

(9)Python のプログラミング環境の例



■PyCharm の場合

ここでは Python を利用するまでの環境整備について PyCharm を利用する例を説明します。内容は各プログラム・ライブラリのバージョンなどで変化しますので、あくまで参考例となります。テスト環境はインテル系の CPU を利用、Windows と Linux に対応、Python3.14 時点の情報です。

1. インストーラーの入手

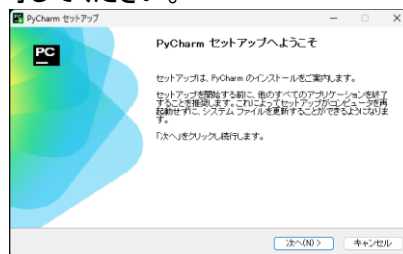
リンクのページを開き、Windows を選択、ダウンロードをクリックします。

<https://www.jetbrains.com/pycharm/download/>

2. インストーラーの実行

ダウンロードしたファイルを実行し、メッセージに従って標準でインストールします。

セキュリティ関連のダイアログも許可してください。



3. 初回実行

メニューに登録された **PyCharm** をクリックすると初回のみ環境設定と情報共有認証が実行されます。

4. プロジェクトの作成

ようこそ画面が表示されます。新規スクリプトをクリックします。



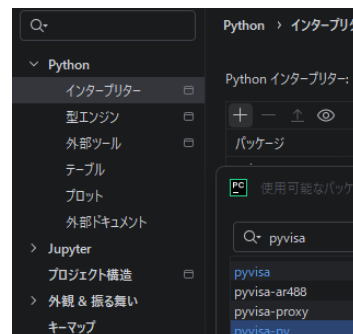
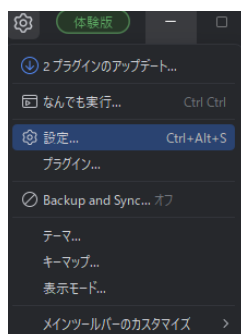
PC 内で Python を検索しインストールされていなければ最新の Python がインストールされ、サンプルスクリプトが表示されます。

5. ライブラリ・モジュールの導入

PyCharm の画面の右上の歯車をクリックし、メニューから設定をクリックします。

左の項目から Python→インタープリターを選択し、+アイコンをクリックします。

検索で必要なモジュールを追加してください。python3.14 でサンプルアプリを実行した場合は Pytk、pyVISA、pyVISA-py、pyUSB、pySerial、pyTCP、zeroconf、psutil が必要でした。



6. ソースの入力

Sample.py を編集するか、ファイルの削除・追加で実際のプログラムを入力します。

7. 環境の確認

Windows では NI-VISA をインストールしてください。なくても動作はできますが、LAN を使う場合はデバイスを登録するツール(NI-MAX)を使うと検索が利用できて便利です。また USB-TMC を使う場合は機器認識に VISA が必要となります。

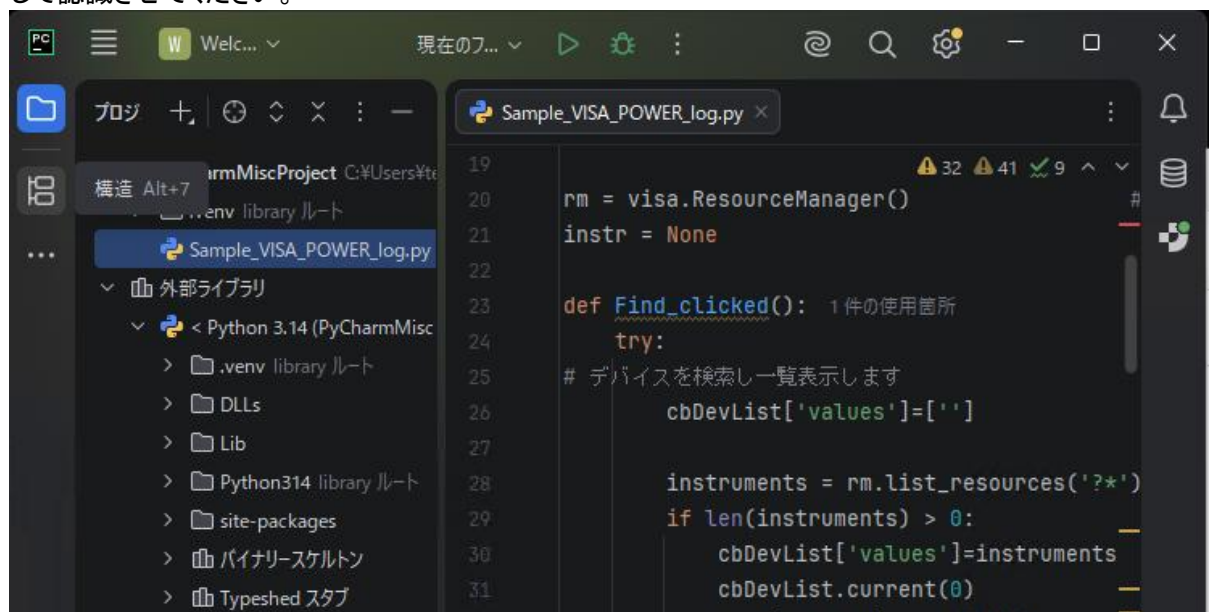
Linux 系では/dev にある使用するデバイスの RW アクセス権をユーザーに開放してください。
(sudo chmod 0777 tty*)

8. プログラムの実行

プログラムの実行は▶ボタンで行います。

実行時にモジュール不足のメッセージが表示される場合は別途追加してください。

LAN デバイスは直接 IP と Socket ポート番号を指定("TCPIP0::192.168.1.100::3000::SOCKET"など)して認識させてください。



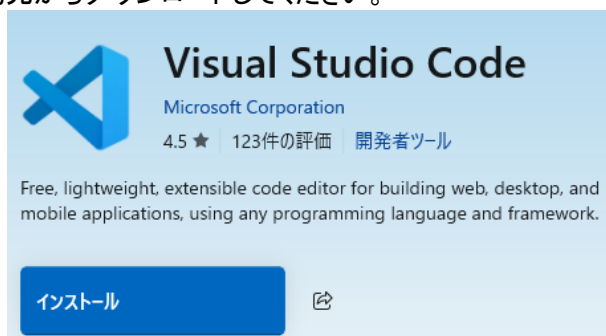


■ Visual Studio Code の場合

ここでは Windows で Python を利用するまでの環境整備について Visual Studio Code を利用する例を説明します。内容は各プログラム・ライブラリのバージョンなどで変化しますので、あくまで参考例となります。テスト環境は Windows11 で Intel 系の CPU を利用、Python3.13 時点の情報ですが、ARM 系の CPU や Linux などでも同じ環境が利用可能です。

1. Microsoft Store によるインストール

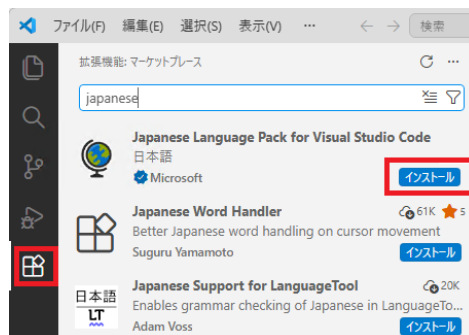
Microsoft Store を開き、**Visual Studio Code** を検索しインストールしてください。次に **python** を検索し、最新版をインストールします。Store アプリは開発元の最新版よりも古い場合があります、必要な場合は開発元からダウンロードしてください。



2. Visual Studio Code の設定

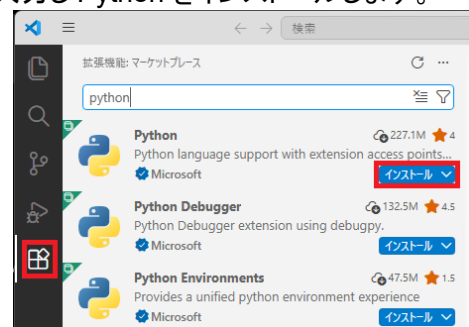
・日本語化

Visual Studio Code を実行し、左側の Extensions(拡張機能)のアイコンをクリックし、検索に Japanese を入力、Japanese Language Pack のインストールをクリックします。インストール後に環境を日本語にするかの確認が表示され、Change Language のアイコンをクリックして日本語化を完了します。



・Python 用の設定

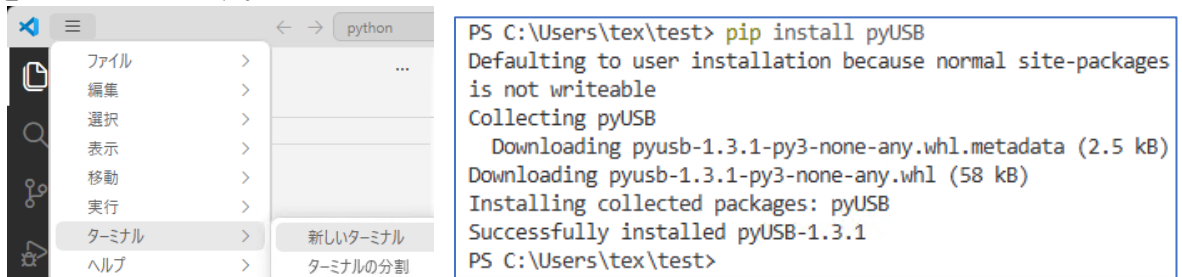
拡張機能の検索に python を入力し Python をインストールします。



メッセージに従って再起動します。

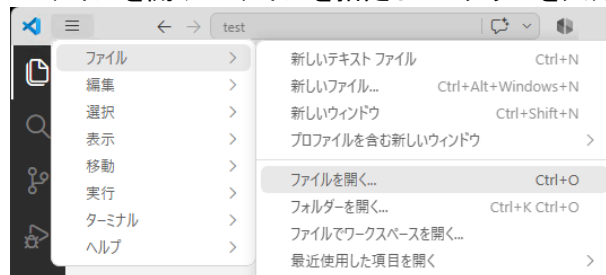
・ライブラリの設定

Visual Studio Code メニューの[≡]または[⌵]から新しいターミナルを開き、Pip install pytk を入力し、pytk をインストールします。同様に pyVISA、pyVISA-py、pyUSB、pySerial、pyTCP、zeroconf、psutil をインストールします。



3. ソースの入力

ファイルの新しいファイルかファイルを開くでファイルを指定しプログラムを入力します。



4. 環境の確認

Windows では NI-VISA などライブラリをインストールしてください。なくても動作はできますが、LAN を使う場合はデバイスを登録するツール(NI-MAX)を使うと検索が利用できて便利です。また USB-TMC を使う場合は機器認識に VISA ライブラリが必要となります。

Linux 系では/devにある使用するデバイスの RW アクセス権をユーザーに開放してください。
(sudo chmod 0777 tty*)

5. プログラムの実行

実行は▶ボタンで行います。実行時にモジュール不足のメッセージが表示される場合は別途追加してください。

LAN デバイスは直接 IP と Socket ポート番号を指定("TCPIP0::192.168.1.100::3000::SOCKET"など)して認識させてください。

