

このドキュメントの目的

Python,Django,virtualenv,VScodeの用意・準備を行う。

VScodeについては、この段階では『プロジェクトの編集が可能』となるところまでを目標とする

目次

1. Pythonのインストール
 2. virtualenvのインストールと準備
 3. Djangoのインストールと準備
 4. VScodeのインストールと事前準備
-

1. Pythonのインストール

コマンドプロンプトでバージョン確認

```
python -V
```

実行結果

インストールされている場合

```
Python 3.7.2
```

インストールされていない場合

```
'python' は、内部コマンドまたは外部コマンド、  
操作可能なプログラムまたはバッチ ファイルとして認識されていません。
```

インストールされていなければ、以下よりインストーラをダウンロードして実行する。

<https://www.python.org/downloads/>

次にpipをバージョンアップする。

```
pip install -U pip
```

プロキシ設定がある場合はプロキシオプションを用いる。

```
pip install -U pip --proxy=[指定のURL]
```

*以下、pipコマンドのプロキシオプションは割愛する。

実行結果

最終行に以下が表示される。

```
Successfully installed pip-x.x.x . . .
```

* pip以外にいくつか付随してインストールされることもある。

2. virtualenvのインストールと準備

Q. そもそもvirtualenvとは？

A. Pythonの仮想環境構築を補助するツール。

プロジェクトごとにPythonバージョンやモジュール・パッケージなどを管理できる。

まずはインストール。

```
pip install virtualenv
```

実行結果

最終行に以下が表示される。

```
Successfully installed virtualenv-x.x.x . . .
```

次に、仮想環境の作成を行う。

コマンドプロンプトで以下のコマンドを実行する。

cd [仮想環境を置きたいディレクトリ]

**virtualenv [仮想環境のディレクトリ名] **

例:ドキュメント直下に『env1』という仮想環境を立てる。

```
cd C:\Users\hoge\Documents
virtualenv env1
```

実行結果

最終行に以下が表示される。

```
Installing setuptools, pip, wheel...
done.
```

doneと表示されたら、仮想環境を使用してみる。

cd [仮想環境のディレクトリ名]

scripts\activate

例:先ほどの仮想環境『env1』に入る \

```
cd env1
scripts\activate
```

実行結果 \

```
(env1) C:\Users\hoge\Documents\env1>
```

上のように、頭に丸かっこで仮想環境のディレクトリ名が表示されていれば成功。
例えばこの状態でpipコマンドを実行しパッケージをインストールすると、
env1にのみそのパッケージが有効化される。
これにより、動作環境のパッケージやバージョン管理が楽に行えるようになる。

なお仮想環境から出たい場合は以下のコマンドで行える。

```
scripts\deactivate
```

3. Djangoのインストールと準備

Q. そもそもDjangoとは？

A. Pythonで実装されたWebフレームワークの1つ。
長所などの細かい話はここでは割愛する。

まずはインストール。

* ここではローカルから実行しているが、仮想環境上でもコマンドは同じ。

```
pip install django==X.X.X
```

* 必要に応じてパッケージ名の後ろにイコールを2つつけてバージョン指定すること。

イコールの後ろにバージョンを書かずに実行すると、現在入手可能なバージョン一覧が出力される。

実行結果

最終行に以下が表示される。

```
Successfully installed django-x.x.x pytz-x.xs qiparse-x.x.x
```

次に、仮想環境へフレームワークを適応させる。
コマンドプロンプトで以下のコマンドを実行する。

cd [仮想環境のディレクトリ名]

mkdir [任意のディレクトリ名]

cd [上記のディレクトリ名]

**django-admin startproject [プロジェクト名] **

例:env1配下に『Practice』というディレクトリを作り、
そこにDjangoを利用したプロジェクト『mybook』を作成する

```
cd C:\Users\hoge\Documents\env1\  
mkdir Practice  
cd Practice  
django-admin startproject mybook
```

正しくプロジェクトが作成できたか、webサーバを稼働させ確認する。

コマンドプロンプトで以下のコマンドを実行する。

cd [上記で作成したプロジェクトディレクトリ]

**python manage.py runserver **

例:プロジェクト『mybook』の起動及び確認を行う。

```
cd mybook
python manage.py runserver
```

実行結果 \

```
Watching for file changes with StatReloader
Performing system checks...
System check identified no issues (0 silenced).
April 21, 2020 - 15:58:53
Django version X.X.X, using settings 'mybook.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK. \
```

表示されているURLをブラウザで開き、Djangoのページが表示されていればフレームワークの有効化及びサーバ起動が正しく行えている。

4.VScodeのインストールと事前準備

正式名は『VisualStudioCode』。

インストール自体は公式サイトからインストーラをダウンロード・実行すればよい。

<https://code.visualstudio.com/>

インストールが終わったら起動し、大きく以下の4つの作業を行う。

1. VScodeの日本語化
2. Python及びDjangoに対応するための拡張機能をインストール
3. 作成しておいた仮想環境の読み込み
4. プロキシ設定

1. VScodeの日本語化

1. VScodeを起動し、『Ctrl』 + 『Shift』 + 『X』 キーを押下する。
2. 『Search Extensions in Marketplace』という入力フォームに『Japanese』と入力する。
3. 『Japanese Language Pack for Visual Studio Code』が見つかるので、それをインストールする。
4. 2の入力フォームを空にすると、『ENABLED』という項目が現れるので、先ほどの日本語化パックがあることを確認する。
5. 『Ctrl』 + 『Shift』 + 『P』を押下すると、VScode上部に入力フォームが現れるので『language』と入力する。
6. 検索結果の中から『Configure Display Language』を探して押下し、さらに『ja』を押下する。
7. VScodeを再起動するか問われるので再起動する。
8. VScode全体が日本語表示になっていれば成功。

2. Python及びDjangoに対応するための拡張機能をインストール

1. 『Ctrl』 + 『Shift』 + 『X』 キーを押下する。
2. 日本語化のときと同様に、入力フォームに『Python』 及び 『Django』 と入力し、各拡張機能をインストールする。
 - * 複数表示されることもあるが、DL数の多いものや評価数が高いものがおすすめ。
3. 入力フォームを空にして『ENABLED』 を表示させ、PythonとDjangoがあれば成功。

3. 作成しておいた仮想環境の読み込み

1. VScode左上の『ファイル』 から、『フォルダーを開く』を押下する。
2. 先ほど作成した仮想環境のディレクトリ(例で言うとenv1のこと)を選択し『フォルダーの選択』を押下する。
3. VScode左側に読み込んだディレクトリとその配下が表示されていれば成功。
 - * 隠れている場合は『Ctrl』 + 『Shift』 + 『E』 キーもしくは左側メニューにある『エクスプローラー』を押下すれば表示される。

4. プロキシ設定

* 環境によっては省略してよい。

1. VScode左上の『ファイル』 から『基本設定』 > 『設定』、もしくは『Ctrl』 + 『,』 キーを押下する。
2. 上部の『設定の検索』 入力フォームに『http:proxy』 と入力する。
3. 『Http:Proxy』 の入力フォームに、指定の設定を入力する。
 - * pipコマンドにつけていたproxyオプションに書いたものと同様だと思われる。

以上で、基本的な開発環境は完成。VScodeの詳細設定については次回解説。