



# WindowsとMacでPython 3.12開発環境を構築しよう

開発環境構築はプログラミングの第一歩です。このプレゼンテーションでは、WindowsとMacでPython 3.12の環境を簡単に作る方法をご紹介します。

WSL2の導入から仮想環境の構築まで、初心者にもわかりやすく解説します。



作成者：Hiroki Abe.

# Windows編：WSL2のインストールと設定

管理者権限でPowerShellを起動

スタートメニューからPowerShellを右クリックし、「管理者として実行」を選びます。

WSL機能の有効化

次のコマンドを実行します：

```
dism.exe /online /enable-feature /featurename:Microsoft-Windows-Subsystem-Linux /all /norestart
```

WSL2のインストール

続いて以下のコマンドを実行：

```
wsl --install -d Ubuntu
```

再起動後、Ubuntuが自動的に起動します。

WSL2の起動

Windowsのスタートメニューから「Ubuntu」を検索して起動するか、コマンドプロンプトやPowerShellでwslと入力して起動します。

# Windows編：WSL2上でのPython 3.12のインストール

## パッケージリストの更新

```
sudo apt update && sudo apt upgrade -y
```

必要な依存パッケージをインストールします：

```
sudo apt install -y build-essential libssl-dev zlib1g-dev  
libbz2-dev libreadline-dev libsqlite3-dev wget  
curl
```

## Python 3.12のインストール

```
sudo add-apt-repository ppa:deadsnakes/ppa  
sudo apt update  
sudo apt install python3.12 python3.12-venv  
python3.12-dev -y
```

インストール確認：

```
python3.12 --version
```



## Windows編：venvモジュールを使った仮想環境の作成と活用



### プロジェクトフォルダの作成

```
mkdir my_project  
cd my_project
```



### 仮想環境の作成

```
python3.12 -m venv venv
```



### 仮想環境の有効化

```
. venv/bin/activate
```

ターミナルの先頭に(venv)が表示されます。



### (オプション) パッケージのインストール

```
pip install pandas numpy matplotlib
```

requirements.txtの作成：

```
pip freeze > requirements.txt
```

# Mac編：HomebrewとPython 3.12のインストール

## 1 Homebrewのインストール

ターミナルを開き、次のコマンドを実行します：

```
/bin/bash -c "$(curl -fsSL  
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

## 2 Pythonのインストール

次のコマンドでPython 3.12をインストールします：

```
brew install python@3.12
```

## 3 PATHの設定

~/zshrcファイルに以下を追加します：

```
export PATH="/opt/homebrew/opt/python@3.12/bin:$PATH"
```

設定を反映させます：

```
source ~/.zshrc
```



# Mac編：Macでのvenv環境構築とパッケージ管理

## 仮想環境の作成

```
mkdir my_mac_project  
cd my_mac_project  
python3.12 -m venv venv
```

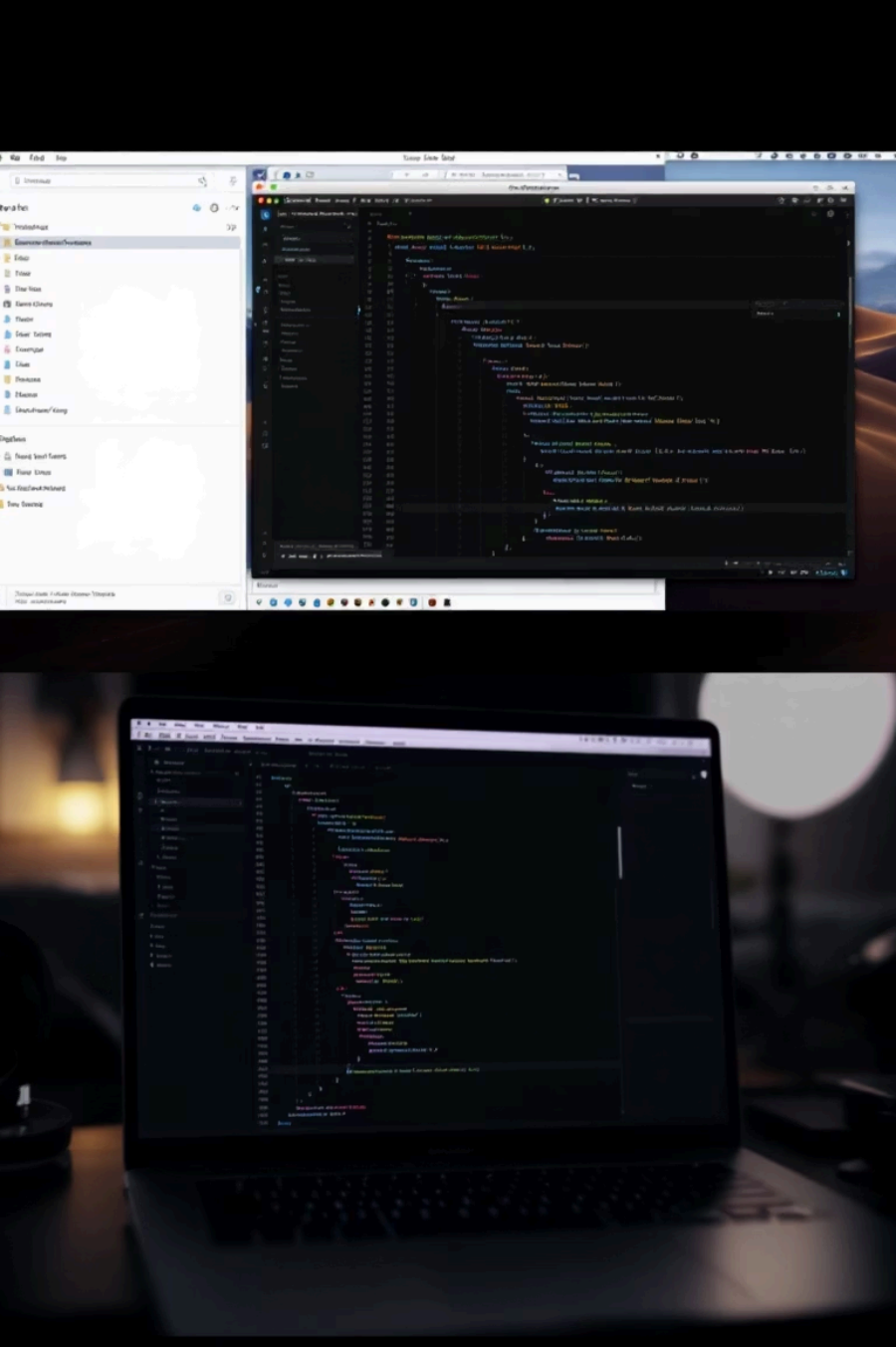
## 環境の有効化：

```
. venv/bin/activate
```

## パッケージ管理のベストプラクティス

- プロジェクトごとに仮想環境を作成する
- requirements.txtでパッケージを管理
- pipを最新に保つ: `pip install --upgrade pip`

複数プロジェクトのパッケージ競合を防ぎます。



# 両環境での統合開発環境(IDE)の設定 とおすすめツール

14

## Visual Studio Code

WindowsとMac両方で使える優れたエディタです。Python拡張機能をインストールし、インタープリターを仮想環境に設定します。



## PyCharm

より高度な機能が必要な場合のIDE。無料のCommunity版でも十分な機能があります。プロジェクト設定で仮想環境を指定できます。



## Jupyter Notebook

データ分析に最適です。仮想環境にインストールするには：

```
pip install jupyter
```



# Python that workflow

Life cycle in development for the Python in environment Python

## まとめ：効率的なPython開発のためのベストプラクティス

### 1 仮想環境を常に使用する

プロジェクトごとに独立した環境を作成し、依存関係の競合を防ぎましょう。

### 2 バージョン管理を活用する

Gitなどでコードとrequirements.txtを管理し、環境の再現性を確保します。

### 3 ドキュメントを読む習慣をつける

公式ドキュメントは最も信頼できる情報源です。特にライブラリの使い方は必ず確認しましょう。

### 4 定期的にアップデートする

セキュリティと新機能のためにPythonとパッケージを定期的に更新しましょう。



#### Code

Enables to the error only script logging



#### Test

Test code and script running and this is not on your hands



#### Test 2

For test code and test results



#### Package

Deliverable to the external systems to use testing going



#### Deploy

Save the script and code on the server



#### Step 3

- The code is the first time you coding for every context to test.
- The code is the first time you coding for every context to test.
- The code is the first time you coding for every context to test.

1

The code is the first time you coding for every context to test.

2

The code is the first time you coding for every context to test.

→

The code is the first time you coding for every context to test.

4

The code is the first time you coding for every context to test.



The code is the first time you coding for every context to test.