

PyCharmとQtでトライ 操作しやすい GUIアプリ作り

第1回 環境の準備

澤田 英宏

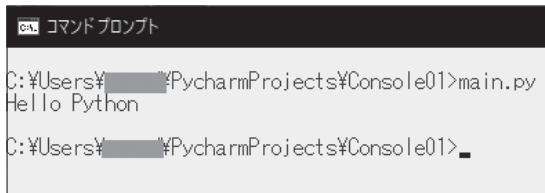
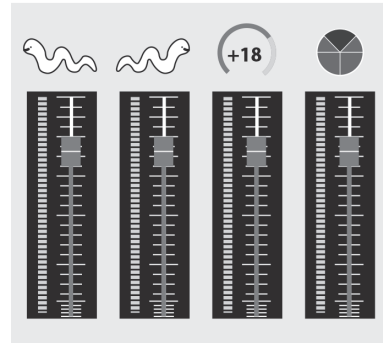


図1 いつものコンソール・アプリ

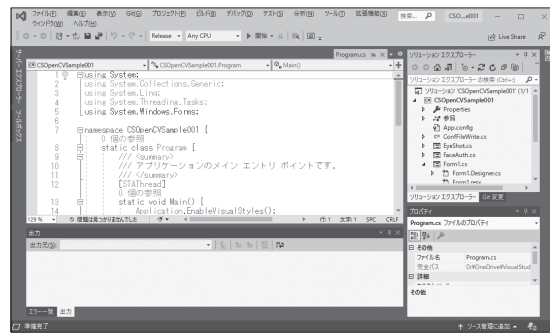
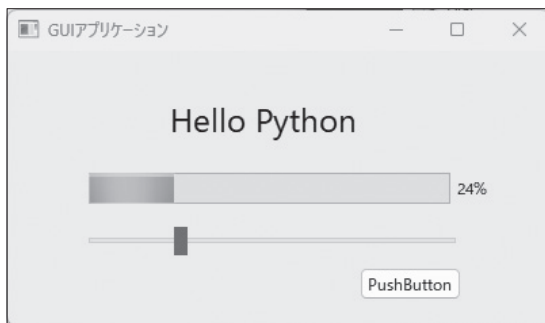


図3 Windows PCでアプリをつくるなら Visual Studio が有名

図2 Qtを使って作るのはGUIアプリ…図1と直接対比できるのではないが、GUIで頑張るとこんな感じに

● 見た目が良い方がいいよね

Pythonでプログラミングしている皆さんの中には、「これってコマンドで操作するコンソール・アプリしか作れないのかな」と思われた方はいませんか。Pythonでも豊富なライブラリを使えば、Windowsで使用するようなGUI(グラフィカル・ユーザ・インターフェース)を使用したアプリケーションを作れます。

図1はコンソール画面に表示するPythonアプリ、図2がGUIを装備したPythonアプリです。

Hello Pythonを表示しただけの2つのアプリですが、読者の皆さんどうでしょう、何らかの操作をするアプリが必要な場合は、明らかに図2のGUIアプリの方が、皆に使ってもらえる気がしませんか、その気になればゲームだって作れます。

この連載では、Pythonで誰もが直感的に使えるアプリケーションを作ってみます。今回はその第1回です。

● PythonでGUIアプリ作りを勧める理由

Windows OSにてGUIアプリを作る場合は、一般的に開発言語としてC#を使います。また、開発環境としてマイクロソフトが提供するVisual Studio(図3)を使用します。

Visual Studioそのものは大変優れていて、C#やC++などでWindowsのネイティブ・アプリケーションを作成したい場合は、ほとんどのケースでVisual Studioを使用します。それならPythonよりもVisual Studioで作った方が良いのでは…との意見もあるでしょう。

今回、筆者が勧める理由は以下です。

- 1、Pythonで利用できる膨大なライブラリを使ってプログラミングができる
- 2、本記事の内容は、ほとんどそのままWindows以外のOSでも動作可能である

プログラミング環境の準備

Pythonプログラムを作る環境としてPyCharmを利用します。以降はWindows OS(10または11)のPCを利用していることを前提に説明します。

● 統合開発環境 PyCharmをインストール

統合開発環境のPyCharmをダウンロードします。

<https://www.jetbrains.com/ja-jp/>

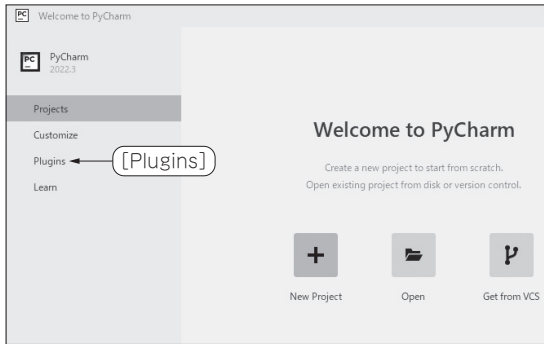


図4 日本語化対応1…[Plugins]を選択

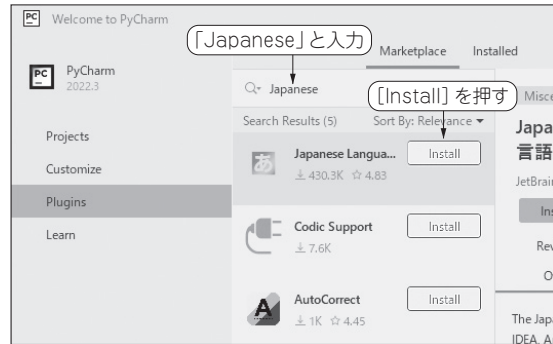


図5 日本語化対応2…検索窓に「Japanese」と入力



図6 日本語化対応3…日本語化完了、新規プロジェクトの作成へ

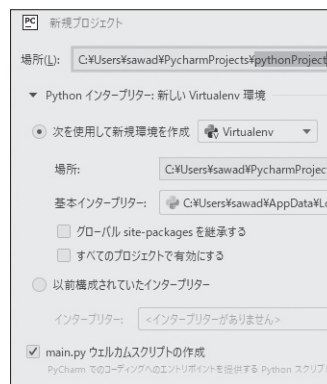


図7 新規プロジェクト作成時に必ず出る画面…何もなければ[作成]ボタンを押す
誌面の都合で右下の作成ボタンは省略している

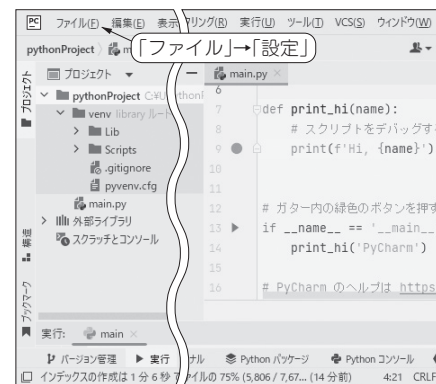


図8 PyCharmの初期…プロジェクトが表示されている状態から始める

リスト1 このプログラムが自動生成されたらプログラムが実行可能状態になっている

```
def print_hi(name):
    # スクリプトをデバッグするには以下のコード行でブレークポイントを使用してください
    print(f'Hi, {name}') # Ctrl+F8を押すとブレークポイントを切り替えます

# ガター内の緑色のボタンを押すとスクリプトを実行します
if __name__ == '__main__':
    print_hi('PyCharm')
```

[pycharm/download/#section=windows](https://www.jetbrains.com/pycharm/download/#section=windows)

上記ウェブ・ページにはProfessionalとCommunityの2つが表示されています。今回は無償版のCommunityをダウンロードします。

ダウンロードが完了したら、保存されたexeファイルを実行します。執筆時点でのファイル名はpycharm-community-2022.3.exeでした。インストール中に「DATA SHARING」の画面が出ますが、

Don't Sendを選択しました。また、Import PyCharm Settings画面では、Do not import settingsを選択しました。

● PyCharmの日本語化対応

PyCharmの基本表示は英語表記ですがプラグインを追加することで日本語に変更できます。図4に示す起動画面の左側メニュー[Plugins]を選択すると図5の画面に変わります。図5の中央上の検索用のテキスト入力欄があり、その部分に「Japanese」と入力すると直下に「Japanese Language Pack/日本語言語パック」が表示されます。表示の直下にある[Install]ボタンを押すと、日本語化パッケージがインストールされます。インストールはすぐに完了して[Install]ボタンが[Restart IDE]に変わったら、そのボタンを押してPyCharmを再起動します。

● プロジェクトの作成

プロジェクトを作成してPythonを実行します。

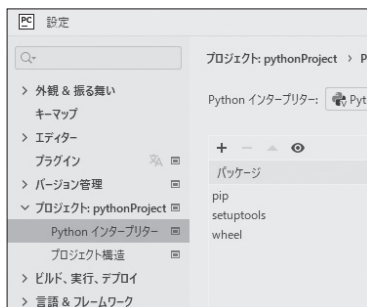


図9 PySide6のインストール1…パッケージ管理画面を開く

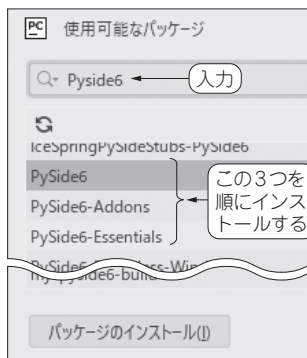


図10 PySide6のインストール2…3つのパッケージを順にインストールする



図11 今回は空のウィンドウを作るまで次回以降のベースとなる画面

図6中に「新規プロジェクト」の表記があり、その上にある[+]ボタンを押すとプロジェクトの作成が開始されます。

図7はプロジェクトの新規作成時に必ず表示される画面です。Pythonの使用バージョンなど細かい設定ができますが、特に指定がない場合はデフォルトのまま使って問題ありません。そのまま右下の[作成]ボタンを押します。

プロジェクト作成が始まり、リスト1のようなプログラムが自動生成されたら、プロジェクトの完成です。

ソースコード上で右クリックして[実行]を選択します。画面下に「Hi, PyCharm」と表示されたら、無事にプログラムが実行可能状態になっています。

Qt開発環境の準備

前項までで、PyCharmで基本的なPythonプログラムを実行できるようになりました。いよいよ今回の目的であるGUIプログラミングを可能にするための設定を行います。GUIの機能開発にはQt(キュート)というアプリケーション開発フレーム・ワークを使います。今回はQtを使うための準備まで説明します。

PyCharmにおいて、図8のようなプロジェクトが表示されている状態から始めます。上部メニューから[ファイル]-[設定]を開きます。

図9の設定画面が表示され、左側メニューの[プロジェクト]-[Pythonインタープリター]を選択すると、パッケージ管理画面が表示されます。ここで、PySide6を追加してプロジェクト内で使えるにします。

図9に示してある[+]を押してください。図10の[パッケージのインストール]ボタンを押して必要なパッケージをインストールします。今回はPySide6をインストールしたいので、図10の検索窓に「Pyside6」と入力します。すると、PySide6の名前が付いたリス

リスト2 空のウィンドウを表示した…次回以降、ここから発展させる

```
000: import sys
001: from PySide6.QtWidgets import QApplication,
    QMainWindow
002:
003: class MainWindow(QMainWindow):
004:     def __init__(self):
005:         QMainWindow.__init__(self)
006:         self.setWindowTitle('GUI アプリケーション')
007:         self.setGeometry(0, 0, 640, 480)
008:
009: if __name__ == '__main__':
010:     app = QApplication()
011:     w = MainWindow()
012:     w.show()
013:     sys.exit(app.exec())
```

トがいくつか表示されます。インストールするのは以下の3つです。

- PySide6
- PySide6-Addons
- PySide6-Essentials

順番に選択してメニュー下の[パッケージのインストール]ボタンを押してください。インストールが完了したら図9の画面に戻ります。選択した3つが追加されていれば完了です。

画面を作る

いよいよGUIアプリを作成します。図11に示す空のウィンドウだけを表示するプログラムがリスト2となります。これはウィンドウを表示する場合の定番の記述なので、そのまま使って、次回以降にさまざまな部品を追加しながらアプリを作成していきます。

さわだ・ひでひろ