

# 白鯖配信しよう！！

Twitter でログインできる配信ページをつくる by そーちゅー

p.4

# 集めた絵を整理するソフトを作りたい

NoSQL で遊んでみよう with Python by 鴉宮冷麺

p.8

# 二郎

ヤサイマシマシニンニクアブラカラカラ by 鴉宮冷麺

p.18

# 裸足のロメオ

怪文書 by にくだんご

p.24

# 写真集

Photo Kamiyukou by 紙湯

p.2,26

紙のジャンクウィキ

2017 冬





野良猫振り向き  
比較的健康そうな凛々しい感じの猫  
しかし、野良なためか警戒心は強く、終始近寄らせてくれなかった



水辺で木の実を食べるリス  
普段はすばしっこいが食べ物には目がないようで夢中になって食べていた



エナガの雛団子  
木の陰で団子になってるエナガの雛たち  
ご飯でも待ってるのかな



浅瀬で一休みするカルガモ  
全体的に地味な色合いのカルガモだが、  
青い羽根が際立っていいアクセントになっている



# 自鯖配信しよう！

Twitter でログインできる配信ページを作る

文 そーちょー



## 自分のサーバーから生放送したい



ですよね??? (断定)

そーちょーと申します。よろしくお願ひいたします。そんなことはどうでもよくてもう時間が無いので (切実)、本題に入りたいと思います。

最近友人の間で、というか世界規模で PUBG 流行ってますね！私も FPS ヘタクソなくせに買ったりしました。

PC でゲームをするといったら…………… ???

そうですね！ 配信したくなりますよね！！！！！！！！！！ (必然)

でもニコ生とかツイキャスとかそういうやつ使いたくないし…

やっぱりここは、周囲に無限に転がっている自分のサーバーを使わない手はありません！  
というわけで自分のサーバーで生配信環境を作っていきたいと思います。



## Nginx で RTMP & HLS 配信



世の中は広いもので (狭い) やっぱり我々と同じ事を考える人は必ず居ます。

たとえば

自家製 RTMP+HLS 配信のすすめ (その1) [goo.gl/eFcpYR](https://goo.gl/eFcpYR) ←短縮リンクはっておきます (ガバガバ)

あたりの手法で構築するとすぐに手元から配信できるようになります！ (丸投げ)

じゃあ自分のサーバーで配信するメリットってなんや？っていったら、

- ・ 画質上げ放題 (プロバイダのアップロード上限に注意)
- ・ 延長とかそういうのいらない
- ・ 配信内容の制約がない (法には触れるなよ?)

という感じですね！最高ですね！！

じゃあ何が不満なんだ？というと

- ・ コメント欄がほしい
- ・ USTREAM みたいにコメントしたらツイートしてほしい
- ・ 認証機能がほしい

くらいでしょうか。

自分でなんとかしろよ！

というわけでなんとかしました (早)



## ガバガバな手っ取り早い解決方法



- ・ コメント欄がほしい  
→ USTREAM のコメント欄貼り付けたれ
- ・ USTREAM みたいにコメントしたらツイートしてほしい  
→ USTREAM のコメント欄貼り付けたれ
- ・ 認証機能がほしい  
→ ドヤ顔 basic 認証！

うわあ……

さっきのサイトみたいに nginx で構築して上の解決方法を適用したページが右です！！

うわあ…… (2 回目)

こんなんでも一応使えるっちゃ使えるので、一定の支持 (筆者の周りの一部から) は得ていました！！

なんとかしたいですね！  
しました！！！！

次ページへ。



▲ガバガバ視聴画面

## ⚙️ Twitter でログインできる配信ページを作る (サブタイ回収) ⚙️

- コメント欄がほしい  
→ どうせ Twitter 見るからとりあえずいいや (妥協)
- USTREAM みたいにコメントしたらツイートしてほしい  
→ 実装しましょう！！
- 認証機能がほしい  
→ Twitter でログインして、フォロワーだったら配信画面をみせるようにしよう！

やりたいことは固まりました！ではとりかかりましょう！

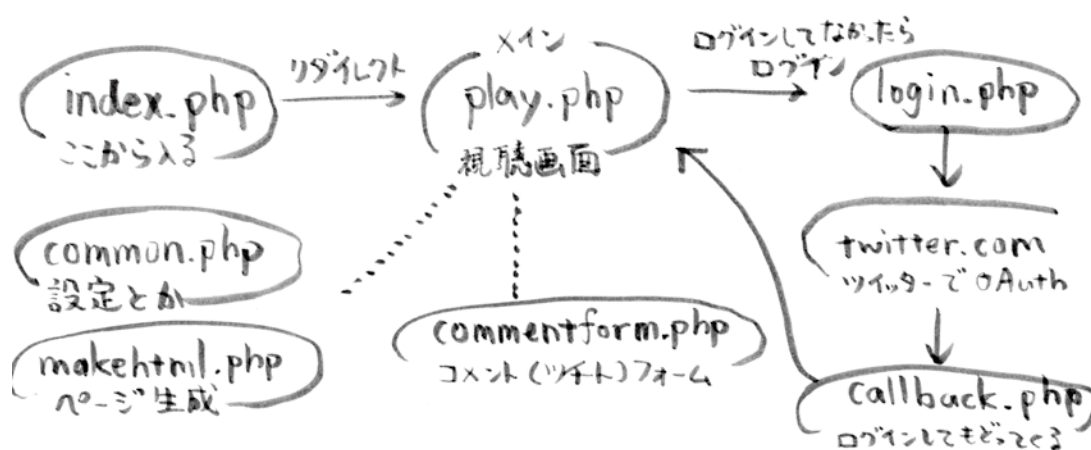
## ⚙️ システム設計 () ⚙️

言語は PHP で作っていいと思います！ 今更 PHP ? とか言わないで…  
ちなみに理由としては

1. 適当に xampp 環境に突っ込むだけで動く！
2. (筆者が) ほんの少しだけ分かる！

です！  
とくに 1 が重要で、ポンッて入れたら勝手に動くのは楽チンで良いですね！

さて実際に作るにあたり、どういう構造にしましょうか？  
ということで考えたのは以下の通りです！



▲フローチャートでもないし何やこれ？ファイル一覧？

ちなみに .php って出るのはクソダサいので .htaccess で拡張子消すことにしました

## ⚙️ 作る (迫真) ⚙️

作りました！作るにあたって

AWS EC2 と Nginx-rtmp-module と Video.js と OBS を使ってライブストリーミング配信 (RTMP と HLS)

<https://qiita.com/nuxaxtu/items/250ca4fb4e102fa425d7>

【PHP】新 TwitterOAuth でログイン機能を実装する

<https://qiita.com/sofpyon/items/982fe3a9cceb8702867>

などを参考にさせて頂きました！

出来上がったページは  
→の通りです！



なお、フォロワーで  
ない場合はその旨  
メッセージが表示され  
ます！

## ⚙️ 使い方&おわりに&今後の予定 (?) ⚙️

くう^~疲れましたw これにて実装完了です！

とりあえず WEB の JUNKWIKI (bsb.jp/jw) にて配布予定なので、興味を持たれた方は是非やってみてください！

使い方としては common.php に twitter の consumer key とか配信者の Twitter ID とか RTMP URL とか設定する項目があるのでパパッと埋めたら終わり！！簡単！！

今後の予定としては

- ・ いらんやろって思ってた見送りしたコメント欄の復活
- ・ ガバガバデザインをなんとかする (スマホ対応とか)

記事も内容もガバガバですみません！許してください！なんでも許してください！  
ここまで読んでくれてありがとナス！！！！

おわり

# 集めた絵を整理するソフトを作りたい

## 一気にプロトタイプまで作ってしまおうの回

文 鴉宮冷麺



### はじめまして。



鴉宮冷麺と申します。

何か怪文書をこしらえてくれ、と某氏に言われてしまったので、この怪文書を書いています。普段は会社員をしながら絵を書き、音楽を作り、プログラムを作り、プラモを作り、ドール用布服を作り、写真を撮り、時には動物園に通いながら死んでいく生き物です。よろしくお願いします。



### で、この怪文書は？



この文書では、何かプログラムを作ります。

以前（一昨年くらい）に旧ブログでこんなことを言っていました。

余談ですが

ファイルサーバの中には作ったものが入ったりあとは集めたえっちな画像が山のようにあるわけですが、その管理に今 Picasa を使ってます。タグを付けて分類するだけです。

んで分類したものをベッドの上で見たいわけですよ、iPod（宅内での移動端末）で。

今考えてるのは、Picasa でつけたタグ（メタデータとして入るので）を収集、データベース化して PHP から呼び出したりなんざできたらいいなって。ランダム表示出来たり、ね。

Picasa のデータベースを直接覗ければいいんですがね…

もうちょっとソフトウェア的に仕様をまとめたら設計してちょちょっと作ってみようかななんて思ってる。

画像整理のために簡単なプログラム組んだりしたんだけど、何も考えずに作り始めたら継ぎ足し継ぎ足しのクワソーみたいなソースしかできなくてこれもうわかんねえな。

（可読性は）ないです。

仕事の関係でプログラム組むのでその練習もあるんで、やってみようかなって、予定は未定

もう三時なので寝ます。

おやすみなさい

何を言っているのか、よくわかりませんね？

まあつまり、「サーバ内の画像を http を経由してどこからでも見れるようにしたい、ついでにタグ付けしたい」と言っている。気がする。

で、今回はこの本をきっかけに一気にある程度動くものまで作ってしまおう、という感じです。

はてさて、どこまでできるかな？



### まずは簡単に要件をば



- ・ 集めた絵にタグを付けて管理したい
- ・ LAN 内どこからどんなデバイスからでも見れるようにしたい
- ・ ランダム表示ができるようにしたいできれば現状のファイルサーバに手を入れたくはない

こんなかんじ。で、  
設計方針はこんな感じ

- ・ ファイルサーバと同居 or 別サーバにファイルサーバをマウントする
- ・ Webserver に Alias でドキュメントルートからファイルサーバの中身が見えるようにしておく
- ・ ファイルサーバ内の管理したい画像のファイル名、パスを couchdb に入れる
- ・ ついでにタグとか作者も紐付け
- ・ API を叩いてタグから画像の情報を呼び出す
- ・ Webserver で GET する（謎）

ということで？ とりあえず適当に作っていきます。  
今回の目標は「画像を表示する」です。ではヒュイゴ



### 作製



#### couchDB に動作確認用データを入れる

まずは DB を作ります。

この辺は前回ブログの記事でセットアップしたとおり CouchDB です。

Futon(WebGUI) から簡単に設定でき…るはずですが、今回はコマンドラインで。どうせソフトだと思わないしね。DB 側の GUI。

curl で PUT を叩けば DB ができます（次ページ）。

```
dev@devserver:~$ curl -X PUT http://127.0.0.1:5984/imagelist
{"ok":true}
dev@devserver:~$ curl -X GET localhost:5984/_all_dbs
["_global_changes","_replicator","_users","imagelist"]
dev@devserver:~$ curl -X GET localhost:5984/imagelist
{"db_name":"imagelist","update_seq":"0-g1AAAAFTeJzLYWBg4MhgTmEQTM4vTc5ISXlwNDLXMwBCwxygFFMiQ5L8____sxIZ8ChKUgCSSfaE1TmA1MUTVpcAUldPUF0eC5BkaABSQKXziVG7AKJ2PzFqD0DU3idG7QOIWpB7swBegI6O","sizes":{"file":33952,"external":0,"active":0},"purge_seq":0,"other":{"data_size":0},"doc_del_count":0,"doc_count":0,"disk_size":33952,"disk_format_version":6,"data_size":0,"compact_running":false,"cluster":{"q":8,"n":1,"w":1,"r":1},"instance_start_time":0}
```

imagelist という DB を作成しました。

ここにテスト用のデータを入れていきます。

couchdb はドキュメント思考データベースなので、まずは json 形式で入れたいデータを作る必要がある。

ちょっとした文法ミスで「invalid!」って怒られる json という形式だが、そんなに難しく考える必要はない。jsonlint というサイトで文法の確認ができるので、ちびちび確認しながら進めよう。

文法の基本は {"key":"value"}。value を入れ子にすることもできる。

```
{"name":"DSC04723.png","filepath":"/mnt/data/images/testimage/DSC04723.png","type":"png"}
```

こんな感じ。path はサンプル用に用意した右の写真の指定しています。

こちらの写真ですが、撮影場所はアクアパーク品川のイルカショーです。

入場料は少々高いですが、工夫を凝らした展示が楽しめます。

夏に行きましたが、花火モチーフにした展示はなかなか良かったです。

ポップな音楽に合わせたイルカショーはノリ

ノリな感じで楽しいです。あとなんか曲が音ゲーっぽい。

一方、一般的な水族館動物園を期待していくと物足りなさを感じました。

なんというか、どちらかと言うと子連れ、カップル向けの水族館かな、という感じです。



▲イルカジャンプ

SONY α 55 SIGMA DC 18-250mm F3.5-6.3 OS HSM 1/80 F5.6 ISO1600

話が逸れました

先程の json をボディにして、POST してやる。PUT でもいいが、一意な id を指定する必要があるので、めんどくさい。POST は付けないで投げると勝手につけてくれる。

```
dev@devserver:~$ curl -X POST localhost:5984/imagelist -H 'content-type:application/json' -d '{"name":"DSC04723.png","filepath":"/mnt/data/images/testimage/DSC04723.png","type":"png"}'
{"ok":true,"id":"6a68cff2f67da8389cef0bb0ca000a3b","rev":"1-4d553ee9df660edbea5f7cb76af3ddc7"}
```

レスポンスで id が返ってくるので、id を指定して確認に GET してみる

```
dev@devserver:~$ curl -X GET localhost:5984/imagelist
{"db_name":"imagelist","update_seq":"1-g1AAAAFTeJzLYWBg4MhgTmEQTM4vTc5ISXlwNDLXMwBCwxygFFMiQ5L8____sxIZ8ChKUgCSSfaE1TmA1MUTVpcAUldPUF0eC5BkaABSQKXziVG7AKJ2PzFqD0DU3idG7QOIWqB7GbMAXoRejw","sizes":{"file":136384,"external":145,"active":382},"purge_seq":0,"other":{"data_size":145},"doc_del_count":0,"doc_count":1,"disk_size":136384,"disk_format_version":6,"data_size":382,"compact_running":false,"cluster":{"q":8,"n":1,"w":1,"r":1},"instance_start_time":0}
dev@devserver:~$ curl -X GET localhost:5984/imagelist/6a68cff2f67da8389cef0bb0ca000a3b
{"_id":"6a68cff2f67da8389cef0bb0ca000a3b","_rev":"1-4d553ee9df660edbea5f7cb76af3ddc7","name":"DSC04723.png","filepath":"/mnt/data/images/testimage/DSC04723.png","type":"png"}
```

できた。

## request from python

今度は python からこのリクエストを打ってみる。

```
import json
import requests

print 'hellow-python'

print 'getapi'

url = 'http://localhost:5984/imagelist/'
key = '6a68cff2f67da8389cef0bb0ca000a3b'

result = requests.get(url+key)

print result.text
```

とりあえず GET して結果を出す感じのパイソン。ID は決め打ち。

API を叩くだけなら、python の request モジュールを使えば簡単に叩ける

```
dev@devserver:~/PhotoView$ ls
getapi.py
dev@devserver:~/PhotoView$ python getapi.py
hellow-python
getapi
{"_id":"6a68cff2f67da8389cef0bb0ca000a3b","_rev":"1-4d553ee9df660edbea5f7cb76af3ddc7","name":"DSC04723.png","filepath":"/mnt/data/images/testimage/DSC04723.png","type":"png"}
```

```
dev@devserver:~/PhotoView$
```

とれた



## 未知の領域 ～web アプリケーション～

画像の情報が取れたということで、今度は web から叩いて画像をブラウザに表示できるようにしたい。

というわけでここからは未知の領域、ウェブアプリ作りの世界に入ります

ここからはこちら [https://qiita.com/yoshizaki\\_kkgk/items/f3c79e38b802c7fc8863](https://qiita.com/yoshizaki_kkgk/items/f3c79e38b802c7fc8863) のサイトを参考に作っていきます。

python そのものの準備はできているので、django のインストールから、ですが。

まずは django のインストール

```
dev@devserver:~/PhotoView$ sudo pip install django
The directory '/home/dev/.cache/pip/http' or its parent directory is not owned by the current user and the cache has been disabled. Please check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
The directory '/home/dev/.cache/pip' or its parent directory is not owned by the current user and caching wheels has been disabled. check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
Collecting django
  Downloading Django-2.0.tar.gz (8.0MB)
    100% |????????????????????????????????| 8.0MB 147kB/s
Complete output from command python setup.py egg_info:
Traceback (most recent call last):
  File "<string>", line 1, in <module>
  File "/tmp/pip-build-rHQ0nk/django/setup.py", line 32, in <module>
    version = __import__('django').get_version()
  File "/tmp/pip-build-rHQ0nk/django/django/__init__.py", line 1, in <module>
    from django.utils.version import get_version
  File "/tmp/pip-build-rHQ0nk/django/django/utils/version.py", line 61, in <module>
    @functools.lru_cache()
AttributeError: 'module' object has no attribute 'lru_cache'

Command "python setup.py egg_info" failed with error code 1 in /tmp/pip-build-rHQ0nk/django/
```

…うまく入りませんね。では、参考どおりに 1.8 を指定

```
dev@devserver:~/PhotoView$ sudo pip install django==1.8
The directory '/home/dev/.cache/pip/http' or its parent directory is not owned by the current user and the cache has been disabled. Please check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
The directory '/home/dev/.cache/pip' or its parent directory is not owned by the current user and caching wheels has been disabled. check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
Collecting django==1.8
  Downloading Django-1.8-py2.py3-none-any.whl (6.2MB)
    100% |????????????????????????????????| 6.2MB 251kB/s
Installing collected packages: django
Successfully installed django-1.8
dev@devserver:~/PhotoView$ ls
```

```
getapi.py
dev@devserver:~/PhotoView$ django-admin startproject photoview
dev@devserver:~/PhotoView$ ls
getapi.py photoview
dev@devserver:~/PhotoView$ cd photoview/
dev@devserver:~/PhotoView/photoview$ ls
manage.py photoview
dev@devserver:~/PhotoView/photoview$
```

入って、プロジェクトを作るとこまで来ました

参考サイトですとこのあと設定して DB を設定して起動になっていますが、今回 DB は couchdb なので、setting.py の該当部分はコメントアウトしておきます

```
dev@devserver:~/PhotoView$ python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).
December 20, 2017 - 23:51:19
Django version 1.8, using settings 'photoview.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[20/Dec/2017 23:53:34]"GET / HTTP/1.1" 200 1942
[20/Dec/2017 23:53:34]"GET /favicon.ico HTTP/1.1" 404 1939
```



とりあえず起動しました。  
これで環境はよし。

▲ Django 起動確認

## Yo! お前の Python! まだ API! 直叩き !?

さて、ここまで毎回 API を叩く度に URL を生成したりしてめんどくさかったのが、便利ツールを導入します。

couchdb-python さんです

<https://github.com/djc/couchdb-python>

python 用に API をラッピングしてくれる便利なやつ。

日本語ドキュメントがどこにもないので英語を読みながら作っていきます。

…これだけで一記事がかけけるのでは？ 今度かこう

まずはインストール… (次ページ)

```
dev@devserver:~$ git clone https://github.com/djc/couchdb-python.git
Cloning into 'couchdb-python'...
remote: Counting objects: 3132, done.
remote: Total 3132 (delta 0), reused 0 (delta 0), pack-reused 3132
Receiving objects: 100% (3132/3132), 971.88 KiB | 43.00 KiB/s, done.
Resolving deltas: 100% (1948/1948), done.
Checking connectivity... done.
dev@devserver:~$ cd couchdb-python/
dev@devserver:~/couchdb-python$ ls
COPYING ChangeLog.rst MANIFEST.in Makefile README.rst RELEASING.rst couchdb doc perftest.py setup.cfg setup.py
tox.ini
dev@devserver:~/couchdb-python$ python setup.py --help
Common commands: (see '--help-commands' for more)

  setup.py build    will build the package underneath 'build/'
  setup.py install  will install the package

Global options:
--verbose (-v)    run verbosely (default)
--quiet (-q)      run quietly (turns verbosity off)
--dry-run (-n)    don't actually do anything
--help (-h)       show detailed help message
--no-user-cfg     ignore pydistutils.cfg in your home directory
--command-packages list of packages that provide distutils commands

Information display options (just display information, ignore any commands)
--help-commands  list all available commands
--name           print package name
--version (-V)   print package version
--fullname       print <package name>.<version>
--author         print the author's name
--author-email   print the author's email address
--maintainer     print the maintainer's name
--maintainer-email print the maintainer's email address
--contact        print the maintainer's name if known, else the author's
--contact-email  print the maintainer's email address if known, else the
                  author's
--url            print the URL for this package
--license        print the license of the package
--licence        alias for --license
--description    print the package description
--long-description print the long package description
--platforms      print the list of platforms
--classifiers    print the list of classifiers
--keywords       print the list of keywords
--provides       print the list of packages/modules provided
--requires       print the list of packages/modules required
--obsoletes      print the list of packages/modules made obsolete

usage: setup.py [global_opts] cmd1 [cmd1_opts] [cmd2 [cmd2_opts] ...]
   or: setup.py --help [cmd1 cmd2 ...]
   or: setup.py --help-commands
   or: setup.py cmd --help

dev@devserver:~/couchdb-python$ python setup.py build
running build
running build_py
creating build
(中略)
copying couchdb/tests/tools.py -> build/lib.linux-x86_64-2.7/couchdb/tests
copying couchdb/tests/__init__.py -> build/lib.linux-x86_64-2.7/couchdb/tests
dev@devserver:~/couchdb-python$ sudo python setup.py install
[sudo] のパスワード:
running install
running bdist_egg
running egg_info
(中略)
Installed /usr/local/lib/python2.7/dist-packages/CouchDB-1.1.1dev_r0-py2.7.egg
Processing dependencies for CouchDB==1.1.1dev_r0
Finished processing dependencies for CouchDB==1.1.1dev_r0
dev@devserver:~/couchdb-python$
```

これで入ったはず。試しに一発叩いてみる。

```
dev@devserver:~/couchdb-python$ python
Python 2.7.12 (default, Nov 20 2017, 18:23:56)
[GCC 5.4.0 20160609] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import couchdb
>>> couch = couchdb.Server()
>>> couch = couchdb.Server('http://localhost:5984/')
File "<stdin>", line 1
    couch = couchdb.Server('http://localhost:5984/')
    ^
IndentationError: unexpected indent
>>> couch = couchdb.Server('http://localhost:5984/')
>>> db['6a68cff2f67da8389cef0bb0ca000a3b']
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'db' is not defined
>>> db = couch['imagelist']
>>> db['6a68cff2f67da8389cef0bb0ca000a3b']
<Document u'6a68cff2f67da8389cef0bb0ca000a3b@u'1-4d553ee9df660edbea5f7cb76af3ddc7' {u'type': u'png',
u'name': u'DSC04723.png', u'filepath': u'/mnt/data/images/testimage/DSC04723.png'}>
>>> for id in db:
...     print id
...
6a68cff2f67da8389cef0bb0ca000a3b
>>>
```

ゴリゴリ API を叩かなくてもドキュメントがとれました。

さて、こいつを使って、さっきの getapi.py と同じ動作をやってみます。

```
import json
import requests
import couchdb

print 'hellow-python'
print 'wrapapi'

couch = couchdb.Server('http://localhost:5984/')

db = couch['imagelist']

for id in db:
    print id
    items = db.get(id)
    print items
    print items.id
    print items['name']
```

```
dev@devserver:~/PhotoView$ python wrapapi.py
hellow-python
wrapapi
6a68cff2f67da8389cef0bb0ca000a3b
<Document u'6a68cff2f67da8389cef0bb0ca000a3b@u'1-4d553ee9df660edbea5f7cb76af3ddc7'
{u'type': u'png', u'name': u'DSC04723.png', u'filepath':
u'/mnt/data/images/testimage/DSC04723.png'}>
```

はい。

簡単ですね。



## ラストパート

簡単になったところで、コイツをウェブアプリにします。

とりあえず、Django さんでアプリを作製、設定。

こんなかんじ→

```
from django.http.response import HttpResponseRedirect
import json
import requests
import couchdb

# Create your views here.

couch = couchdb.Server('http://localhost:5984/')

db = couch['imagelist']

def image_view(request):
    items = db.get('6a68cf2f67da8389cef0bb0ca000a3b')
    itemtext = items['name']
    return HttpResponseRedirect(itemtext)
```

```
from django.http.response import HttpResponseRedirect
import json
import requests
import couchdb

# Create your views here.

couch = couchdb.Server('http://localhost:5984/')

db = couch['imagelist']

def image_view(request):
    items = db.get('6a68cf2f67da8389cef0bb0ca000a3b')
    itemname = items['name']
    itemdir = items['filepath']
    srcdir = itemdir.replace('/mnt/data/images/testimage/', '/images/images/')

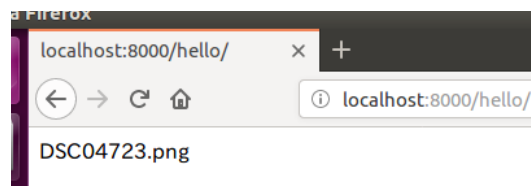
    header = '<html><body>imagetests<br></body></html>'

    responsehtml = header + srcdir + footer

    return HttpResponseRedirect(responsehtml)
```

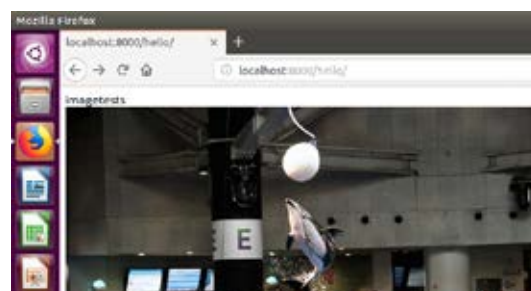
```
dev@devserver:~/PhotoView$ python manage.py startapp hello
dev@devserver:~/PhotoView$ ls
getapi.py hello manage.py photoview wrapapi.py
dev@devserver:~/PhotoView$ cd hello/
dev@devserver:~/PhotoView/hello$ ls
__init__.py admin.py migrations models.py tests.py views.py
dev@devserver:~/PhotoView$ vi photoview/settings.py
以下省略
```

←コイツを動かしてみると…



ここまでくればあとは目標である「画像を表示する」まで一直線。

ゴリゴリゴリゴリゴリゴリ…



デエソ

できました



## 振り返り



というわけで、途中からかなーり端折りましたが、目標までたどり着きました。

DB に場所を聞いてただ表示する（しかもハードコーディング）だけですが、コレができればあとは Python を好き勝手いじればできると思っています。

そんなに甘くないと思います。

今回 DB に一件しか入っていませんでしたが、次は複数件入れて、複数件のデータ取得と、出力を行いたいところです。

また、python の中で html のコードを生成してそれをそのまま投げつけている感じだったりするので、テンプレート使ったりして、もう少しスマートなコードにしていきたいところです。

あと、今のところコメントレベルでも日本語が入ると実行時にエラーが出てしまうので、なんとかしたい。



## おわりに



さて、目に見えて動く部分まで作れると達成感がありますね

なんというか、日々ちびちび作っていくより、時間を撮って一気に進めるとよくすすみますね。

仕事柄 python をいじることはよくあるんですが、だいたいテストコードか、コマンドラインで実行して何かするくらいの物しか書いていないので、こういう「目に見えて動くもの」をいじれるのは新鮮だったり。

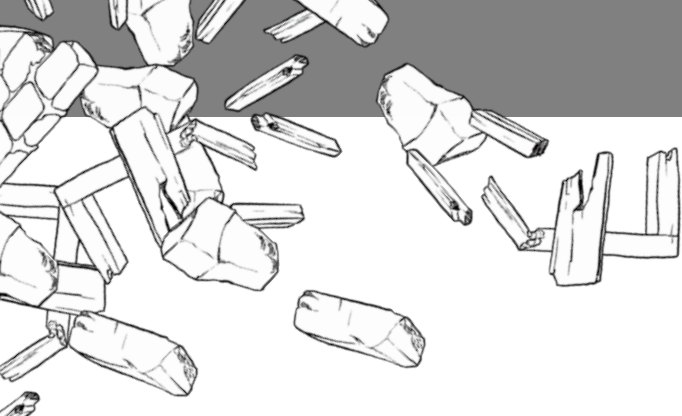
そんな感じです。

さて、次回の予定は、振り返りでも上げたように「複数レコードを呼び出して表示する」「python から直レスポンスではなく、html を通す」というところでしょうか。

ちなみに、コレを書いているのはとっくにメ切を過ぎたあとです。

PUBG やってたのが行けないんですが。

おわり



## 二郎

### 文 鴉宮冷麺

#### ⚙️ はじめまして。 ⚙️

鴉宮冷麺と申します。

何か怪文書をこしらえてくれ、と某氏に言われてしまったので、この怪文書を書いています。普段は会社員をしながら絵を書き、音楽を作り、プログラムを作り、プラモを作り、ドール用布服を作り、写真を撮り、時には動物園に通いながら死んでいく生き物です。よろしくお願いします。

#### ⚙️ で、この怪文書は？ ⚙️

行ったことのあるラーメン二郎を覚えている範囲で書いていきます。思い出程度なのであまり参考にならないかもしれません。と思います。当時の写真も見つかった分だけ載せます。画質はケータイ+リサイズなのでお察しだけ、雰囲気。油の量とかを感じ取っていただければ。トッピングは原則「全部マシ」コールとさせていただきます。

あと、味は完全に主観（と思い出補正）です。「俺の感想と違うぞ！」と言われても困ります。僕の舌なんで…

#### ⚙️ 二郎の思い出 ⚙️

##### 大宮店（閉店）

自分の二郎の基本になっている二郎。

野菜が茹でたてホクホクなのはほんと良かった。マシにして唐辛子とカラメで食べる野菜は美味かった。

そしてショウガコール。寒い日とか病み上がりに食べると体が温まってとてもよかった。

豚のブレは大きかった気がする。大当たりはトロトロの塊がドカンと乗っていた。運がいいと端っこの方の切り落としがドーンと乗っていた。ハズレはパサパサゴワゴワ。

麺は他の二郎よりさらに太めで固め、特にコールドしなくても麺の中心に芯が残る感じ。アブラにも程よく味が付いていて、塊で、それでいて控えめに乗ってくれるのがホント好きだった。

あと店長の日か店員の日か、で味が変わった。店長の日が好きだった。

偶に小麦粉研究会ってあって、麺に使う小麦粉を変えて、無化調かつ豚がブランド豚（何か忘れた）になる日があった。一回しか行けたことはないが、優しい味（二郎比）でアレはアレでよかった。



##### 神田神保町店

なんか漬れるんじゃないか！？って話があったが移転するだけらしい。

妙に麺の量が多い。

ブレが少なくそこそこ美味しい。



##### 新宿小滝橋店

以前はあまり評判が良くなかったお店。

実際味は薄い野菜はふにゃふにゃだし麺はブニョブニョ。かといって硬めコールすると硬すぎ。豚は…思い出せない程度の感じ。

最近美味しくなったと聞か…マシにはなったがまだコレジャナイ感がある。

味はまあ良くなったが、豚を改善してくれ豚を。ただ、三田行ったときに気づいたのは、この麺は本店に似ている。だからなんだって話だが…

##### 新宿歌舞伎町店

以前はあまり評判が (ry

自分が初めて行ったのは移転後なので、移転前のことはわからない。

今は別段悪くないと感じる。

あと、カウンターにラー油があるのはなかなかおもしろいと思う。香りも変わって新感覚。ただしブレは大きい。この前いったら小滝橋が出てきた。



##### 池袋東口店

あまり評判は良くないみたいだが個人的には嫌いじゃない。

…よく覚えてないけど

##### 立川店（休業）

野菜がめちゃくちゃ多いイメージがあった。実際多かった。うっかりして豚を残してしまった事がある。

麺はちょっと少なめだった…というか野菜メイン

##### 赤羽店（閉店）

美味しかったんだけどなあ…

全体的に豚が美味しかった記憶。

店員さん（店長の奥さんかな？）がナイスマダムだった。

奥さんが修行して再開して？

##### 札幌店

仲間内だと「二郎食べるために北海道行ったやべーやつ」と言われるがそうじゃないぞ！クマ牧場と旭山動物園が目当てだったんだぞ！まあ二郎も旅程にしっかり入れてましたが。なんというか、平均的な二郎だった気がする。よく覚えてない。

##### 府中店

なんかアブラが美味しかった記憶。アブラだけじゃなくてスープのだしに使う香味野菜もちょこっと混じってて美味だった。

あと、カウンターの雑巾があぶらべっちょりで触れたものではなかった。

そういえば、店長さんの体調不良説が流れてきてたけど、大丈夫だったのかしら？

##### 品川店

覚えてないが、微妙だった記憶。

##### 京急川崎店

隣の人が食べる前にインスリン注射をしてるのがショッキングだった。そこまでして食いたいか。俺でもそうするわ。

味は覚えてない。



### 湘南藤沢店

江ノ島旅行の帰りに寄った思い出。  
味は覚えていない。店に近くの絵画教室作のおっきいラーメンのデッサンがあった記憶。

### JR 西口蒲田店

Pio の帰りによる所  
昔の小滝橋ほどじゃないが、首をかしげる味。

### 新小金井街道店

小金井公園の帰りによく寄っている。  
日替わりで色々やってくれる（辛めとか酸っぱめとか）ところがいい。  
ここのおすすめはつけ麺。ごま油が効いてとても美味しい。  
これ、ラーメンにも入らないかなって思ってた  
ら、偶に「つけ味」というのをやってくれた。（つけ麺の味付けのラーメン）  
某ほぼ毎氏はここの近辺に住んでいたという噂がある。



### ひばりヶ丘店

二郎にしては上品過ぎる。いやすごい旨いんだけど。  
店長さんがフレンドリー。



### 荻窪店

美味しい。  
しかし荻窪を食った後に砂コン行くとスコアが出ない。NAZE



### 亀戸店

麺は柔め。豚が柔らかくて美味しかった記憶。



### 川越店

美味しかった記憶。  
豚が味がしっかりしていた。なんか、大宮を思い出す味と麺。  
しかし夏バテした胃に豚増しはきつかった



### 三田本店

聖地。今の勤め先のすぐ近くなので現状ホーム  
と言えるかもしれない。残業した後だと確実に  
麺切れしている。  
今のところ息子さんの担当する夜の部しか行けていないが、いろんな二郎のいろんなところを  
合わせたキメラ的な感じがする。いや逆か。  
少なくとも美味しい。豚が美味しい。大宮っぽかった。野菜はしんなり。  
麺は結構柔らかか。  
あと机と椅子が少々ボロい。高さが合って無くて少々食べにくい気がする。



## 🌀 インスパイアの思い出 🌀

### ラーメンエース

某大生憩いの場。というかこれ以外にまともな飯屋がない。  
なんというか、かなり美味しい。ほとんどブレないうえに常に当たり状態。ヤバい。  
麺は比較的固めで、スープは味濃いめのやつ。  
期間限定メニューもよくやっているし、これがまた美味しい。  
あと汁なしとつけ麺常設でいつでも食べられる。  
あと、生唐辛子が結構好きだった。  
また行きたい…が、遠いなあ。八王子の山の中は。



### 八王子田田

みょーににんにくが臭かった。  
ポイントカード制度があったが、店員さんがにんにく盛った手でそのままハンコ押すもんだからポイントカードまでにんにく臭かった思い出。  
つけ麺頼んだら丼3つ（汁と具、麺、野菜と豚）で出てきたのは驚いた。





## ラーメン大各店

有名な大。  
コールの「多め」と「マシ」で量が違ううえに  
アブラとニンニクとカラメにも適用される  
うっかりすると立川の野菜マシに背脂を塗りたくった感じのやつが出てくる。  
背脂チャッチャ系。豚がちょっと微妙感がある。



## サブローラーメン (閉店)

浪人生時代に大晦日の予備校帰りに行った思い出。  
なぜか唐揚げのお持ち帰りをやっていた (が時間が遅くて終わっていた)。  
味は覚えていないが、コールは無くてニンニクは机の上にあるものを自由に入れる形になっていた。  
次に行ったときには店長体調不良で長期休業、その次に行ったときは告知もなく潰れていた。



## 野郎ラーメン

野菜炒め。  
豚野郎と呼ばれたことはない。  
秋葉原店はなぜかアイスクリームを売っている。



## いごっそう

元三田の店員さんがやっているお店。  
言われてみればたしかに本店っぽさがある。  
ラーメンは若干やわ麺で味薄め、野菜は柔らかか。  
あと塩とかマー油とか、種類がある。  
ここは何と言っても豚めしが美味い。細かく刻んだ豚とカラメを混ぜ込んだご飯。うまくないわけがない。  
ラーメン自体の量が少ない (本家比) からラーメンライスできる。なおカロリー  
店長さんはアマチュアランナーらしく、東京マラソン前後はお休みになる。応援してます。



## メガガンジャ

メガ! という感じ。よく覚えていない。

## ジャンクガレッジ

汁なし特化インスパイア。チーズとマヨはヤバイって。中毒性が。

## 影武者

以前行った時、普通のラーメンが量が少なく行く気がなかったが、久しぶりに行ったら肉塊に押しつぶされた。  
なんだあれ。旨いが腹に入らんぞあの肉塊。



## 麺通

ふらりと立ち寄ったらまさかインスパイアだとは...



## 特麺コツ丁ラーメン

インスパイアではないが記載。  
なんでかという、店員ジロリアンだろって感じだから。  
何の変哲もない豚骨系ラーメンに見えるが、麺とか味とかニンニクコールに「二郎好きです」感が見え隠れする。  
普通に美味しいラーメン屋さんなので、タゲワン帰りに通うか...

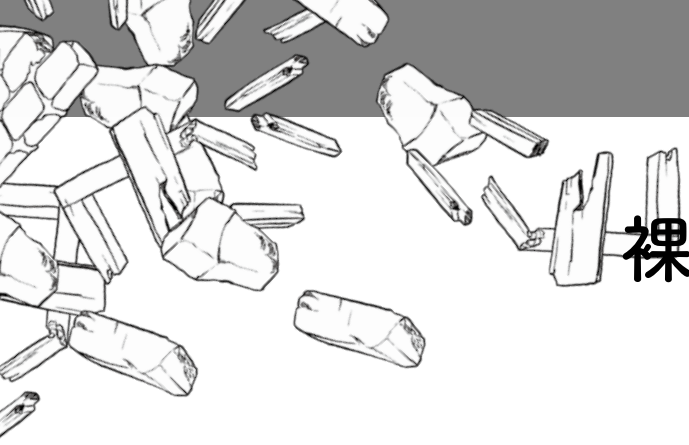


## おわり



異常、怪文書もとい行ったことのある二郎 & インスパイアの感想というか思い出でした。

おわり



# 裸足のロメオ

## 文にくだんご

僕が食パンでオナニーをしたのは東京に引っ越してすぐだった。今まで実家暮らしだった男が突然一人暮らしを始める。そういった場合男は絶対とんでもない事を考えるのだ。栃木の田んぼだらけを抜け出し都会に進出した僕は高揚感を隠せない。つい先日はこんにやくを使ってみた。昔の人はよく考えたものだと感心をした。痒くなったけど。

こんにやくで吐精をする快感を得た僕は、次なるターゲットを探していた。安価でどこにでも売っていて、肉棒を突っ込める食品、食パンである。あのやわらかな超熟を突き破った時、きつと僕を導いてくれる。と、考えていたのだが事後に残ったのは後悔だけだった。まったく感触がない。陰茎にまとわりついたパンくずを洗うお風呂場で、僕はどこで間違ってしまったんだと思案した。

僕の高校は四方をドブ川の檻に囲まれた公立男子校だ。なんでも昔県庁がそこにあって、周りを堀で囲んでいたらしい。ドブ川だ。

新学期、僕は体育館で嘔吐した。春休みに一歩も外に出ず、引きこもりライフを満喫していた時に急に高校が始まり体力測定だから。懸念していたあだ名がゲローマンになることはなかったが、精神的ダメージは大きかった。そこそこの頭の良い高校だったので、どうせ不細工ガリ勉しかいないと思っていた。しかしそこにそろったのはスポーツ万能成績優秀イケメンだ。許せない。

男子高校には華がない。ついこの間まで男女が入り混じった空間に通っていたのに、急に男だけになるのだ。気が狂う。そんな高校でも僕は小学校時代から友達で、たまにメールのやり取りをしている女子がいた。Iって名前だ。メールの内容はテストの話とか、アニメの話とか…。そんな女の子から届いたメール。

『今週の金曜日 Rちゃんと幸楽苑のラーメン食べにいくのだけど君も一緒に食べに行かない?』

断る理由がない。Iは幸楽苑でバイトをしているって前に言っていたな。僕は友達を1人誘い、約束の時間に幸楽苑に行った。女性陣も一人増やして3人で来るらしい。

Iが新たに連れてくるとか言ったのは通称ヤギュウと呼ばれた不細工女だ。中学時代に同じクラスだった事がある。あの衝撃的なあだ名を忘れはしない。しかし今はヤギュウの事はどうでもいい、僕はIとお話しがしたいんだ。

とりあえず席を確保し待機。だが、来ない。注文もしないで席に座りっぱなしはマズい。とりあえず餃子を注文。ゆっくり食った餃子も食べ終わったところ、彼女らは来た。でも、僕たちの席には座ってくれなかった。一体何が起きているのか理解ができない。僕はお水を取りに行く時にチラッと言葉を交わし、交流終了。一緒に食べるとは一体。一緒とは一体。同じ建物に居るだけでは一緒に食べているとは言わな

い。頼んだ醤油ラーメンはいつもよりショッパイ。

家に帰って僕は何を間違っていたのかメールを読み直した。僕が友達を1人連れてきてもいいかと聞いた回答に答えがあった。

『じゃあ呼んでえ』

ヤギュウさんが嫌らしいから席とかは一緒に座れないけど近くの席にしょーよ』

『じゃあ呼んでえ』の後の文章を完全に見落としていた。何だ、これは。嫌って、何だ。ふざけんな。全部はヤギュウさんが悪かったのか。僕はそんなにヤギュウさんから嫌われていたのか。もう少し僕の気持ちに配慮した表現は無かったのか。僕は誘われた立場なのにやっぱり一緒に席は嫌って、おかしいじゃん。そもそも女性とお話をするのにラーメン屋って、いくら高校生とはいえ、もう少しマトモな場所でしょう普通。

それ以降、僕は女性とまったく話もメールもする機会もないまま、成人式も面倒で行かなかったし。童貞はソーブランドで卒業したし。

Iは僕の事をどう思っていたのだろうか。好意を向けられていると、そういうシーンは今まで何度かあった気がする。少なくとも、ラーメン屋には誘ってくれた。あの頃僕は痩せていたから。もう、元には戻らない僕の青春と贅肉。







2016 横田基地友好祭 A-10  
日本にはあまり姿を見せなかった A-10 だが、その勇姿はやはり重厚感のあるものだった  
GAU-8 も存在感抜群



2016 入間基地航空祭ブルーインパルス  
6 機編隊で会場に進入してくる様は圧巻



2017 入間航空祭 逆光アタック  
入間航空祭は太陽のある方角を飛行機が飛ぶのでこういうこともある…効\*~効\*-



2017 入間航空祭 全力帰投  
F-15J のアフターバーナーふかしての帰投  
こう見るとノズル部分は結構メカメカしい



そーちょー  
がばがばおじさん。



にくだんご  
今どき人権放棄。



紙湯  
ただの神。



鴉宮冷麺  
ジロリアン。



「紙の JUNKWIKI 2017 冬」

発行日 : 2017 年 12 月 29 日

発行者 : そーちょー

サークル名 : JUNKWIKI

印刷所 : ジャンクプリンター

発行イベント名 : コミックマーケット 93