

初めてのプロジェクトを作ろう

環境の構築、おつかれさまでした。

いよいよDjangoでの開発をしていきます！まずはプロジェクトを作るところからはじめましょう。

プロジェクトとアプリケーション

Djangoで開発するときには、「プロジェクト」と呼ばれるディレクトリと、「アプリケーション」と呼ばれるディレクトリの大きく分けて二つのディレクトリを管理することになります。

この二つについて、簡単に説明しますね。実際に作って見ないと具体的なイメージが湧かないと思いますので今は完璧に理解できなくても大丈夫です。

まず、「アプリケーション」とは、ウェブサービスに必要な機能を実現するための部分のことです。例えば、「ブログをウェブページに表示させる機能」や「自分の好きなブログをお気に入り登録する機能」といった、なにか特定の役割を持つ機能を担っている部分のことです。

そして「プロジェクト」とは、ウェブサービス全体に関する設定やそのウェブサービスの各機能を構成する「アプリケーション」を一つにまとめたものです。基本的に、ウェブサービスを一つ作るときには、Djangoプロジェクトが一つと、そのプロジェクトに対応する複数のアプリケーションがあることになります。

さっそく、Djangoのプロジェクトを作ってみましょう。

まずは、1つ前のレッスンで作った「DjangoBros」ディレクトリに移動して、仮想環境を有効にしてください。

~/DjangoBros/

```
$ source djangobros_venv/bin/activate  
(djangobros_venv) $
```

今回は、Djangoでブログ管理サイトを作るので、「django_blog」という名前のプロジェクトを作ります。カレントディレクトリが「DjangoBros」であることを確認して、以下のコマンドを実行してください。

※プロジェクトを作成

~/DjangoBros/

```
$ django-admin startproject django_blog
```

これで「django_blog」というプロジェクトが作成されました。1つ下の階層に作成された「django_blog」という名前のディレクトリに移動して、lsコマンドでその中のディレクトリ構成を確認してください。

~/DjangoBros/

```
$ cd django_blog
$ ls
```

基本的なディレクトリ構成

ディレクトリを確認して見ると、以下のような構成になっています。

~/DjangoBros/django_blog/

```
django_blog
manage.py
```

プロジェクト名と同じ `django_blog` というディレクトリと、`manage.py` というファイルがありますね。`.py` の拡張子がついたファイルは、Python形式で書かれたファイルであることを示しています。

まず、manage.pyですが、こちらはその名の通りDjangoプロジェクトをmanage[管理・運営]する時に使うファイルです。サーバーを立ち上げたり、プロジェクトの管理者情報を作成する時に使います。こちらのファイルの中身を編集することは基本的にありません。

django_blogはこのプロジェクトのPythonパッケージです。中には4つのファイルが含まれています。

1. **init.py:** django_blogがPythonパッケージであることを示すための空ファイルです。こちらを変更することはありません
2. **settings.py:** Djangoプロジェクトの設定ファイルです。今後様々な場面で利用します。
3. **urls.py:** ウェブサイトの各ページのURLを設定するファイルです。
4. **wsgi.py:** サーバーの設定などを行うファイルです。

少し難しく感じるかもしれませんが、今後必要な場面で説明を加えるので、現時点では各ファイルの役割を覚える必要はありません。

初めてのDjangoページを表示しよう

それでは初めてのDjangoプロジェクトを自分のパソコンで起動してみましょう。

はじめに、プロジェクトのルートディレクトリに移動してください。ルートディレクトリとは、1番上の階層という意味です。現在、DjangoBrosディレクトリの中にdjango_blogとい

うプロジェクトを作っていますので、(DjangoBros/django_blog) がプロジェクトのルートディレクトリとなります。

~/DjangoBros/django_blog/

```
$ pwd
/DjangoBros/django_blog
```

ルートディレクトリに移動したら、lsコマンドで1つ下の階層に `manage.py` ファイルがあることを確認して、`python manage.py runserver` コマンドを打ってください。これはローカルサーバーを起動するコマンドです。

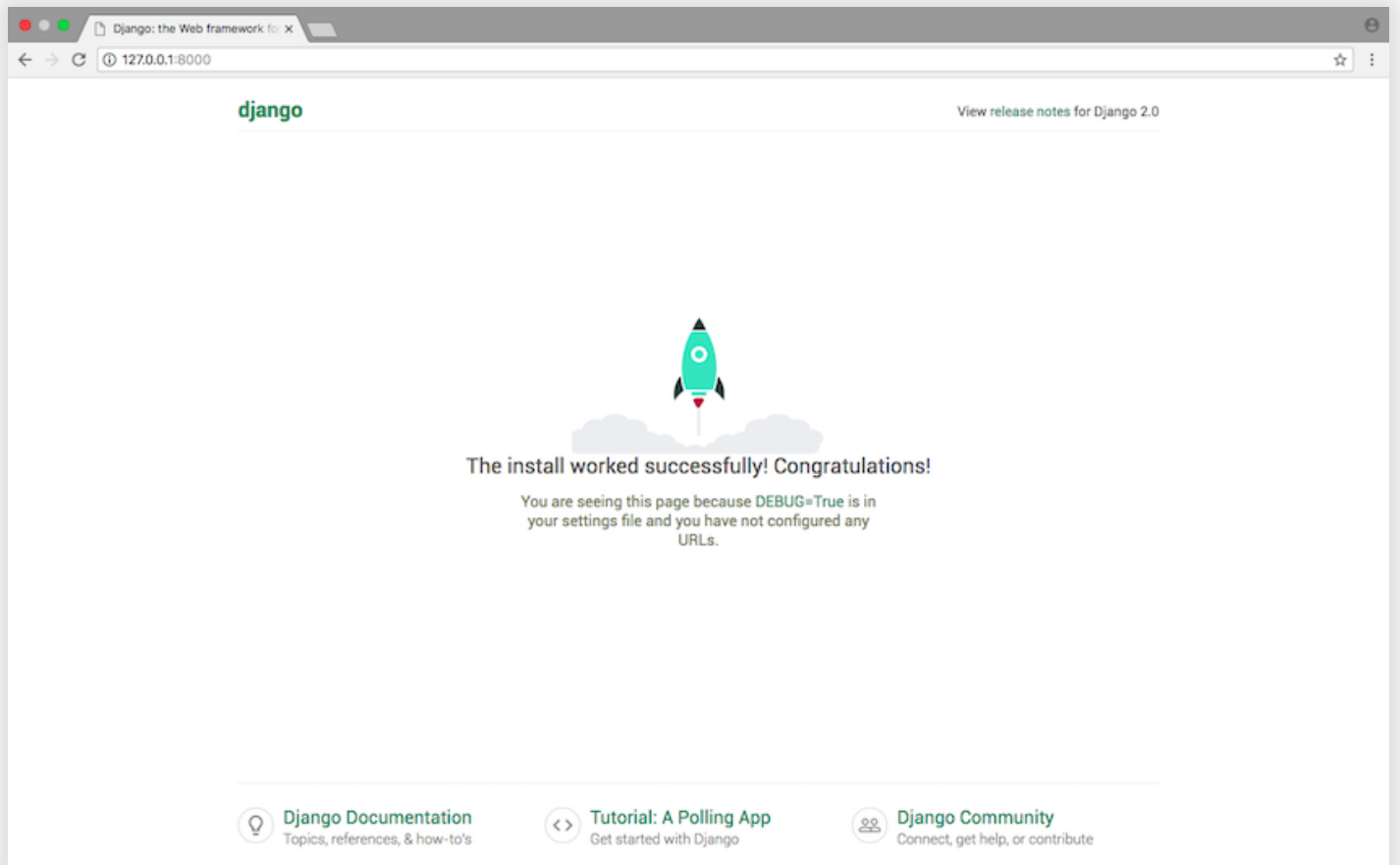
~/DjangoBros/django_blog/

```
$ ls
django_blog manage.py
$ python manage.py runserver
```

今後、`python manage.py ~` というコマンドを頻繁に使うことになりますが、このコマンドは `manage.py` ファイルの一つ上の階層にいないと使えませんので注意してください。

`python manage.py runserver` コマンドにより、コンピュータ内でそのDjangoプロジェクトのサーバーが立ち上がります。最初は、ターミナルに「マイグレートしてください」という内容のメッセージが英語で表示されると思いますが、いったん気にしなくて大丈夫です。

この状態で、`http://127.0.0.1:8000/` というURLにアクセスすると、DjangoデフォルトのWelcomeページが表示されると思います。



これで初めてDjangoのページを表示させることができました！

サーバーを停止するときは、ターミナルで `control + c` を押してください。サーバーが停止されるとページも表示されなくなります。

また、サーバーを起動した状態だと同じターミナルのウィンドウではコマンドを実行できなくなるので、新たにコマンドを打ちたい時は、ターミナルで新しいウィンドウを開いてください。新しいウィンドウでも仮想環境に入ってから作業するように注意してください。

settings.pyで設定してみよう

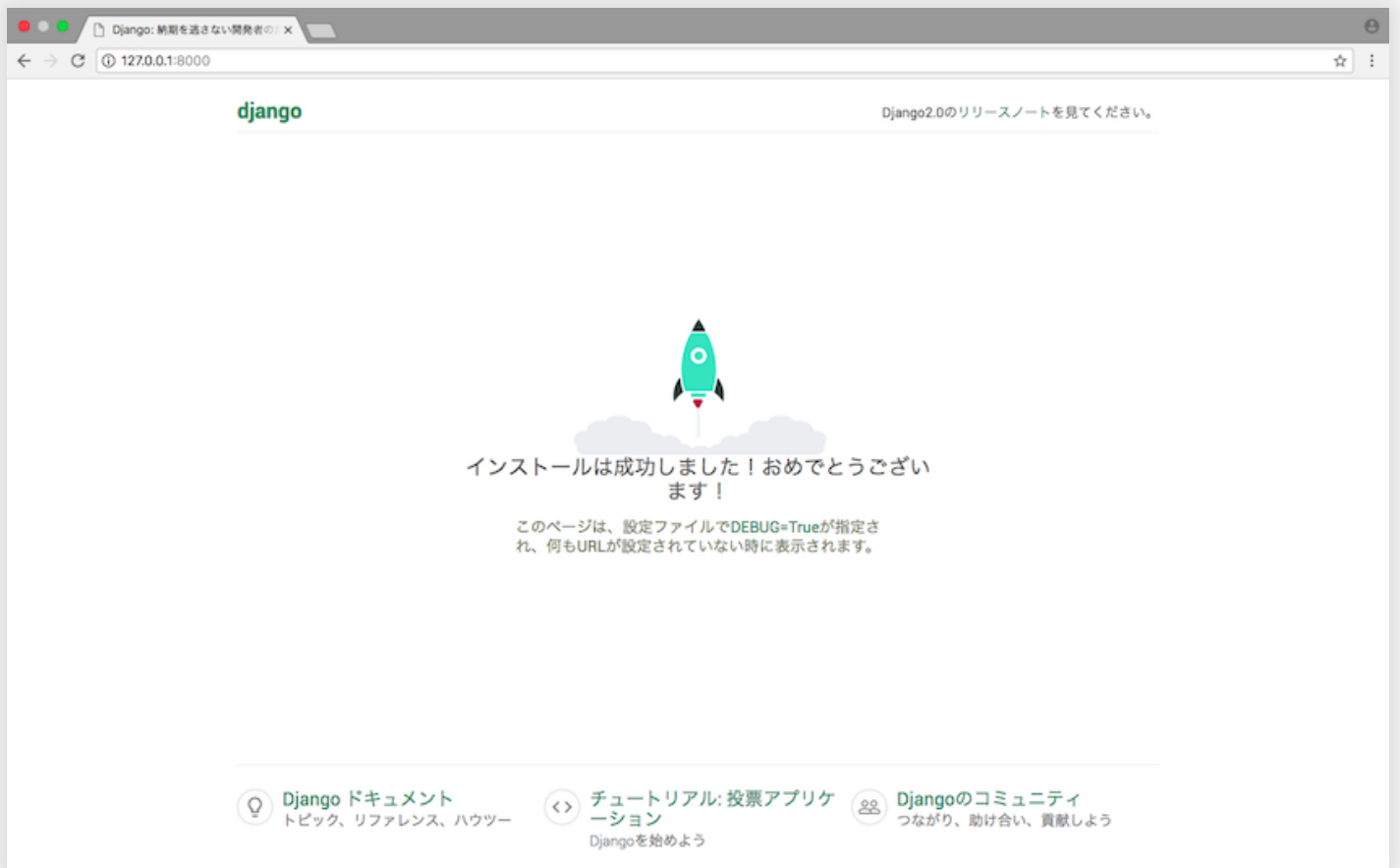
初めてのDjangoページを表示させることはできましたが、全て英語なのでちょっとわかりづらいですね。これは、使用する言語に英語が設定されているからです。開発しやすいように、使用する言語を日本語に変えてみましょう。

エディタで `settings.py` ファイルを開き、`LANGUAGE_CODE = 'en-us'` と書かれている場所を探してください。ここで使用言語を設定しています。日本語に変更するために、ここを'ja'と書き換えてください。

~/DjangoBros/django_blog/django_blog/settings.py

```
LANGUAGE_CODE = 'ja'
```

書き換えたら変更を保存して、もう一度 `http://127.0.0.1:8000/` にアクセスしてください。すると、画像のように表示が日本語になっているはずです。



続いて、`LANGUAGE_CODE` のすぐ下にある `TIME_ZONE` も書き換えてみましょう。ここでは時間に関する設定をしています。

Djangoでは、`now` という文字を書くだけで現在の日時を表示する便利な機能があるのですが、現在の時刻といってもどこのタイムゾーンを使うかで時刻が変わってきますよね。その基準を決めているのがこの `TIME_ZONE` の部分です。

時刻を日本基準にしたい場合は 'Asia/Tokyo' と設定します。

```
~/DjangoBros/django_blog/django_blog/settings.py
```

```
TIME_ZONE = 'Asia/Tokyo'
```

現在時刻の表示だけでなく、ブログの投稿日時などもこの設定が基準となるので、初めに自分の国を設定しておくとういでしょう。

ここまでで、`settings.py` ファイルが「ウェブサービス全体に関わる基本的な設定をしている」ことがなんとなくイメージできましたか？ ファイルには他にもたくさんの文字が並んでいますが、まだ理解しなくて大丈夫なので安心してください。

データベースを作ろう

ウェブサービスでは、ユーザー情報や投稿されるブログの記事など、たくさんのデータを管理しなくてはなりません。そのデータを保管するのが「データベース」です。

Djangoではデフォルトで「sqlite3」というデータベースを使うように設定されており、settings.pyファイルの中の以下の部分で設定されています。

~/DjangoBros/django_blog/django_blog/settings.py

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.sqlite3',
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),
    }
}
```

プロジェクトを立ち上げただけの現段階では、デフォルトで使うデータベースの指定がされているだけです。 `python manage.py migrate` というコマンドで実際にデータベースを作成する必要があります。ルートディレクトリに移動して、このコマンドを打ってください。

~/DjangoBros/django_blog/

```
$ python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying sessions.0001_initial... OK
```

たくさん文字が出てきますがこのような文字が表示されていれば成功で、無事データベースが作成されています。

ウェブサービスの開発を進めていく中で、データベースの設計は随時更新されますが、その変更をデータベースに反映するのが `python manage.py migrate` コマンドです。詳しくは、後ほど説明することとします。

これでこのレッスンは終了です。ついにDjangoプロジェクトを立ち上げることができましたね！続いては、ブログ管理サイトを構成するより具体的な機能の部分を実装していきます。