

Python で GUI アプリを作る

PYthon で GUI アプリを作るときの選択肢とおすすめの選択、利用上の注意点

PYthon で GUI アプリを作るときの選択肢とおすすめの選択

Python で GUI アプリを作るためのフレームワークの選択肢はいくつかある。
TKInter, PySimpleGUI, PyQt6, PySide6 等である。

それぞれ、ひとことで言うと以下のとおり。

フレームワーク	メリット	デメリット
TKInter	Pythonインストール時に同梱、学習しやすい	デザインがシンプル、高度な機能は限定的
PySimpleGUI	そこそこのUIを関数ベースで作れる	構造が深い入れ子になりやすい、商用利用については最近有償化されてしまった
PyQT	かなりキレイなUIを作れる。UIデザイナーもある	商用利用ライセンスが高額
PySide6	PyQTと同様、高機能	学習コストがやや高い、シンプルなアプリには多少オーバースペック

PyQT, PySide6 とも、QT という(他の言語でも利用可能な)UI生成用ライブラリ(dll)を内部で呼び出しています。

PyQTが高額なことに反発して別団体によって作られたのがPySide6です。
Unixに対するLinuxみたいなものです。

学習者向けの推奨順序

初心者・学習目的: まずは **TKInter** から始めることをおすすめします。 - Python標準ライブラリなので追加インストール不要 - 基本的なGUIプログラミングの概念を学習しやすい - オブジェクト指向プログラミングの理解を深められる

中級者・実用目的: TKInterに慣れたら **PySide6** にステップアップ。 - 本格的で美しいアプリケーションが作成可能 - 商用利用も基本的に無料 - 長期的な投資価値が高い

PySide6のインストール

```
# 以下で、QT も自動的にインストールされます。  
pip install PySide6
```

PySide6利用上の注意

TKInter以外の選択肢ではライセンスに注意する必要があります。

なお、PySide6の利用にあたっては、以下点のにご注意ください。
(PySide6の利用許諾は、LGPL v3 ライセンスです)

1. ライセンス表記すること
2. 再配布時には改変しないこと(PyInstallerで .exe ファイルを作るときに重要)
3. 極めて凝ったwidgetについては、ひょっとするとLGPLではなくGPLライセンスかもしれないということ

特に「3.」については、QTというUIライブラリの長年問題とされているところです。
「極めて凝ったwidget」については、個別にソースコード等を見るなどして確認をする必要があります。
ただし、その「確認」がまた大変...

上記それぞれについてももしも違反があった場合は？

1. ライセンス表記をしなかった: 故意でなければ「すみません」して問題解消すれば大事になることはないだろう。(そもそも指摘されること自体稀だが)
2. PySide6のソースコードを改変したうえで再配布してしまった: プロジェクト全体のソースコードを公開する責任が生じる(かもしれない。燃料次第)
3. GPLライセンスのQT widgetを使ってしまった: プロジェクト全体のソースコードを公開する責任が生じる(かもしれない。燃料次第)

では、どのように使えばいいの？

- PySide6 から import 文でライブラリを呼び出して使っている分には、(つまり、「普通に使っていれば」) PySide6のソースコードを変更することはまずありません。
- PyInstaller で .exe 化してもライセンス違反と解釈されて問題になることはまずありません。(その仕組は講座内で追って説明します)
- 「極めて凝ったwidget」については、個別にソースコード等を見るなどして確認をする必要があります。(ただし、その「確認」がまた大変...)

上記のとおりなので、個人利用する分には心配ないですが、「会社で使う」となると、「ほぼ安心だけど、ちょっとだけ気になる点はある...」というところです。

Python GUI開発ライブラリ比較

項目	Tkinter	PySimpleGUI	PyQt6	PySide6
開発元	Python標準ライブラリ	PySimpleGUI LLC	Riverbank Computing	Qt Group
ライセンス	Python License	LGPL/商用	GPL/商用	LGPL
インストール	標準搭載	pip install	pip install	pip install
初期の学習コスト	低～中	低	中～高	中～高
外観・デザイン	シンプル	シンプル	モダン	モダン
ネイティブルック	✖	✖	✔	✔
高DPI対応	部分的	✖	✔	✔
カスタマイズ性	低	低	高	高
豊富なウィジェット	基本的	基本的	豊富	豊富

項目	Tkinter	PySimpleGUI	PyQt6	PySide6
マルチプラットフォーム	✓	✓	✓	✓
ドキュメント品質	普通	良い	優秀	優秀
コミュニティサイズ	大	中	大	大
商用利用	✓	制限あり	有償	✓
開発効率	中	高	中	中〜高
保守性	中	低	高	高
パフォーマンス	普通	普通	高	高
国際化対応	基本的	基本的	優秀	優秀
アニメーション	基本的	✗	豊富	豊富
描画・グラフィック	基本的	基本的	高機能	高機能
データバインディング	✗	✗	✓	✓
スタイルシート	✗	✗	✓ (QSS)	✓ (QSS)
UIデザイナー	✗	✗	✓	✓
適用場面	学習・簡単なツール	プロトタイプ	本格的アプリ	本格的アプリ

各ライブラリの特徴

Tkinter

メリット: - Python標準ライブラリなので追加インストール不要 - 軽量で起動が早い - 学習に最適な構造とドキュメント - 基本的なGUIプログラミングの概念をしっかりと身につけられる - オブジェクト指向プログラミングの実践練習に適している

デメリット: - デザインがシンプルで、現代的な見た目には限界がある - 高度なウィジェットが少ない - 複雑なアプリケーションには機能不足

PySimpleGUI

メリット: - 学習コストが非常に低い - 短時間でGUIアプリが作成可能 - 直感的なAPI設計

デメリット: - コードのまとまりが悪くなりがち - 商用利用に制限がある - 複雑なアプリケーション開発には不向き

PySimpleGUIは2023年から商用利用について有償化された。

PyQt6

メリット: - 非常に高機能で本格的なアプリケーション開発が可能 - 豊富なウィジェットとツール - 高いパフォーマンス

デメリット: - 商用利用には有償ライセンスが必要 - 学習コストがやや高い(クラスとインスタンスについての知識が必要) - ライセンス問題で導入を躊躇するケースも

PySide6

メリット: - PyQt6とほぼ同等の機能性 - LGPLライセンスを選択して利用すれば商用利用も基本

的に無料 - 優秀なドキュメントとコミュニティ - 本格的なアプリケーション開発に必要な機能が全て揃っている

デメリット: - 学習コストが高い（クラスとインスタンス、オブジェクト指向の深い理解が必要） - 初心者にはコンセプトが複雑すぎる場合がある - 簡単なアプリには多少オーバースペック - QT のあまり高度なコンポーネントだと LGPL ではなく GPL ライセンスかもしれない

まとめ：段階的なGUIフレームワーク学習戦略

学習初期：TKInterで基礎固め

TKInterから始める理由：

1. **導入の簡単さ:** 追加インストール不要で、すぐに学習を開始できる
2. **学習コストの低さ:** 基本的なGUIプログラミングの概念に集中できる
3. **オブジェクト指向の練習:** クラスベースの設計を自然に身につけられる
4. **デバッグの容易さ:** シンプルな構造で問題を特定しやすい

TKInterがおすすめなプロジェクト： - 学習・練習用アプリケーション - 簡単な業務ツール - プロトタイプ開発 - 基本的なデータ入力フォーム

発展期：PySide6で本格開発

TKInterに慣れたらPySide6にステップアップする理由：

1. **商用利用の自由度:** LGPLライセンスにより、ほとんどの商用プロジェクトで自由に使用可能
2. **機能の豊富さ:** 本格的なデスクトップアプリケーション開発に必要な機能が全て揃っている
3. **現代的な外観:** ネイティブルックで美しいUIが作成可能
4. **長期的な価値:** 一度覚えれば長期間使える技術スタック

PySide6がおすすめなプロジェクト： - 商用デスクトップアプリケーション - データ分析・可視化ツール - 社内業務ツール - クロスプラットフォーム対応が必要なアプリ - 将来的な機能拡張を見据えたプロジェクト

PySide6のライセンスについて

PySide6は**LGPLライセンス**（GNU Lesser General Public License）で提供されています。ただし、PySide6が依存しているQTは、ほぼLGPL、一部の凝ったwidgetだけはGPLで提供されています。

[LGPL, GPL については software_licenses.md を参照のこと](#)

実際の商用利用での扱い

✅ **問題なく利用できるケース:**

- PySide6 のライブラリをインポートして利用する
- PySide6のソースコード自体は書き換えない(通常書き換える必要など発生しないが)
- 再配布時も、「静的リンク」でライブラリを利用する(PyInstallerで .exe 化する分にはこの方法になるので心配ない)

```
# あなたのアプリケーションコード
import sys
from PySide6.QtWidgets import QApplication, QMainWindow

class MyCommercialApp(QMainWindow):
    def __init__(self):
        super().__init__()
        # あなた独自の商用ロジック
        self.setup_ui()
```

⚠ 注意が必要なケース:

- PySide6自体のソースコードを修正した場合は、その修正部分の公開が必要になる
- インストーラの生成時に CPython や Nuitka 等のコンパイルツールを使うと改変物が .exe に含まれてしまう可能性大

ライセンス表示の例

アプリケーションに以下のような表示を含めることで、ライセンス要件を満たせます：

```
このソフトウェアはPySide6を使用しています。
PySide6は LGPL ライセンスの下で配布されています。
詳細: https://www.qt.io/licensing/
```