

pythonで webサーバ

by ABE Hiroki aka hATrayflood

お前
誰？

by ABE Hiroki aka hATrayflood

自己紹介

- 名前 : アベヒロキ ([@hATrayflood](#))
- 職業 : 自宅サーバ管理者
- URL : rayflood.org/diary-temp/
- 地元 : 東部町

第100回

by ABE Hiroki aka hATrayflood

という記念すべき
回にも関わらず

空気読まずに
関係ないことを

by ABE Hiroki aka hATrayflood

やります

pythonで webサーバ

by ABE Hiroki aka hATrayflood

開発中に
こんなこと
ないすかね？

その1

index.html
jquery.js
script.js
data.json

```
$.getJSON("data.json", function(data){  
    data1 = data;  
});
```

ローカルファイルとして
開くとjsonが読めない
(ことがある)

そ の 2

ユーザ認証API のコールバック

by ABE Hiroki aka hATrayflood

twitter, facebook
google, (githubと ?)

テスト目的でも
ローカルファイルには
コールバックできない

http://localhost:8000/
などは可能

となるところここで
欲しくなるのは

webサーバー

by ABE Hiroki aka hATrayflood

けど既存の webサーバ 立てるの

by ABE Hiroki aka hATrayflood

メ トル い

by ABE Hiroki aka hATrayflood

IISとか
Apacheとか
nginxとか

コマンド一発で
起動して使いたい

そんな
ある？

あ る

by ABE Hiroki aka hATrayflood

```
python2 -m SimpleHTTPServer
```

```
python3 -m http.server
```

カレントディレクトリを そのままhttpで公開

by ABE Hiroki aka hATrayflood

/ もindex.html
を読んでもくれる

wsgji

by ABE Hiroki aka hATrayflood

単にhtml
を読むだけ
じゃなくて

by ABE Hiroki aka hATrayflood

pythonで
プログラム
動かしたい

by ABE Hiroki aka hATrayflood

そうになると
cgi的
な
仕組み？

IISとか
Apacheとか
nginxとか

設定
メドしい

設定なしで ソース一本で

by ABE Hiroki aka hATrayflood

コマンド一発で
起動して使いたい

そんな人
ある？

あ る

by ABE Hiroki aka hATrayflood


```
import wsgiref.simple_server

def application(environ, start_response):
    start_response("200 OK", [("Content-Type", "text/plain")])
    yield b"Hello World\n"

httpd = wsgiref.simple_server.make_server("", 8081, application)
httpd.serve_forever()
```

wsgirefは pythonの 標準ライブラリ

by ABE Hiroki aka hATrayflood

くまひ

pythonが
使えれば
確実に動く

by ABE Hiroki aka hATrayflood

え？

リクエストオブジェクト？

レスポンスオブジェクト？

2018/5/26 NSEG Vol.100

application()の引数environの中身:

```
SERVER_NAME na512e
GATEWAY_INTERFACE CGI/1.1
SERVER_PORT 8081
REMOTE_HOST
CONTENT_LENGTH
SCRIPT_NAME
SERVER_PROTOCOL HTTP/1.1
SERVER_SOFTWARE WSGIServer/0.2
REQUEST_METHOD GET
PATH_INFO /
QUERY_STRING
REMOTE_ADDR 127.0.0.1
CONTENT_TYPE text/plain
HTTP_HOST localhost:8081
HTTP_USER_AGENT Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:60.0) Gecko/20100101 Firefox/60.0
HTTP_ACCEPT text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE ja,en-US;q=0.7,en;q=0.3
HTTP_ACCEPT_ENCODING gzip, deflate
HTTP_DNT 1
HTTP_CONNECTION keep-alive
HTTP_UPGRADE_INSECURE_REQUESTS 1
HTTP_CACHE_CONTROL max-age=0
wsgi.input <_io.BufferedReader name=780>
wsgi.errors <_io.TextIOWrapper name="" mode='w' encoding='utf-8'>
wsgi.version (1, 0)
wsgi.run_once False
wsgi.url_scheme http
wsgi.multithread True
wsgi.multiprocess False
wsgi.file_wrapper
```

by ABE Hiroki aka hATrayflood

基本的には
昔のCGIと
同じやり方

by ABE Hiroki aka hATrayflood

environつまり 環境変数から 拾って使う

レスポンスは
これに直接書く

```
start_response("200 OK", [("Content-Type", "text/html")])
```

ボディは

```
yield b"<h1>タイトル</h1>"
```

または

```
string = "<p>%s</p>" % name  
yield string.encode()
```

あれ？
ルーティングは？
テンプレートは？
セッションは？

そんなものはない
(AAry

db(は) ?

標準ライブラリで
sqlite3が使える

```
import sqlite3  
db = sqlite3.connect("test.db")
```

え？ 他^のdbは？
モテ^リングは？
マイグ^レーシヨンは？

そんなものは(ry

wsgi
そのものは
webアプリ共通
インターフェース

必要最低限のもの
しか提供しない

素のwsggiでは
限界がある

by ABE Hiroki aka hATrayflood

階層化しない
数ページの
サイトなら可

階層化して
db使って
ログインする

フレームワーク の仕事

by ABE Hiroki aka hATrayflood

小さいもの

Bottle
Flask

by ABE Hiroki aka hATrayflood

```
from flask import *  
app = Flask(__name__)  
  
@app.route("/")  
def hello():  
    session["name"] = "abe"  
    return render_template("index.html", name="abe")
```


大きいもの

Django

MVCフルサポートの 全部入りフレームワーク ログイン機能や データ管理画面もある

by ABE Hiroki aka hATrayflood

使ったことない

Zope/Plone

Pylons/Pyramid

規模は違えど
どれもwsgi準拠

つまりwsggiが動けば
マネージドサービス
でも動かせる

heroku
google app engine
など

webサーバと 連携する

by ABE Hiroki aka hATrayflood

apacheで動かす

mod_wsgi

nginxで動かす

uwsgi

ただしuwsgiを起こす
コマンドが要る

IISで動かす

wfastcgi

らしい

mod_wsgiが一番
手がからないので
よく使う

最近の愛読書



by ABE Hiroki aka hATrayflood

・リンク集

twitter @hATrayflood

<http://twitter.com/hATrayflood>

開発用ローカルサーバを立ち上げる方法 Qiita@higuma

<https://qiita.com/higuma/items/b23ca9d96dac49999ab9>

Python2系でもPython3系でも簡易Web鯖をワンライナーで立てる - tututenの備忘録

<http://tututen.hatenablog.jp/entry/2014/01/09/121428>

簡潔な HTTP リクエストハンドラ — Python 2.7.15 ドキュメント

<https://docs.python.org/ja/2.7/library/simplehttpserver.html>

HTTP サーバ — Python 3.6.5 ドキュメント

<https://docs.python.org/ja/3.6/library/http.server.html>

WSGI ユーティリティとリファレンス実装 — Python 2.7.15 ドキュメント

<https://docs.python.org/ja/2.7/library/wsgiref.html>

WSGI ユーティリティとリファレンス実装 — Python 3.6.5 ドキュメント

<https://docs.python.org/ja/3.6/library/wsgiref.html>

Bottle: Python Web Framework

<http://bottlepy.org/docs/dev/>

Flask (A Python Microframework)

<http://flask.pocoo.org/>

The Web framework for perfectionists with deadlines | Django

<https://www.djangoproject.com/>

初心者がハマった、初めてのDjangoアプリのHerokuデプロイ Qiita@RyuSA

<https://qiita.com/RyuSA/items/0cbc7d5b0145585861a8>

Deploying Python Applications with Gunicorn | Heroku Dev Center

<https://devcenter.heroku.com/articles/python-gunicorn>

[Python] Flask を Google App Engine 上で動作させる Qiita@togoturns

<https://qiita.com/togoturns/items/579b0cf73370f4022f29>

Python ランタイム | App Engine flexible environment for Python docs

<https://cloud.google.com/appengine/docs/flexible/python/runtime>