

QFont

QFontは、フォントの種類、サイズ、スタイルなどを定義するクラスです。ウィジェットのテキスト表示フォントを設定する際に使用します。

インポート

```
from PySide6.QtGui import QFont
```

基本的な使用方法

```
from PySide6.QtGui import QFont

font = QFont("Arial", 12)
widget.setFont(font)
```

主要なメソッド

コンストラクタ

QFont()

```
font = QFont() # デフォルトフォント
```

デフォルトフォントを作成します。

QFont(family, pointSize)

```
font = QFont("Arial", 14)
```

フォントファミリーとサイズを指定してフォントを作成します。

パラメータ	型	説明
family	str	フォントファミリー名
pointSize	int	フォントサイズ（ポイント）

QFont(family, pointSize, weight)

```
font = QFont("Arial", 14, QFont.Weight.Bold)
```

フォントファミリー、サイズ、ウェイトを指定してフォントを作成します。

パラメータ	型	説明
family	str	フォントファミリー名
pointSize	int	フォントサイズ（ポイント）
weight	QFont.Weight	フォントの太さ

フォント設定

setFamily(family)

```
font.setFamily("Times New Roman")
```

フォントファミリーを設定します。

パラメータ	型	説明
family	str	フォントファミリー名

setSize(pointSize)

```
font.setSize(16)
```

フォントサイズを設定します。

パラメータ	型	説明
pointSize	int	フォントサイズ（ポイント）

setWeight(weight)

```
font.setWeight(QFont.Weight.Bold)
```

フォントの太さを設定します。

パラメータ	型	説明
weight	QFont.Weight	フォントの太さ

setBold(bold)

```
font.setBold(True)
```

太字の有効/無効を設定します。

パラメータ	型	説明
bold	bool	True で太字

setItalic(italic)

```
font.setItalic(True)
```

斜体の有効/無効を設定します。

パラメータ	型	説明
italic	bool	True で斜体

setUnderline(underline)

```
font.setUnderline(True)
```

下線の有効/無効を設定します。

パラメータ	型	説明
underline	bool	True で下線

フォント情報取得

family()

```
family_name = font.family()
```

フォントファミリー名を取得します。

戻り値: str - フォントファミリー名

pointSize()

```
size = font.pointSize()
```

フォントサイズを取得します。

戻り値: int - フォントサイズ（ポイント）

weight()

```
weight = font.weight()
```

フォントの太さを取得します。

戻り値: QFont.Weight - フォントの太さ

bold()

```
is_bold = font.bold()
```

太字かどうかを確認します。

戻り値: bool - 太字の場合True

italic()

```
is_italic = font.italic()
```

斜体かどうかを確認します。

戻り値: bool - 斜体の場合True

使用例

基本的なフォント設定

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

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

        # 標準フォント
        label1 = QLabel("標準フォント")
        layout.addWidget(label1)

        # カスタムフォント
        label2 = QLabel("カスタムフォント")
        custom_font = QFont("Arial", 16, QFont.Weight.Bold)
        label2.setFont(custom_font)
        layout.addWidget(label2)

        # 斜体フォント
        label3 = QLabel("斜体フォント")
        italic_font = QFont("Times", 14)
        italic_font.setItalic(True)
        label3.setFont(italic_font)
        layout.addWidget(label3)

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

様々なフォントスタイル

```
from PySide6.QtWidgets import QWidget, QVBoxLayout, QLabel
from PySide6.QtGui import QFont

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

        font_examples = [
            ("細字フォント", QFont("Arial", 12, QFont.Weight.Light)),
```

```
        ("標準フォント", QFont("Arial", 12, QFont.Weight.Normal)),
        ("太字フォント", QFont("Arial", 12, QFont.Weight.Bold)),
        ("極太フォント", QFont("Arial", 12, QFont.Weight.Black)),
    ]

    for text, font in font_examples:
        label = QLabel(text)
        label.setFont(font)
        layout.addWidget(label)

# 組み合わせ例
combo_label = QLabel("太字+斜体+下線")
combo_font = QFont("Arial", 14)
combo_font.setBold(True)
combo_font.setItalic(True)
combo_font.setUnderline(True)
combo_label.setFont(combo_font)
layout.addWidget(combo_label)
```

動的フォント変更

```
from PySide6.QtWidgets import QWidget, QVBoxLayout, QLabel, QPushButton, QSlider
from PySide6.QtGui import QFont
from PySide6.QtCore import Qt

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

        # テキストラベル
        self.text_label = QLabel("サンプルテキスト")
        self.current_font = QFont("Arial", 12)
        self.text_label.setFont(self.current_font)
        layout.addWidget(self.text_label)

        # フォントサイズスライダー
        self.size_slider = QSlider(Qt.Orientation.Horizontal)
        self.size_slider.setMinimum(8)
        self.size_slider.setMaximum(48)
        self.size_slider.setValue(12)
        self.size_slider.valueChanged.connect(self.change_font_size)
        layout.addWidget(self.size_slider)

        # 太字トグル
        bold_button = QPushButton("太字切り替え")
        bold_button.setCheckable(True)
        bold_button.clicked.connect(self.toggle_bold)
        layout.addWidget(bold_button)

        # 斜体トグル
        italic_button = QPushButton("斜体切り替え")
        italic_button.setCheckable(True)
        italic_button.clicked.connect(self.toggle_italic)
        layout.addWidget(italic_button)

    def change_font_size(self, size):
        self.current_font.setPointSize(size)
        self.text_label.setFont(self.current_font)

    def toggle_bold(self, checked):
        self.current_font.setBold(checked)
        self.text_label.setFont(self.current_font)

    def toggle_italic(self, checked):
        self.current_font.setItalic(checked)
        self.text_label.setFont(self.current_font)
```

日本語フォント設定

```
from PySide6.QtWidgets import QWidget, QVBoxLayout, QLabel
from PySide6.QtGui import QFont

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

        # 日本語フォントの例
        japanese_fonts = [
            ("メイリオ", "こんにちは、世界！"),
            ("MS ゴシック", "こんにちは、世界！"),
            ("MS 明朝", "こんにちは、世界！"),
            ("游ゴシック", "こんにちは、世界！"),
        ]

        for font_name, text in japanese_fonts:
            label = QLabel(f"{font_name}: {text}")
            font = QFont(font_name, 14)
            label.setFont(font)
            layout.addWidget(label)
```

フォントウェイト定数

定数	値	説明
QFont.Weight.Thin	100	極細
QFont.Weight.ExtraLight	200	極細
QFont.Weight.Light	300	細字
QFont.Weight.Normal	400	標準（デフォルト）
QFont.Weight.Medium	500	中字
QFont.Weight.DemiBold	600	準太字
QFont.Weight.Bold	700	太字
QFont.Weight.ExtraBold	800	極太字
QFont.Weight.Black	900	最太字

よく使用されるフォントファミリー

Windows

- "Arial"
- "Times New Roman"
- "Courier New"
- "Verdana"
- "メイリオ"
- "MS ゴシック"
- "MS 明朝"

macOS

- "Helvetica"
- "Times"
- "Courier"
- "Hiragino Sans"
- "Hiragino Mincho ProN"

Linux

- "Liberation Sans"
- "Liberation Serif"
- "Liberation Mono"
- "DejaVu Sans"
- "Noto Sans CJK JP"

注意事項

- フォントファミリーが存在しない場合、システムのデフォルトフォントが使用されます
- 日本語テキストを表示する場合は、日本語対応フォントを指定してください
- フォントサイズは通常8〜72ポイントの範囲で設定します
- `setBold(True)` と `setWeight(QFont.Weight.Bold)` は同じ効果があります