン
状態	三状態サポート	二状態のみ

### QRadioButton の基本的な使用方法

```
from PySide6.QtWidgets import QRadioButton, QButtonGroup

# ラジオボタンの作成
radio1 = QRadioButton("オプション1")
```

```
radio2 = QRadioButton("オプション2")
radio3 = QRadioButton("オプション3")

# グループ化（排他制御）
button_group = QButtonGroup()
button_group.addButton(radio1)
button_group.addButton(radio2)
button_group.addButton(radio3)

# デフォルト選択
radio1.setChecked(True)
```

## スタイリングとカスタマイズ

### CSS スタイルシートの例

```
checkbox.setStyleSheet("""
    QCheckBox {
        font-size: 14px;
        color: #333;
        spacing: 10px;
    }
    QCheckBox::indicator {
        width: 18px;
        height: 18px;
    }
    QCheckBox::indicator:unchecked {
        border: 2px solid #cccccc;
        background-color: white;
        border-radius: 3px;
    }
    QCheckBox::indicator:checked {
        border: 2px solid #007acc;
        background-color: #007acc;
        border-radius: 3px;
    }
    QCheckBox::indicator:checked {
        image: url(checkmark.png);
    }
""")
```

## ベストプラクティス

### 1. 明確なラベルの使用

```
# 良い例
checkbox = QCheckBox("毎日メール通知を受け取る")

# 避けるべき例
checkbox = QCheckBox("メール")
```

### 2. 適切なグループ化

```
# 関連するオプションをグループ化
notification_group = QGroupBox("通知設定")
notification_layout = QVBoxLayout()

email_cb = QCheckBox("メール通知")
sms_cb = QCheckBox("SMS通知")
push_cb = QCheckBox("プッシュ通知")

notification_layout.addWidget(email_cb)
notification_layout.addWidget(sms_cb)
notification_layout.addWidget(push_cb)
notification_group.setLayout(notification_layout)
```

### 3. 状態変更の処理

```
def handle_option_changed(checkered):
    if checked:
        print("オプションが有効になりました")
        # 関連する機能を有効化
    else:
        print("オプションが無効になりました")
        # 関連する機能を無効化

checkbox.toggled.connect(handle_option_changed)
```

## 注意事項

- アクセシビリティ: ラベルは簡潔で分かりやすくする
- ユーザビリティ: 関連するオプションは視覚的にグループ化する

- 3. **状態管理**: 複雑な依存関係がある場合は、状態管理を適切に設計する
- 4. **パフォーマンス**: 大量のチェックボックスがある場合は、レイアウトの最適化を検討する

## 関連するクラス

- **QRadioButton**: 単一選択用のラジオボタン
- **QButtonGroup**: ボタンのグループ管理
- **QGroupBox**: ウィジェットのグループ化コンテナ
- **QAbstractButton**: ボタン系ウィジェットの基底クラス

## 参考リンク

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