

Flask のインストールと 基本 (Windows 上)

Python の Web フレーム
ワークの定番の 1 つ

金子邦彦



Flask

- Python のパッケージ
- **Web アプリケーションのフレームワーク**
- **Webサーバ**の機能を含む
- リクエストURLを解析し，処理と返答を行うプログラムを簡単に書くことができる.

<関連情報>

- **Flask のインストール:**
<https://www.kkaneko.jp/pro/webui/flask.html>

Flask のインストール手順 (Windows 上)

① 前準備として Python をインストール

<https://youtu.be/2MlVmx-yLM8>

② インストール

Window のコマンドプロンプトを管理者として実行

```
python -m pip install -U simplejson flask
```

Flask を使うプログラム例

```
from flask import Flask  
app = Flask(__name__)
```

リクエスト URL '/' に対する処理

```
@app.route('/')
```

```
def hello_world():  
    return 'Hello, World!'
```

```
if __name__ == '__main__':
```

Web サーバの起動. ポート番号 5000

```
app.run(debug=False, host="0.0.0.0", port=5000)
```

Flask を使うプログラム例

```
from flask import Flask  
app = Flask(__name__)
```

@app.route('/') リクエスト URL '/' に対する処理

```
def root():  
    return 'root'
```

@app.route('/hello') リクエスト URL '/hello' に対する処理

```
def hello():  
    return 'hello'
```

@app.route('/user/<username>') リクエスト URL '/user/<username>' に対する処理
これは /usr/1 /usr/2 などにマッチする
username は変数. マッチングの結果は変数に入る (/usr/1 に対しては, 1 が変数に入る)

```
def user(username):  
    return 'user %s' % username
```

if __name__ == "__main__":
 app.run(debug=False, host="0.0.0.0", port=5000)

Web サーバの起動. ポート番号 5000

- **Web サーバの機能:** HTTP メソッドと, リクエスト URL を受け取って処理と返答を行うサーバ
- **HTTP メソッド:** Web サーバに対する要求内容を示すもの. **GET** (取得), **POST** (送信), **PUT** (送信), **DELETE** (削除) がある.
- **Python の Flask:** リクエストURLを解析し, 処理と返答を行うプログラムを簡単に書くことができる.
- **Flask の Web サーバ:** 「`app.run(debug=False, host="0.0.0.0", port=5000)`」のように書くと, ポート 5000 で Web サーバが起動する.