

Qt定数とフラグ

Qtクラスには、PySide6で使用される様々な定数とフラグが定義されています。これらは配置、マウスボタン、キーボードなどの設定に使用されます。

インポート

```
from PySide6.QtCore import Qt
```

アライメント（配置）フラグ

水平配置

| 定数 | 説明 |
|-------------------------------|--------|
| Qt.AlignmentFlag.AlignLeft | 左揃え |
| Qt.AlignmentFlag.AlignRight | 右揃え |
| Qt.AlignmentFlag.AlignHCenter | 水平中央揃え |
| Qt.AlignmentFlag.AlignJustify | 両端揃え |

垂直配置

| 定数 | 説明 |
|-------------------------------|--------|
| Qt.AlignmentFlag.AlignTop | 上揃え |
| Qt.AlignmentFlag.AlignBottom | 下揃え |
| Qt.AlignmentFlag.AlignVCenter | 垂直中央揃え |

組み合わせ配置

| 定数 | 説明 |
|------------------------------|-------------|
| Qt.AlignmentFlag.AlignCenter | 中央揃え（水平＋垂直） |

使用例

```
from PySide6.QtWidgets import QLabel
from PySide6.QtCore import Qt

label = QLabel("中央揃えのテキスト")
label.setAlignment(Qt.AlignmentFlag.AlignCenter)

# 複数のフラグを組み合わせ
label.setAlignment(Qt.AlignmentFlag.AlignLeft | Qt.AlignmentFlag.AlignTop)
```

オリエンテーション（方向）

| 定数 | 説明 |
|---------------------------|------|
| Qt.Orientation.Horizontal | 水平方向 |
| Qt.Orientation.Vertical | 垂直方向 |

使用例

```
from PySide6.QtWidgets import QSlider
from PySide6.QtCore import Qt

horizontal_slider = QSlider(Qt.Orientation.Horizontal)
vertical_slider = QSlider(Qt.Orientation.Vertical)
```

マウスボタン

| 定数 | 説明 |
|---------------------------|----------|
| Qt.MouseButton.NoButton | マウスボタンなし |
| Qt.MouseButton.LeftButton | 左ボタン |

| 定数 | 説明 |
|-----------------------------|------|
| Qt.MouseButton.RightButton | 右ボタン |
| Qt.MouseButton.MiddleButton | 中ボタン |

使用例

```
def mousePressEvent(self, event):
    if event.button() == Qt.MouseButton.LeftButton:
        print("左クリック")
    elif event.button() == Qt.MouseButton.RightButton:
        print("右クリック")
```

キーボード修飾キー

| 定数 | 説明 |
|-------------------------------------|---------------------------|
| Qt.KeyboardModifier.NoModifier | 修飾キーなし |
| Qt.KeyboardModifier.ShiftModifier | Shiftキー |
| Qt.KeyboardModifier.ControlModifier | Ctrlキー |
| Qt.KeyboardModifier.AltModifier | Altキー |
| Qt.KeyboardModifier.MetaModifier | Metaキー（WindowsキーまたはCmdキー） |

使用例

```
def keyPressEvent(self, event):
    if event.modifiers() & Qt.KeyboardModifier.ControlModifier:
        if event.key() == Qt.Key.Key_S:
            print("Ctrl+Sが押されました")
```

よく使用されるキーコード

制御キー

| 定数 | 説明 |
|----------------------|-------------|
| Qt.Key.Key_Enter | Enterキー |
| Qt.Key.Key_Return | Returnキー |
| Qt.Key.Key_Escape | Escapeキー |
| Qt.Key.Key_Space | スペースキー |
| Qt.Key.Key_Tab | Tabキー |
| Qt.Key.Key_Backspace | Backspaceキー |
| Qt.Key.Key_Delete | Deleteキー |

矢印キー

| 定数 | 説明 |
|------------------|-----|
| Qt.Key.Key_Up | 上矢印 |
| Qt.Key.Key_Down | 下矢印 |
| Qt.Key.Key_Left | 左矢印 |
| Qt.Key.Key_Right | 右矢印 |

文字キー（例）

| 定数 | 説明 |
|--------------|-------------|
| Qt.Key.Key_A | Aキー |
| Qt.Key.Key_B | Bキー |
| ... | （他のアルファベット） |
| Qt.Key.Key_Z | Zキー |

数字キー

| 定数 | 説明 |
|--------------|-----|
| Qt.Key.Key_0 | 0キー |
| Qt.Key.Key_1 | 1キー |

| 定数 | 説明 |
|--------------|--------|
| ... | (他の数字) |
| Qt.Key.Key_9 | 9キー |

ウィンドウフラグ

基本ウィンドウタイプ

| 定数 | 説明 |
|----------------------------|--------------|
| Qt.WindowType.Widget | 標準ウィジェット |
| Qt.WindowType.Window | ウィンドウ |
| Qt.WindowType.Dialog | ダイアログ |
| Qt.WindowType.Sheet | シート (macOS) |
| Qt.WindowType.Drawer | ドロワー (macOS) |
| Qt.WindowType.Popup | ポップアップ |
| Qt.WindowType.Tool | ツールウィンドウ |
| Qt.WindowType.ToolTip | ツールチップ |
| Qt.WindowType.SplashScreen | スプラッシュスクリーン |

ウィンドウヒント

| 定数 | 説明 |
|--|----------|
| Qt.WindowType.FramelessWindowHint | フレームなし |
| Qt.WindowType.WindowTitleHint | タイトルバー表示 |
| Qt.WindowType.WindowSystemMenuHint | システムメニュー |
| Qt.WindowType.WindowMinimizeButtonHint | 最小化ボタン |
| Qt.WindowType.WindowMaximizeButtonHint | 最大化ボタン |
| Qt.WindowType.WindowCloseButtonHint | 閉じるボタン |
| Qt.WindowType.WindowStaysOnTopHint | 常に最前面 |

使用例

```
from PySide6.QtWidgets import QWidget
from PySide6.QtCore import Qt

# 常に最前面のフレームなしウィンドウ
widget = QWidget()
widget.setWindowFlags(
    Qt.WindowType.FramelessWindowHint |
    Qt.WindowType.WindowStaysOnTopHint
)
```

フォーカスポリシー

| 定数 | 説明 |
|----------------------------|----------------|
| Qt.FocusPolicy.NoFocus | フォーカスなし |
| Qt.FocusPolicy.TabFocus | Tabキーでフォーカス |
| Qt.FocusPolicy.ClickFocus | クリックでフォーカス |
| Qt.FocusPolicy.StrongFocus | Tab + クリック |
| Qt.FocusPolicy.WheelFocus | マウスホイールでもフォーカス |

使用例

```
from PySide6.QtWidgets import QPushButton
from PySide6.QtCore import Qt

button = QPushButton("ボタン")
button.setFocusPolicy(Qt.FocusPolicy.NoFocus) # フォーカスを受け取らない
```

サイズポリシー

| 定数 | 説明 |
|----------------------------|-------|
| Qt.SizePolicy.Policy.Fixed | 固定サイズ |

| 定数 | 説明 |
|---------------------------------------|----------|
| Qt.SizePolicy.Policy.Minimum | 最小サイズ |
| Qt.SizePolicy.Policy.Maximum | 最大サイズ |
| Qt.SizePolicy.Policy.Preferred | 推奨サイズ |
| Qt.SizePolicy.Policy.Expanding | 拡張可能 |
| Qt.SizePolicy.Policy.MinimumExpanding | 最小拡張 |
| Qt.SizePolicy.Policy.Ignored | サイズヒント無視 |

カーソル形状

| 定数 | 説明 |
|-----------------------------------|--------------|
| Qt.CursorShape.ArrowCursor | 標準矢印 |
| Qt.CursorShape.CrossCursor | 十字 |
| Qt.CursorShape.WaitCursor | 待機（砂時計など） |
| Qt.CursorShape.IBeamCursor | テキスト入力（Iビーム） |
| Qt.CursorShape.SizeVerCursor | 垂直リサイズ |
| Qt.CursorShape.SizeHorCursor | 水平リサイズ |
| Qt.CursorShape.SizeBDiagCursor | 斜めリサイズ（\） |
| Qt.CursorShape.SizeFDiagCursor | 斜めリサイズ（/） |
| Qt.CursorShape.SizeAllCursor | 全方向リサイズ |
| Qt.CursorShape.BlankCursor | 透明カーソル |
| Qt.CursorShape.SplitVCursor | 垂直分割 |
| Qt.CursorShape.SplitHCursor | 水平分割 |
| Qt.CursorShape.PointingHandCursor | 指差し（リンク） |
| Qt.CursorShape.ForbiddenCursor | 禁止 |
| Qt.CursorShape.WhatsThisCursor | ヘルプ |
| Qt.CursorShape.BusyCursor | ビジー状態 |
| Qt.CursorShape.OpenHandCursor | 開いた手 |
| Qt.CursorShape.ClosedHandCursor | 閉じた手 |

使用例

```
from PySide6.QtWidgets import QLabel
from PySide6.QtCore import Qt

label = QLabel("ホバーしてください")
label.setCursor(Qt.CursorShape.PointingHandCursor)
```

実用的な使用例

アライメントの組み合わせ

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

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

        alignments = [
            ("左上", Qt.AlignmentFlag.AlignLeft | Qt.AlignmentFlag.AlignTop),
            ("中央上", Qt.AlignmentFlag.AlignHCenter | Qt.AlignmentFlag.AlignTop),
            ("右上", Qt.AlignmentFlag.AlignRight | Qt.AlignmentFlag.AlignTop),
            ("中央", Qt.AlignmentFlag.AlignCenter),
            ("左下", Qt.AlignmentFlag.AlignLeft | Qt.AlignmentFlag.AlignBottom),
        ]

        for text, alignment in alignments:
            label = QLabel(text)
            label.setAlignment(alignment)
            label.setStyleSheet("border: 1px solid black; min-height: 50px;")
            layout.addWidget(label)
```

キーボードショートカット

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

```
from PySide6.QtCore import Qt

class ShortcutExample(QWidget):
    def keyPressEvent(self, event):
        key = event.key()
        modifiers = event.modifiers()

        if modifiers & Qt.KeyboardModifier.ControlModifier:
            if key == Qt.Key.Key_N:
                print("新規作成 (Ctrl+N)")
            elif key == Qt.Key.Key_O:
                print("開く (Ctrl+O)")
            elif key == Qt.Key.Key_S:
                print("保存 (Ctrl+S)")

        elif key == Qt.Key.Key_F1:
            print("ヘルプ (F1)")
        elif key == Qt.Key.Key_Escape:
            print("キャンセル (Esc)")

        super().keyPressEvent(event)
```

注意事項

- フラグを組み合わせる場合は `|` 演算子を使用してください
- 古いPySide6バージョンでは一部の定数名が異なる場合があります
- Qt定数は大文字小文字を区別します
- 適切な名前空間（`Qt.AlignmentFlag.AlignCenter` など）を使用してください