

Django

[[Python](#)][[AngularJS x Django](#)]

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

練習

■ インストール

- ・ [インストール](#)
- ・ [運用環境の構築](#)

■ 最初のアプリケーション 1 (環境設定 ~ モデルの作成)

- ・ [最初のアプリケーション 1 \(環境設定 ~ モデルの作成 \)](#)

■ 最初のアプリケーション 2 (Admin サイトの構築)

- ・ [最初のアプリケーション 2 \(Admin サイトの構築\)](#)

■ 最初のアプリケーション 3 (View の作成)

- ・ [最初のアプリケーション 3 \(View の作成\)](#)

■ 運用環境の構築

- ・ [Django 運用環境の構築](#)
- ・ [Django 運用環境の構築 1.5.1](#)

Django APIs

■ Django APIs(View,Model,Forms,Template)

- ・ [Django APIs](#)

■ Django Model

- ・ [Django Model](#)

■ Django テンプレート

- ・ [Django テンプレート](#)

■ Django ミドルウェア

- ・ [Django ミドルウェア](#)

■ Django 備忘

- ・ [Django 備忘](#)

■ Django Admin

- ・ <http://docs.djangoproject.jp/en/latest/ref/contrib/admin/index.html>

Eclipse

■ Eclipse から Django

- ・ [Eclipse から Django](#)

Link

■ The Django Book

- ・ <http://www.djangobook.com/>

■ The Django Book(日本語訳)

- ・ <http://djangobook-ja.appspot.com/>

■ Django チュートリアル (Codezine)

- ・ <http://codezine.jp/article/detail/4065?p=1>

Tips

バージョンの確認

```
>>> import django
>>> django.VERSION
```

リクエストオブジェクトとレスポンスオブジェクト

- ・ [リクエストオブジェクトとレスポンスオブジェクト](#)

File アップロード

- ・ [Django File アップロード例](#)

静的ファイルを提供

- ・ [Django 静的ファイルを提供する](#)

モデル

■ 多対多関係の実装

- ・ [Django 多対多関係の実装](#)

■ 日付の比較

- ・ 日付の比較を行う

クエリ

- ・ Django クエリ

JSON

■ Django のモデル

```
from django.core import serializers
# シリアライズ
users = User.objects.all()
print serializers.serialize("json", users)

# デシリアライズ
for item in serializers.deserialize("json", users_json):
    print str(item)
```

■ Python オブジェクト

```
import json
# シリアライズ
l = [1,2,3]
print json.dumps(l)

# デシリアライズ
l = json.loads("[1,2,3]")
```

ロギング

■ Django ロギング設定

- ・ Django ロギング設定
- ・ Django 生成される SQL をロギングする

■ SQL のロギング

- ・ <http://www.dabapps.com/blog/logging-sql-queries-django-13/>

```
>>> import logging
>>> l = logging.getLogger('django.db.backends')
>>> l.setLevel(logging.DEBUG)
>>> l.addHandler(logging.StreamHandler())
>>> for b in Bookmark.objects.all():
...     print b.page.url
...
http://google.co.jp
http://yahoo.co.jp
(0.000 ) SELECT "Bookmark_bookmark"."id", "Bookmark_bookmark"."page_id",
"Bookmark_bookmark"."user_id" FROM "Bookmark_bookmark"; args=()
(0.000 ) SELECT "Bookmark_page"."id", "Bookmark_page"."url" FROM "Bookmark_page" WHERE
"Bookmark_page"."id" = 1 ; args=(1,)
(0.004 ) SELECT "Bookmark_page"."id", "Bookmark_page"."url" FROM "Bookmark_page" WHERE
"Bookmark_page"."id" = 2 ; args=(2,)
```

■ Django のテストで SQL をロギングする

- ・ セットアップでロガーを有効にする

```

from django.conf import settings
from django.test import TestCase
from django.test.client import Client

class SimpleTest(TestCase):
    @staticmethod
    def setUpClass():
        settings.DEBUG = True
        import logging
        l = logging.getLogger('django.db.backends')
        l.setLevel(logging.DEBUG)
        l.addHandler(logging.StreamHandler())

```

jQuery

■ jQuery から配列を送り Django で受け取る

- ・ jQuery から配列を送り Django で受け取る

■ jQuery UI の Autocomplete を Django で実装

- ・ jQuery UI の Autocomplete を Django で実装

■ 開発サーバー

- ・ Django 開発サーバーに他ホストから接続

AngularJS

■ AngularJS Django の CSRF トークンを送信するように config で設定する

認証

- ・ Django から python-twitter で OAuth 認証して Twitter API を使う

デプロイ

■ CentOS

- ・ ServersMan(CentOS 6) に Django 環境を構築する (1) Python2.7 のインストールまで
- ・ Django を CentOS7 で実行する

■ AWS

- ・ AWS Elastic Beanstalk Django アプリケーション環境を CentOS6.5 上に PyDev を使って構築する (1) Django 起動まで

デバッグ

■ デバッグモードを利用

views

```

from django.conf import settings

```

```
def ag_basic(request):
    t = loader.get_template('test.html')
    print settings.DEBUG
    c = RequestContext(request,
        {'DEBUG': settings.DEBUG})
    return HttpResponse(t.render(c))
```

template

```
{% if DEBUG %}
DEBUG
{% else %}
RELEASE
{% endif %}
```

■ Django 1.7.1 のユニットテストを PyDev で実行すると AppRegistryNotReady が発生する

- ・ Django 1.7.1 のユニットテストを PyDev で実行すると AppRegistryNotReady が発生する