

雰囲気ですPythonを使っている人のためのPython再入門

- しんせいたろう
- 米国株とれーだー
- 雰囲気ですpython書いている(10年以上)
- 時系列データ分析(株、暗号資産、為替)、投資ストラテジー作成
- python の個別指導
- GTUGGirls
- Finpy
- 月刊Fintalk
- モグモグDjango

Python を確認

- install を確認
- which python
- where python

仮想環境構築

仮想環境とは

- ただのホルダ
- 仮想環境に入る=そこでつくったコピー環境を使う
- プロジェクトディレクトリ=レポジトリの感覚
- レポジトリがわからないひとにはただのホルダでオッケ
- 作成、削除、改変、なんでも出来る

構築手順

1. project dir 作成
2. cd project dir
3. `python -m venv .venv`
4. `.venv` ディレクトリ作成確認
5. 仮想環境に入る
 - (linux / mac) `source .venv/bin/activate`
 - (win) `venv\Script\activate`

以後、必ずするクセ付けて

- (linux / mac)

```
$ source .venv/bin/activate  
$ which python
```

- (win)

```
venv\Script\activate  
where python
```

ライブラリインストール

pip とは

ライブラリインストール

- `pip install ライブラリ名` するのは最初の一回だけ、`pip install -U pip`

```
touch requirements.txt
```

```
# linux / mac  
$ which python  
# windows  
$ where python
```

```
$ pip install -r requirements.txt
```


演習

1. ディレクトリ `gtugtest` 作成
2. その中に python の仮想環境を作成
3. 以下のライブラリを仮想環境下にインストール
4. `gtugtest` 削除

Anaconda / miniconda について

Python は jupyter notebook ではない

- jupyter を python と思ってるひとが多数
- jupyter でしか python を実行出来ないひと多数
- ヤメて

エラーは読んで

```
>>> 10/0
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ZeroDivisionError: division by zero
```

- エラーメッセージは下から上に読んでいく

python の概念,キーワード,知らないとツライ単語

- オブジェクト
- メソッド
- 関数
- クラス
- インスタンス
- スコープ
- モジュール

DRY

- 概念がわかってないとかけないPythonの哲学

部品を作ってループで回す

Pythonista がこのむ python らしい書き方

- 内包表記
- アンパック代入
- 高階関数 (lambda, map, filter)
- f-リテラル

test

python 概念と pythonist らしいコードの例

構造

- 私が「キレイ」だと思ったpythonのfile構造
- モジュールの説明も兼ねる
- `main.py` を実行ファイルとして、あとのファイルには関数群を作る

python はオブジェクト

- beautifulsoup でスクレイピングしたデータを
- pandas に流し込んで
- 可視化

VSCode の設定

```
{
  // PythonのPATHをワークスペースの仮想環境にする
  "python.pythonPath": "${workspaceFolder}/venv/bin/python",
  // 仮想環境にインストールしたファイルは監視対象から除外する
  "files.watcherExclude": {
    "**/venv/**": true
  },
  // リンタでPyLintは使わない
  "python.linting.pylintEnabled": false,
  // リンタでFlake8を使う
  "python.linting.flake8Enabled": true,
  // コードフォーマッタでBlackを使う
  "python.formatting.provider": "black",
  // Blackは貼り付け時の整形に対応していないので無効にする
  "editor.formatOnPaste": false,
  // 1行の文字数を88文字とする
  "python.linting.flake8Args": ["--max-line-length", "88"],
  // languageServerにPylanceを使う
  "python.languageServer": "Pylance",
  // Pylanceの型チェックをbasicにする
  "python.analysis.typeCheckingMode": "basic",
  // Pylanceの括弧補完を有効にする
  "python.analysis.completeFunctionParens": true,
}
```

VSCode の設定

```
{
  // PythonのPATHをワークスペースの仮想環境にする
  "python.pythonPath": "${workspaceFolder}\\venv\\Scripts\\python.exe",
  // 仮想環境にインストールしたファイルは監視対象から除外する
  "files.watcherExclude": {
    "**/venv/**": true
  },
  // リンタでPyLintは使わない
  "python.linting.pylintEnabled": false,
  // リンタでFlake8を使う
  "python.linting.flake8Enabled": true,
  // コードフォーマッタでBlackを使う
  "python.formatting.provider": "black",
  // Blackは貼り付け時の整形に対応していないので無効にする
  "editor.formatOnPaste": false,
  // 1行の文字数を88文字とする
  "python.linting.flake8Args": ["--max-line-length", "88"],
  // languageServerにPylanceを使う
  "python.languageServer": "Pylance",
  // Pylanceの型チェックをbasicにする
  "python.analysis.typeCheckingMode": "basic",
  // Pylanceの括弧補完を有効にする
  "python.analysis.completeFunctionParens": true
}
```

