

# QStyleSheet リファレンス

PySide6におけるスタイルシート（CSS風スタイリング）についての包括的なリファレンス資料です。

## 概要

QStyleSheetは、Qtウィジェットの外観をCSSライクな記法でカスタマイズできる機能です。

## 基本的な構文

### セレクター

```
/* ウィジェットタイプセレクター */
QPushButton { }
```

```
/* クラス名セレクター */
QWidget { }
```

```
/* オブジェクト名セレクター */
QPushButton#myButton { }
```

```
/* 子セレクター */
QDialog QPushButton { }
```

```
/* 疑似状態セレクター */
QPushButton:hover { }
QPushButton:pressed { }
QPushButton:disabled { }
```

### プロパティ

```
QPushButton {
    background-color: #3498db;
    color: white;
    border: 2px solid #2980b9;
    border-radius: 4px;
    padding: 8px 16px;
    font-size: 14px;
    font-weight: bold;
}
```

## よく使用されるプロパティ

### 背景・前景

```
QWidget {
    background-color: #f0f0f0;
    background-image: url(background.png);
    background-repeat: no-repeat;
    background-position: center;
    color: #333333;
}
```

### ボーダー

```
QLineEdit {
    border: 2px solid gray;
    border-radius: 5px;
    border-style: solid;
    border-top: 1px solid red;
    border-right: 2px dashed blue;
}
```

### フォント

```
QLabel {
    font-family: Arial, sans-serif;
    font-size: 12pt;
    font-weight: bold;
    font-style: italic;
    text-decoration: underline;
}
```

## マージン・パディング

```
QPushButton {
    margin: 10px;
    margin-top: 5px;
    margin-right: 15px;
    padding: 8px 16px;
    padding-left: 20px;
}
```

## 疑似状態

### 一般的な疑似状態

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

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

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

/* フォーカス時 */
QLineEdit: focus {
    border: 2px solid #3498db;
}

/* チェック時 (チェックボックス、ラジオボタン) */
QCheckBox: checked {
    color: #27ae60;
}
```

### ウィジェット固有の疑似状態

```
/* QTabWidget */
QTabWidget::tab-bar: selected {
    background-color: #3498db;
}

/* QScrollBar */
QScrollBar: vertical {
    background-color: #f0f0f0;
}

/* QComboBox */
QComboBox: drop-down {
    border: none;
}
```

## ウィジェット別スタイリング例

### QPushButton

```
QPushButton {
    background-color: #3498db;
    border: none;
    color: white;
    padding: 8px 16px;
    border-radius: 4px;
    font-weight: bold;
}

QPushButton: hover {
    background-color: #2980b9;
}

QPushButton: pressed {
    background-color: #21618c;
}
```

### QLineEdit

```
QLineEdit {
    border: 2px solid #bdc3c7;
    border-radius: 4px;
    padding: 8px;
}
```

```
        font-size: 14px;
        background-color: white;
    }

    QLineEdit:focus {
        border-color: #3498db;
    }

    QLineEdit:disabled {
        background-color: #ecf0f1;
        color: #7f8c8d;
    }
}
```

### QTabWidget

```
QTabWidget::pane {
    border: 1px solid #bdc3c7;
    background-color: white;
}

QTabWidget::tab-bar {
    left: 5px;
}

QTabBar::tab {
    background-color: #ecf0f1;
    border: 1px solid #bdc3c7;
    padding: 8px 16px;
    margin-right: 2px;
}

QTabBar::tab:selected {
    background-color: white;
    border-bottom: none;
}

QTabBar::tab:hover {
    background-color: #d5dbdb;
}
}
```

### QScrollBar

```
QScrollBar:vertical {
    background-color: #f8f9fa;
    width: 12px;
    border-radius: 6px;
}

QScrollBar::handle:vertical {
    background-color: #bdc3c7;
    border-radius: 6px;
    min-height: 20px;
}

QScrollBar::handle:vertical:hover {
    background-color: #95a5a6;
}
}
```

## 高度なテクニック

### グラデーション

```
QPushButton {
    background: qlineargradient(
        x1: 0, y1: 0, x2: 0, y2: 1,
        stop: 0 #3498db,
        stop: 1 #2980b9
    );
}

/* 放射状グラデーション */
QWidget {
    background: qradialgradient(
        cx: 0.5, cy: 0.5, radius: 0.5,
        stop: 0 white,
        stop: 1 gray
    );
}
}
```

### アニメーション対応

```
QPushButton {
    background-color: #3498db;
    transition: background-color 0.3s ease;
}

QPushButton:hover {
}
```

```
        background-color: #2980b9;
    }
```

## サブコントロール

```
QComboBox {
    border: 1px solid gray;
    border-radius: 3px;
    padding: 1px 18px 1px 3px;
}

QComboBox::drop-down {
    subcontrol-origin: padding;
    subcontrol-position: top right;
    width: 15px;
    border-left: 1px solid darkgray;
}

QComboBox::down-arrow {
    image: url(down_arrow.png);
}
```

## テーマの実装例

### ダークテーマ

```
QWidget {
    background-color: #2c3e50;
    color: #ecf0f1;
}

QPushButton {
    background-color: #34495e;
    border: 1px solid #5d6d7e;
    color: #ecf0f1;
    padding: 8px 16px;
    border-radius: 4px;
}

QPushButton:hover {
    background-color: #5d6d7e;
}

QLineEdit {
    background-color: #34495e;
    border: 1px solid #5d6d7e;
    color: #ecf0f1;
    padding: 5px;
    border-radius: 3px;
}
```

### マテリアルデザイン風

```
QPushButton {
    background-color: #2196F3;
    border: none;
    color: white;
    padding: 12px 24px;
    border-radius: 4px;
    font-weight: 500;
    text-transform: uppercase;
}

QPushButton:hover {
    background-color: #1976D2;
    box-shadow: 0 2px 4px rgba(0,0,0,0.2);
}

QPushButton:pressed {
    background-color: #0D47A1;
}
```

## デバッグとツール

### スタイルシートのデバッグ

```
# スタイルシートの適用確認
widget.setStyleSheet("border: 2px solid red;")

# 現在のスタイルシート取得
current_style = widget.styleSheet()
print(current_style)
```

## 動的スタイル変更

```
def toggle_theme(self):
    if self.dark_theme:
        self.setStyleSheet(self.light_theme_css)
    else:
        self.setStyleSheet(self.dark_theme_css)
    self.dark_theme = not self.dark_theme
```

## 注意点とベストプラクティス

- パフォーマンス: 過度に複雑なスタイルシートは描画性能に影響する
- 継承: 親ウィジェットのスタイルが子に継承される
- 特異性: より具体的なセクターが優先される
- プラットフォーム: OS固有のスタイルを上書きする場合がある

## 参考リンク

- [Qt Documentation - Qt Style Sheets](#)
- [Qt Style Sheets Reference](#)