# Django
# ビギナーズブック

滝澤成人◉著

CUTT
カットシステム

・本書の内容についてのご意見、ご質問は、お名前、ご連絡先を明記のうえ、小社出版部宛文書（郵送またはE-mail）でお送りください。
・電話によるお問い合わせはお受けできません。
・本書の解説範囲を越える内容のご質問や、本書の内容と無関係なご質問にはお答えできません。
・匿名のフリーメールアドレスからのお問い合わせには返信しかねます。

# 前書き

本書は、Web フレームワーク Django（バージョン 2.2）の基本を習得するための書籍です。Django フレームワークとは何かを簡単に言うと、Python で凝った Web サイトを作るための便利なプログラム集です。

知り合いに Django フレームワークを薦められたり、入社した会社が Django フレームワークを使うようだけど、自分に合った Django フレームワークのチュートリアルが見つからない方、他にも、Python の基本的な文法は知っており、Web の分野にも手を出してみたい方、そういった方々に向けて、本書を執筆しました。

Python の文法や HTML、CSS の解説は極力行わず、Django フレームワークの解説にフォーカスし、仕組みの部分であったり、なぜそういう書き方をするか、などもできるだけ説明するようにしました。Django を雰囲気で使えるようになったけど、もう少し仕組み的な部分も知りたい……といった方にも、ぜひ読んでいただきたいです。

また、初学者の方がよく疑問に思う部分や、よく起こるエラーを FAQ という形式で各節の最後にまとめています。こちらも、ぜひ目を通していただければと思います。

## ■ 注意

Django フレームワーク 2.2 は、**Python 3.5 以上**で動作します。そのため、本書のサンプルコードを試すには、Python 3.5 以上をお使いの PC にインストールする必要があります。

# 目次

# 第1章

# Django と Web と HTTP レスポンス

# 1.1 概要

この章の目的は、ビューやテンプレート、URL ディスパッチャといった、Django の基本となる機能についての解説と、「**Django のビューの目的は HTTP レスポンスを作成すること**」という一文を理解できるようにすることです。例えば、関数タイプのビューでシンプルなページを表示するだけならば、次のようなコードになります。

```
def top(request):
    return render(request, 'myprofile/top.html')
```

HTML の経験がある方や、Web 系の経験がある方ならば、なんとなく、HTML ページを表示するための処理なのかなと思うかもしれません。それは概ね間違いではありません。

この 2 行のコードには、沢山の情報が詰まっています。HTTP レスポンスを表現するオブジェクトを作り、レスポンスボディ部分をテンプレートファイルと呼ばれる雛形ファイルを利用して作成し、それを返却しています。

この動作をきちんと理解しようとすると、そもそもブラウザと Web サーバーがどういう風にやり取りをしているか、Django フレームワークはどこで動いているのか、この関数がどこから呼び出されているのか、そういった部分についても触れなければなりません。

難しいと思うかもしれませんが、この仕組みを理解しておくと大変捗ります。Django の理解にも繋がりますし、他の Web フレームワークを利用する際や、Web エンジニアとしても、有意義な知識となるでしょう。それに、順を追って見ていくと案外難しいものでもありません。

この章が終われば、簡単な Web サイトぐらいは作れるようになります。実際に、シンプルなプロフィールページを作りながら進めていきます。

# 1.2 スタートプロジェクト

## 1.2.1 必要な作業の概要

まずは Django を実際に動かして、ブラウザでページが表示されるまで進めます。この章の作業は、Django での開発を始める際に毎回行うことになります。折り目をつけたり栞を挟んでおきましょう。

Django で開発をするためには、いくつかの初期設定が必要です。先に作業内容の概要を紹介します。後で見返すときなどに利用してください。

1. Django をインストールします（まだの場合）。
2. 任意のディレクトリ（フォルダーのこと）を作成し、その中でパワーシェルやコマンドプロンプト、ターミナルを開き、次のコマンドを実行します（表示されているウィンドウに対して、キーボードで次の文字列を入力します）。

```
django-admin startproject project .
python manage.py runserver
```

3. Django の設定ファイルである settings.py を開き、最後の方にある記述を次のように書き換えます。

```
LANGUAGE_CODE = 'ja'  # en-usからjaに変更

TIME_ZONE = 'Asia/Tokyo' # UTCからAsia/Tokyoに変更
```

これらの作業内容について、以降で詳しく説明します。

## 1.2.2 Django のインストール

Django は pip でインストールできます。

```
pip install django
```

書籍や Web サイトではインストールコマンドについて、上のように説明されることも多いです。しかし、これの意味するところは **pip でインストールできるよ**というだけであって、コマンドをそのまま入力しなさいという意味ではありません。

上のコマンドは環境に合わせて書き換えてください。仮想環境を利用しているのか、グローバル環境に入れるのか、Windows なのか Mac なのか、そういった環境によって変わります。「pip install django」のままで良いかもしれないし、Mac や Linux なら「python3 install django」、Windows ならば「py -m pip install django」かもしれません。Pipenv を使っていれば、「pipenv install django」になるでしょう。

## 1.2.3 Django プロジェクトの作成

無事にインストールできたら、さっそく Django プロジェクトを作成しましょう。

Django をインストールすると、django-admin というコマンドが利用できるようになります。このコマンドを使って Django プロジェクトの作成が可能です。

プロジェクトを作成する前に、どこかにディレクトリを 1 つ作成してください。私はデスクトップに **helloworld.com** というディレクトリを作りました。

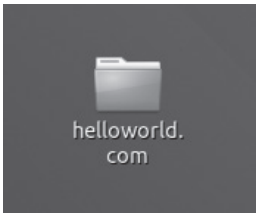



**図2.1●私が作ったディレクトリ**

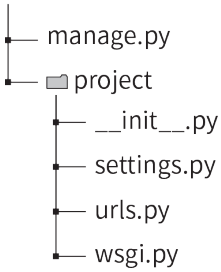
ディレクトリの中でパワーシェルやターミナルを開き、次のコマンドを実行します。最後のピリオドを忘れないようにしてください。

```
django-admin startproject project .
```

仮想環境などを利用していなくて、Windows ユーザーであり、特に環境変数パスの追加もしていないならば、次のコマンドになるでしょう。

```
py -m django startproject project .
```

上のコマンドの結果、helloworld.com ディレクトリの中身は次のようになるはずです。




Django プロジェクトとは、**Django で作る 1 つの Web サイト全体のこと**と思ってください。「django-admin startproject」コマンドは、Django で作る Web サイトに必要なファイルやディレクトリを自動生成してくれるコマンドと言えます。

各ファイルやディレクトリについて簡単に説明します。

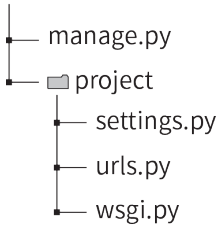
- helloworld.com ディレクトリ
  Django プロジェクトそのものです。もし、Django での開発中にエラーが出て、誰かに相談した際に「プロジェクトを見せて」と言われたら、helloworld.com ディレクトリを送付しましょう。
- manage.py
  Django プロジェクトを管理するためのファイルです。開発用サーバーの起動をはじめ、さまざまな操作を提供しています。今後頻繁に利用します。
- project ディレクトリ
  Django プロジェクトの設定に関するディレクトリです。コマンド内の project 部分と同じ名前で作成されます。本書では、**設定ディレクトリ**と今後呼びます。今回の例ならば、プロジェクト名が「helloworld.com」で、設定ディレクトリの名前が「project」ということです。
- settings.py
  Django プロジェクトの設定に関するメインファイルです。日本語や英語など、言語の設定であったり、どの国のタイムゾーンで表示するかを設定したり、その他にも多くの設定

があります。こういった設定ファイルは .ini や .json、.yaml などのファイルを使うことも多いのですが、Python では 1 つの Python モジュールをそのまま設定ファイルとして利用することも多いです。

- urls.py
  Web ページの URL と呼び出す処理を登録するファイルです。URLconf と呼ばれることもあるファイルで、この場所にある urls.py をルートの URLconf とも呼びます。本書では、単純に urls.py と呼びます。どういった動作をするかは、1.5 節「URL ディスパッチャ」で詳しく説明します。
- wsgi.py
  私たちが編集することは滅多にありません。Django の内部で利用される他、Web サイトを公開する際に意識することがある程度です。
- __init__.py
  Python パッケージに必要なファイルです。存在を意識しなくて構いません。

## FAQ

**Q.**「django-admin startproject project」（ピリオドなし）ではダメですか。
**A.** ピリオドをつけない場合、例えばデスクトップで実行したならば、デスクトップに project というディレクトリが作られ、次のような中身になっています。




少しわかりにくいかもしれませんが、project ディレクトリの中に project ディレクトリがあり、その中に settings.py などが作成されます。つまりプロジェクト名（先ほどの例ならば helloworld.com）とプロジェクトの設定ディレクトリ（先ほどの例なら project）が同じ名前で作成されるのです。

Django プロジェクトの名前は、「helloworld.com」のように実際に公開するドメイン名にすることが多いです。しかし、「django-admin startproject helloworld.com」のようにしてプ

ロジェクトを作成することはできません。コマンド内には、ピリオドやハイフンは使えないのです。

「helloworld.com」のようなプロジェクト名にしたいならば、「django-admin startproject project」（ピリオドなし）とした後にプロジェクト名だけリネームするか、今回行ったようにディレクトリ、つまりプロジェクト名だけ先に作り、そこを Django プロジェクトにするかのどちらかの方法になります。前者の方法だと、パワーシェルやターミナルでのコマンド操作が少し増えます。

プロジェクト名を自由に設定可能で、操作も多少シンプルになるため後者の方法で進めましたが、そういった事情が気にならない方はピリオドなしの単純なコマンドでプロジェクトを作成してください。この方法でなければならないというものではありません。

**Q.** 設定ディレクトリですが、「project」以外に使われる名前はありますか。
**A.** 「conf」や「config」も見かけます。startproject の標準の動作通り、プロジェクト名と設定ディレクトリ名を同一にする人も多くいます。プロジェクトを「narito.ninja」にして、設定ディレクトリの名前はピリオドやハイフンを抜いた「naritoninja」にしている例もありました。これも、この名前でなければならないというものではありません。本書では「project」という名前で進めます。

**Q.** 設定ディレクトリの名前を変更したら動かなくなりました。
**A.** プロジェクト名とは違い、設定ディレクトリの名前はいくつかのファイルの内部ですでに使われています。具体的には、settings.py に 2 か所、manage.py に 1 か所、wsgi.py に 1 か所です。それらの該当部分を新しい名前に書き換える必要があります。

### 1.2.4 開発用サーバーの起動

皆さんが普段使っている PC で Django で作ったものを確認できるように、Django には開発用サーバーが組み込まれています。次に示すのは、それを起動するためのコマンドです。

```
python manage.py runserver
```

インストールの時と同様に、Windows の方は次のようなコマンドになるかもしれません。

今後、本書で出てくるコマンドも同様に置き換えてください。

```
py manage.py runserver
```

開発用サーバーが無事に起動すると、コマンドプロンプトなどに次のように表示されます。

```
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
October 05, 2019 - 11:07:36
Django version 2.2.4, using settings 'project.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

開発用サーバーを起動しないと、Django で作ったものをすぐに確認することができません。そのため、**Django で開発する際には毎回このコマンドを実行することになります**。開発を中断したくなったり、疲れて寝たくなったら、コマンドプロンプトなどに表示されているとおり Ctrl キーを押しながら Break キーを押すことで開発用サーバーを停止することができます。ただし、環境によって停止コマンドは変わります。Ctrl キーと C キーの場合もありますし、Mac なら Ctrl キーの代わりに Command キーかもしれません。上の画面の最後に停止用のコマンドが表示されているので、それを確認しておきましょう。

ブラウザを開いて、URL 入力欄に http://127.0.0.1:8000/ と入れて、Enter キーを押しましょう。この「127.0.0.1:8000」は、自身の PC を指す少し特殊な URL です。次節であらためて説明します。

次のような画面になったら、上手く動いています。

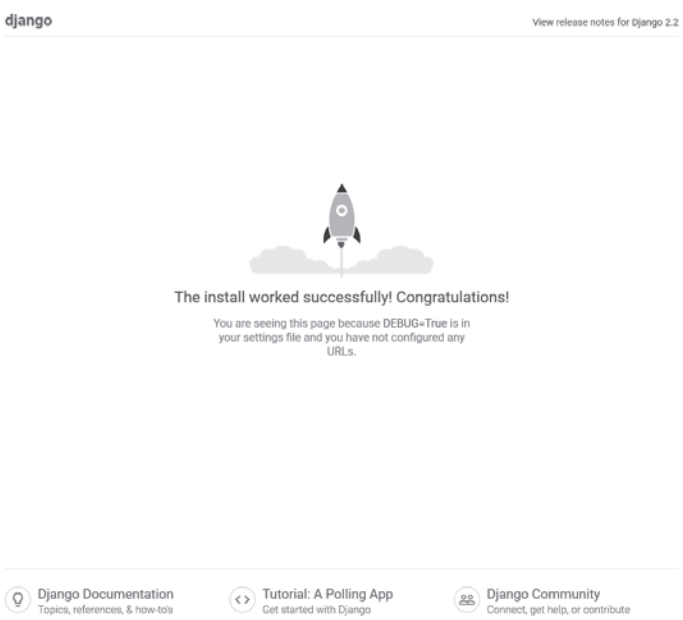



**図2.2●Djangoのようこそページ**

これは Django プロジェクトの初期ページです。まだ何も設定していない Django プロジェクトでは、このようなページが表示されます。

あと少し作業が必要です。settings.py を開き、下側（一番下から探したほうが早いです）にある LANGUAGE_CODE と TIME_ZONE を書き換えてください。

```
LANGUAGE_CODE = 'ja'  # en-usからjaに変更

TIME_ZONE = 'Asia/Tokyo'  # UTCからAsia/Tokyoに変更
```

LANGUAGE_CODE は、その Django プロジェクトをどの国の言語で表示するかを指定する変数です。ja は日本を意味します。

TIME_ZONE は、タイムゾーンの指定です。Asia/Tokyo は、日本における標準時を意味します。

これらを設定しないと、Django 内での表示時間がズレたり、さまざまなメッセージが英語で表示されてしまいます。注意してください。

Django プロジェクト内のファイルを変更すると、Django はファイルに変更があったことを自動で検知し、すべてのファイルをリロードします。settings.py を編集して保存したら、も

う一度 http://127.0.0.1:8000/ にアクセスしましょう。言語を日本語にしたので、初期ページも日本語になったはずです。


**図2.3●初期ページも日本語に！**

なお、自動リロードでは、ファイル追加のようないくつかの行動は検知されません。また、スペックによってはリロードされるまでの時間が長くなるかもしれません。エラーが出たり更新が反映されないときは、一度「manage.py runserver」で開発用サーバーを再起動してください。それでも同様の現象が起これば、それは書いたコードで何らかのエラーになっています。

## FAQ

**Q.**「UTC」とは何ですか。
**A.** 協定世界時のことで、今の世界の基準となる時間のことです。Django のデフォルト設定では、この UTC が指定されています。UTC という用語は色々な場所で見かけるので、ぜひ覚えてください。

**Q.** runserver 時に、「UnicodeDecodeError: 'utf-8' codec can't decode byte」と表示されます。
**A.** コンピューター名に日本語や記号が含まれている可能性があります。コンピューター名を半角英数字のみに変更してください。