

連載

Windowsユーザーに贈るLinux超入門

第28回

# Visual Studio CodeとWSLでPython開発に挑戦してみよう

後藤大地

2019/08/07 09:15

## 追実験してみよう

[前々回](#)、Visual Studio CodeとWSLでLinux向けアプリケーションの開発を行うというテーマを取り上げた。その時は「[Take your Linux development experience in Windows to the next level with WSL and Visual Studio Code Remote](#)」を引き合いに出し、いくつかのスクリーンショットを掲載しつつ、実施に動作している様子を紹介した。

しかし、あの内容は開発者であれば追実験できたと思うが、開発者以外では難しかったように思う。

マイクロソフトから類似の記事は随時公開されているのだが、2019年7月22日に公開された次の記事が追実験に適している。ボリュームもそこそこで、初めてセットアップして使ってみるにはいい内容だ。

- [An In Depth Tutorial on Linux Development on Windows with WSL and Visual Studio Code](#)

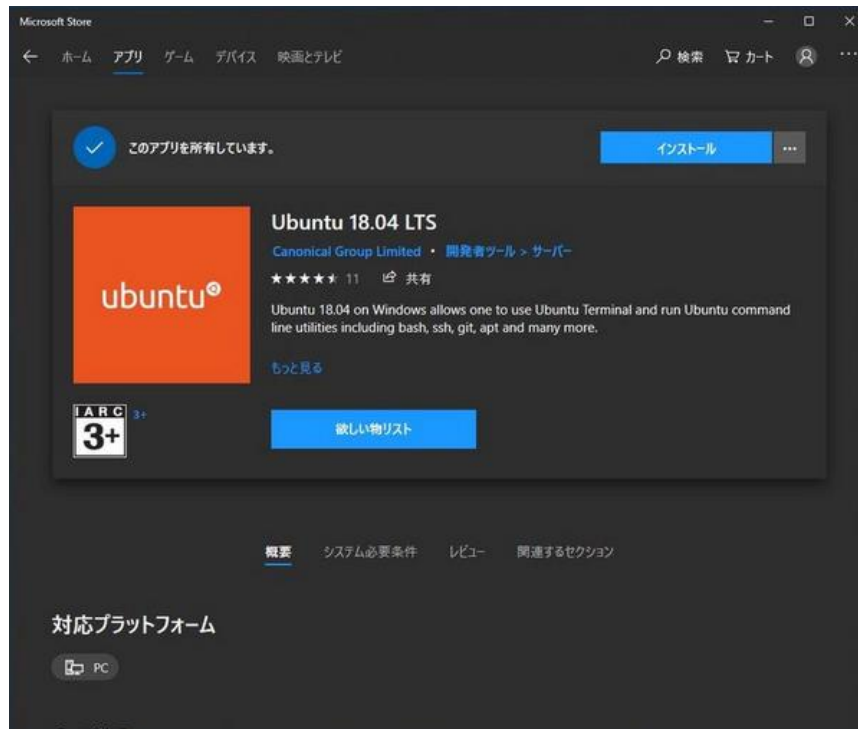
そこで、今回は上記の記事を追う形で、実際に試してみる方法を紹介する。作業の流れは次のとおりだ。

- Ubuntu 18.04 LTSのインストール
- Ubuntu 18.04 LTSにPythonをインストール
- Visual Studio Codeのインストール
- Visual Studio Codeにエクステンションをインストール
- Visual Studio CodeでPythonのソースコードを編集し、Ubuntu 18.04 LTS上のPythonを使って実行する

Visual Studio CodeとWSLを使った開発をまだ経験したことがなければ、ぜひ一度試してみていただきたい。Visual Studio CodeとWSLによる開発がどのようなものか感じてもらえと思う。

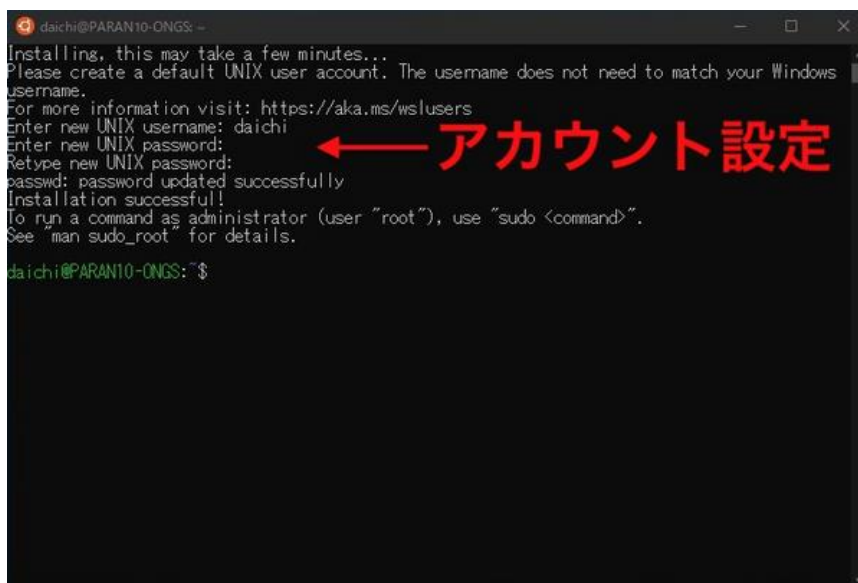
## インストールとセットアップ、そしてサンプル開発

まず、Microsoft StoreからUbuntu 18.04 LTSをインストールする。



Microsoft StoreからUbuntu 18.04 LTSをインストール

Ubuntu 18.04 LTSをインストールしたら、そのまま起動する。起動途中でユーザ名とパスワードの入力を求められるので、入力を行う。



Ubuntu 18.04 LTSを起動し、途中でユーザー名とパスワードを入力

シェルが起動してきたら、次のようなコマンドを入力してパッケージを最新の状態に更新するとともに、日本語環境のセットアップを行う。

## パッケージの更新と日本語環境のセットアップ

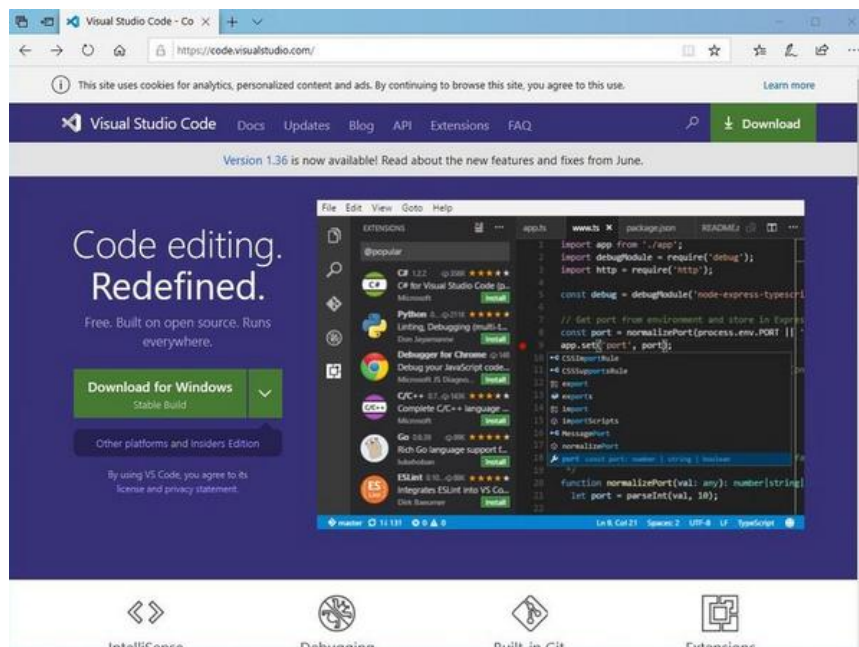
```
sudo apt update
sudo apt upgrade -y

sudo apt install -y language-pack-ja
sudo update-locale LANG=ja_JP.UTF8
```

## Python 3とPIPをインストール

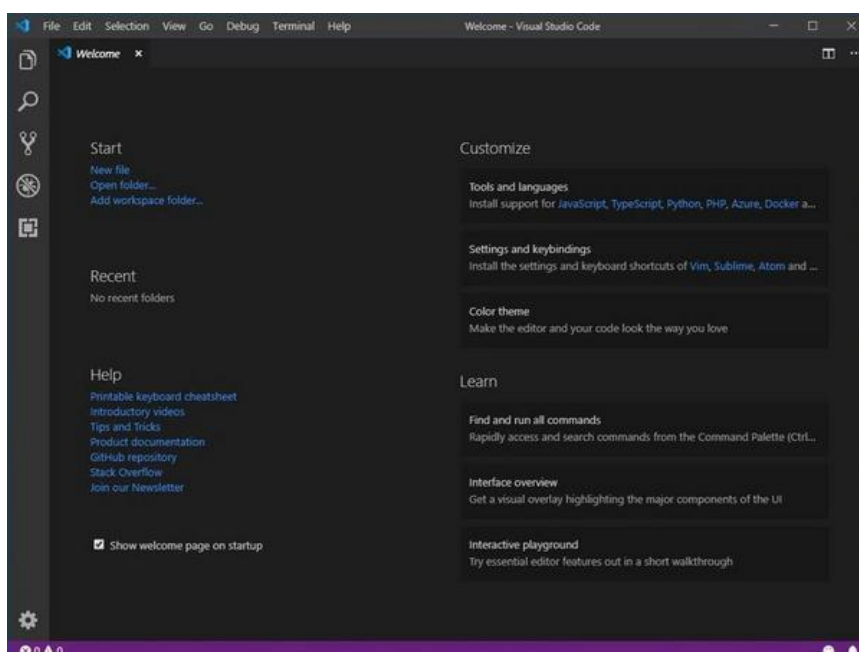
```
sudo apt install -y python3 python3-pip
```

ここまで準備したら、次はVisual Studio Codeをインストールする。「[Visual Studio Code - Code Editing, Redefined](#)」からダウンロードしてインストールする。



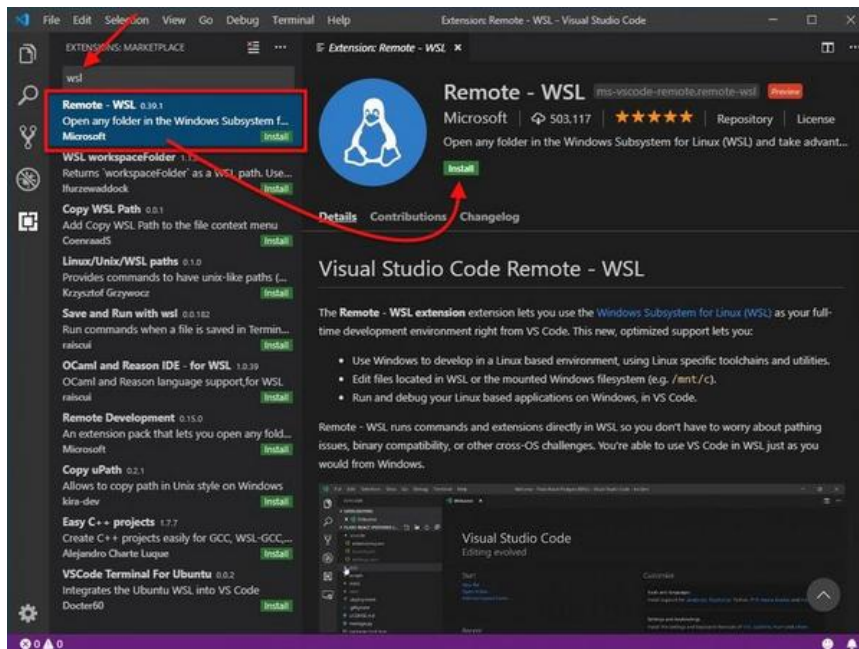
Visual Studio Code - Code Editing, RedefinedからVisual Studio Codeのダウンロードとインストールを実施

インストールしたらVisual Studio Codeを起動する。



Visual Studio Codeを起動

Visual Studio CodeのメニューからExtensionという項目を探してクリックする。次のようにエクステンションのエクスポージャー画面が表示されるので、左上のフィールドに「wsl」と入力して「Remote - WSL」を表示させる。「Remote - WSL」を選んだら「Install」ボタンをクリックしてRemote - WSLエクステンションのインストールを実施する。



Remote - WSLエクステンションをインストール

この段階でUbuntu 18.04 LTSに戻り、次のようにディレクトリの作成とPythonのソースコード作成を行う。

### Ubuntu 18.04 LTSでPythonのソースコードを用意

```
cd
mkdir helloWorld && cd helloWorld
echo 'print("hello from python running in Linux on windows!")' >> hello.py
```

ここで一旦Ubuntu 18.04 LTSのシェルを終了する。メニューからUbuntu 18.04 LTSを選択して再度Ubuntu 18.04 LTSを起動する。起動したら次のように、先程作成したPythonソースコードと同じディレクトリでcodeコマンドを実行する。

### 作成したPythonソースコードと同じディレクトリでcodeコマンドを実行

```
cd
cd helloWorld
code .
```



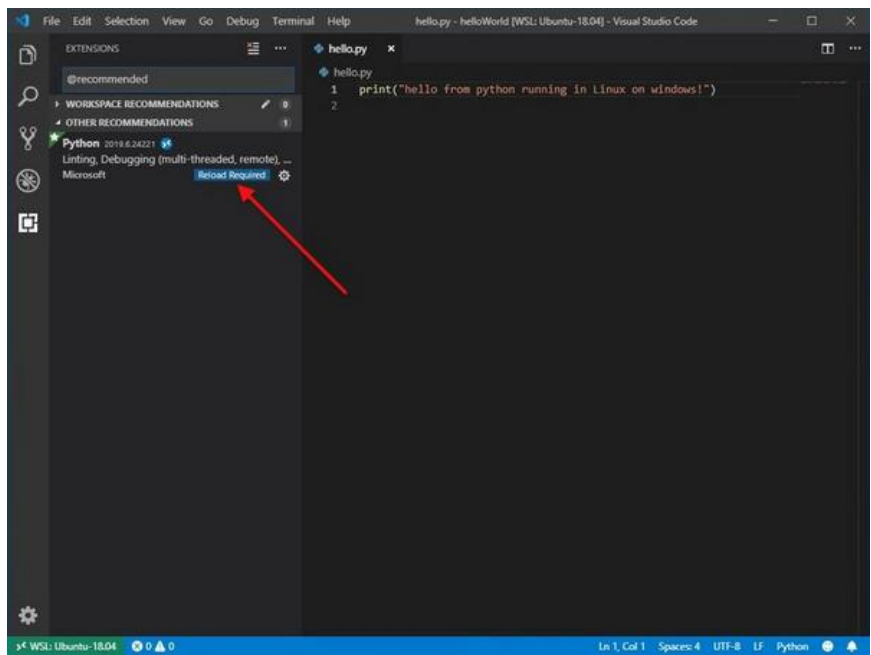
The screenshot shows the Visual Studio Code interface with a Python file named 'hello.py' open. The file content is as follows:

```
1 print('hello from python running in linux on windows!')
```

The status bar at the bottom indicates the file is saved and the Python extension is recommended. A red arrow points to the 'Install' button for the Python extension.

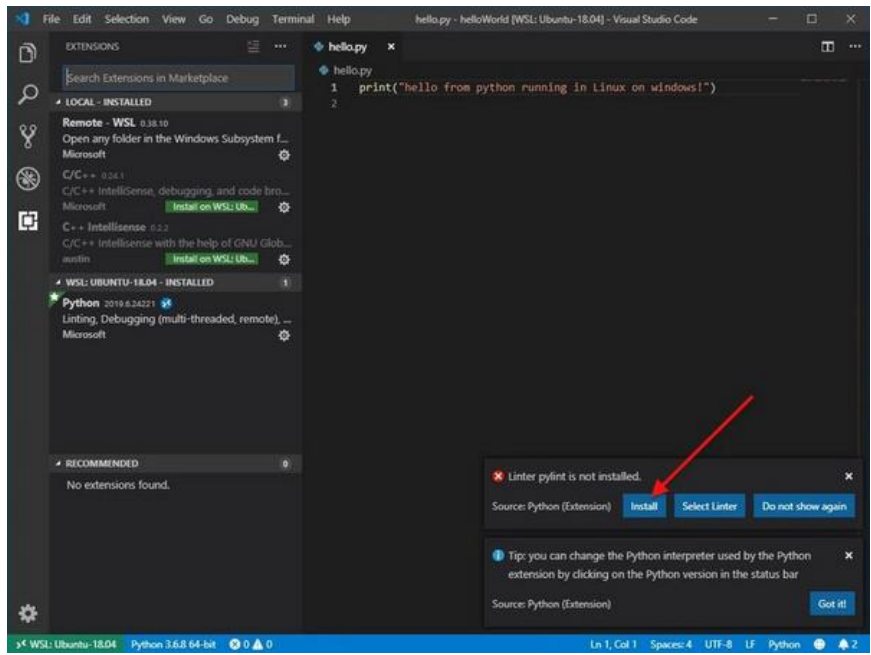
Pythonエクステンションのインストールが終わると再読み込みが必要だというメッセージが表示されるので、「Reload Required」をクリックして再読込を実施する。





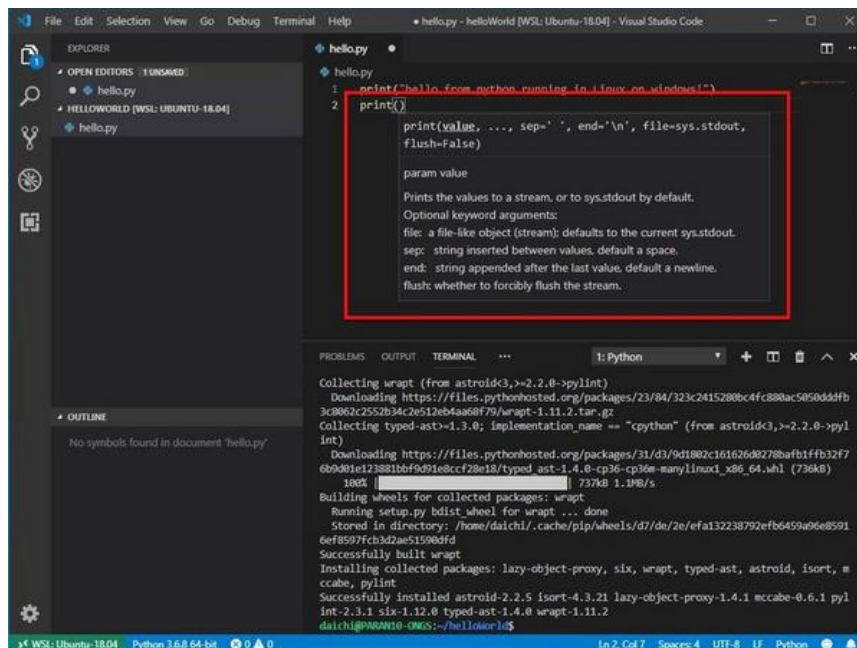
Pythonエクステンションインストール後は再読み込みが必要

すると、今度はpylintがインストールされていないので、インストールするようにメッセージが表示されるので、「Install」をクリックしてインストールを行う。



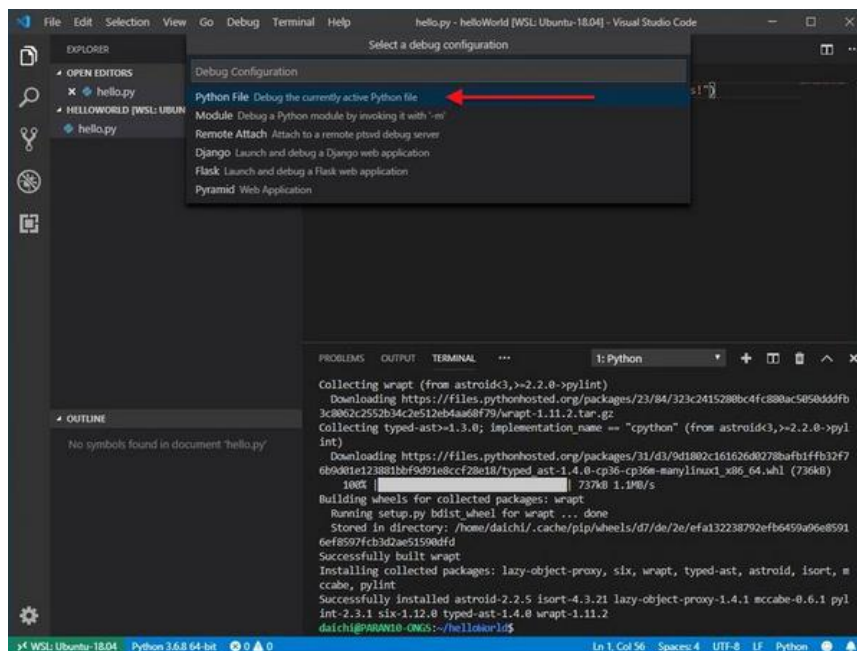
pylintをインストール

pylintのインストールが完了すると、例えば次のようにコーディング中に説明ドキュメントが表示されるようになる。



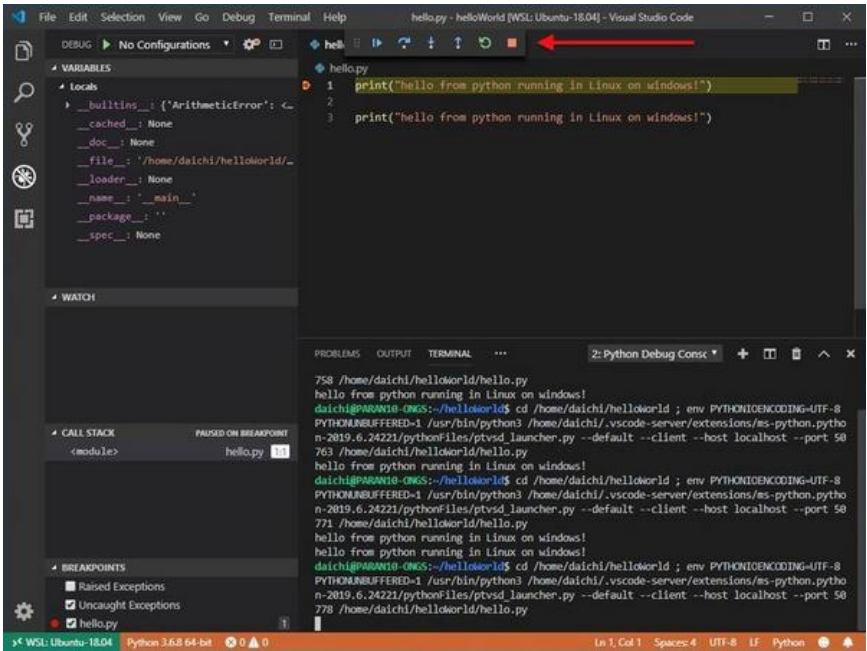
pylintインストール後に有効になる機能

ここまででセットアップは完了だ。早速作成したPythonソースコードを実行してみよう。DebugメニューからStart Debuggingを選択するか、F5キーを押す。次のようにデバッグ設定を聞かれるので、「Python File」を選択する。



F5でデバッグを開始し、設定として「Python File」を選択

次のように、右下に表示されている統合ターミナルで実装が実施され、エディタが側がそれに対応した動作を行ってくれる。ブレークポイントを設定すれば途中で止めてステップ実行することも可能だ。



デバッグ実行中のようす

こんな感じで、Windows 10でVisual Studio Codeを実行しながら、開発したPythonのソースコードはWSL経由で動作しているUbuntu 18.04 LTSで実行するといったことができる。実際にLinuxで開発を行うため、本番環境デプロイ時のパスの調整といった面倒な作業が発生しにくくなる。"いいとこ取り"した開発環境である。

## まずは最初の一步から

Windows 10でLinuxアプリケーションの開発を行うというのは、しばらくの間はトレンドになるように見える。デスクトップとしてはWindowsが8割から9割を占めているが、エッジサーバとしてはLinuxが使われることが多い。WindowsでLinuxアプリケーションの開発を行うというのは、現状に即しているのだ。

今回取り上げた記事はボリュームがほどよく、最初に取り掛かる資料として優れている。この記事で実際の感覚を掴んだら、もっと複雑な説明を行っている記事の内容を試してみるといったことができるようになると思う。ぜひ試してもらえればと思う。

※本記事は掲載時点の情報であり、最新のものと異なる場合があります。予めご了承ください。

### 連載一覧

- 第28回
- Visual Studio CodeとWSLでPython開発に挑戦してみよう
- 第27回
- ここでおさらい、WSL 1とWSL 2の違いとは？
- 第26回
- Visual Studio CodeとWSLを使って開発してみよう
- 第25回
- Windows Terminalプレビュー版登場、その注目点は？
- 第24回
- Vimより簡単なエディタ「nano」をカスタマイズしてみよう

目次ページへ