

# Django 4

## ファーストガイド

必要最小限の準備で Django アプリ作成の基本を固める

日向俊二◎著



### ■ サンプルファイルのダウンロードについて

本書掲載のサンプルファイルは、下記 URL からダウンロードできます。

<https://----->

- 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。

本書で取り上げられているシステム名／製品名は、一般に開発各社の登録商標／商品名です。本書では、™ および ® マークは明記していません。本書に掲載されている団体／商品に対して、その商標権を侵害する意図は一切ありません。本書で紹介している URL や各サイトの内容は変更される場合があります。

# はじめに

---

かつて、ジャンゴ・ラインハルト (Django Reinhardt、1910～1953) というギタリストがいました。ジャンゴ・ラインハルトにちなんで命名された Web フレームワークが Django です。

Django は、プログラミング言語として Python を使い、データベースを主体とした Web アプリケーションを構築するためのフレームワークです。

Django と適切なライブラリやツールを使うと、見栄えが良くて機能が豊富な大規模な Web サイトを比較的短時間で構築できます。しかし、そのためには、Python と HTML や CSS、データベースの知識に加えて、さまざまな知識が必要になります。たとえば、フロントエンドライブラリ Bootstrap を使うと見栄えの良いサイトを短時間で実現することができますし、高機能な開発ツールを使えばコードの編集からサーバーでサイトを稼働させてテストすることまでをその開発ツールの機能を使って行うことができます。しかし、そうするためにはさまざまな種類の膨大な知識とそれぞれの要素に関する経験が必要になります。さらに Django には膨大なドキュメントがありますが、必ずしも初心者にとってわかりやすいものとはいえません。そのため、Django に手を出したものの途中で挫折してしまうこともあるでしょう。

本書では、Python と最も基本的な HTML と CSS、そしてデータベース SQLite3 に範囲を限定して、Django を使って Web アプリケーションを作成する方法をやさしく解説します。本書の範囲内で作る Web サイトは大規模なものでも見栄えの良いものでもありませんが、本書で Django の本質的な部分を理解してしまえば、たとえば、データベースの種類を変えることに集中するだけで大規模なアプリケーションにすることができ、Bootstrap を導入することに集中すれば見栄えの良さに重点を置いたサイトを実現することができます。Git と GitHub の知識を追加すれば多数の共同作業者と共に開発できるようになります。

本書はそうした将来の発展の基礎となる Django 開発の情報を提供することに重点を置いています。本書を活用して Django の第一歩を踏み出してください。

## 本書の表記

- > Windows のコマンドプロンプトを表します。
- \$ Linux や WSL など UNIX 系 OS のコマンドプロンプトを表します。
- >>> Python のインタラクティブシェル（インタプリタ）のプロンプト（一次プロンプトともいう）を表します。Python のインタラクティブシェルでの実行例で >>> が掲載されていても、>>> は入力しません。
- ... Python のインタラクティブシェル（インタプリタ）の行の継続（二次プロンプト）を表します。Python のインタラクティブシェルでの実行例で ... が掲載されていても、... は入力しません。
- () ひとまとまりの実行可能なコードブロックである関数であることを示します。たとえば、main という関数を表すときに、「main という名前の関数」や「関数 main()」と表記しないで、単に「main()」と表記することがあります。
- 太字** ユーザー（プログラマ）が入力する式や値、プログラムコードなどであることを表します。
- [] 書式の説明において [ と ] で囲んだものは省略可能であることを示します。



本文を補足するような説明や、知っておくとよい話題です。

## 対象とするソフトウェアのバージョン

- Django 4.0 以降
- Python 3.8 以降

本書の Django サンプルプロジェクトの実行を確認した環境

- Windows 10、Python 3.11.3、Django 4.2.0
- Linux (ubuntu 22.04.2)、Python 3.10.6、Django 4.2.0

## ご注意

- 本書の内容は本書執筆時の状態で記述しています。Python や Django などのバージョンによっては本書の記述と実際とが異なる可能性があります。
- 本書は Python や Django のすべてのことについて完全に解説するものではありません。必要に応じて他のリソースを参照してください。
- 本書に掲載のコード断片や特定のファイルをそのまま実行することはできません。Django では複数のファイルやコードが有機的に結合して機能します。コードの意味を理解してから実行したり編集したりサイトとして表示したりしてください。
- 本書のサンプルは、プログラミングやコーディングを理解するために掲載するものです。実用的なプログラムとして提供するものではありませんので、ユーザーのエラーへの対処やセキュリティ、その他の面で省略してあるところがあります。そのままのかたちで本番環境では使わないでください。

## 本書に関するお問い合わせについて

本書に関するお問い合わせは、sales@cutt.co.jp にメールでご連絡ください。

なお、お問い合わせは本書に記述されている範囲に限らせていただきます。特定の環境や特定の目的に対するお問い合わせ等にはお答えできませんので、あらかじめご了承ください。特に、特定の環境における特定の開発ツールのインストールや設定、使い方、読者固有の環境におけるエラーなどについてご質問いただいてもお答えできませんのでご了承ください。

お問い合わせの際には下記事項を明記してくださいますようお願いいたします。

- 氏名
- 連絡先メールアドレス
- 書名
- 記載ページ
- お問い合わせ内容
- 実行環境



# 目次

はじめに .....	iii
------------	-----

## 第 1 章 Django と Python ..... 1

1.1 Django の概要 .....	2
◆ フレームワーク .....2	
◆ サーバーとクライアント .....2	
◆ Django の特徴 .....3	
1.2 インストール.....	5
◆ Python のインストール .....5	
◆ pip のインストール .....6	
◆ Django のインストール .....7	
1.3 Python との対話 .....	8
◆ プロンプト .....9	
◆ 単純な加算 .....10	
◆ print() を使った出力 .....11	
1.4 スクリプトファイル.....	13
◆ ファイルの作成と保存 .....13	
◆ スクリプトの実行 .....15	

## 第 2 章 はじめての Django ..... 17

2.1 プロジェクトの作成.....	18
◆ 初期プロジェクトの生成 .....18	
◆ 初期プロジェクトの内容 .....19	
◆ サイトの表示 .....21	
◆ ポートの変更 .....23	
2.2 関数版 Hello .....	24
◆ 作成するアプリケーション .....24	
◆ settings.py の編集 .....24	
◆ URL の追加 .....25	
◆ ビューの追加 .....26	
2.3 クラス版 Hello .....	28
◆ 作成するページ .....28	
◆ URL の追加 .....28	
◆ ビューの追加 .....29	
◆ 設定の変更 .....30	
◆ テンプレートファイルの作成 .....31	
2.4 インデックスページの作成.....	32
◆ URL の追加 .....32	
◆ ビューの追加 .....33	
◆ テンプレートファイルの作成 .....33	
2.5 Django サイトの仕組み .....	34
◆ ファイルの役割 .....34	
◆ manage.py .....36	

---

---

## 第 3 章 データモデル .....39

3.1	データベース.....	40
◆	データベース	40
◆	データベース管理システム	40
◆	データベースの構造	41
◆	リレーショナルデータベース	43
◆	データベースの完全性	44
◆	データベースの種類	45
◆	モデルクラス	46
3.2	アプリケーションの作成.....	46
◆	プロジェクトの作成	46
◆	アプリケーションの作成	48
◆	ビューの作成	49
◆	URL の設定	50
◆	アプリケーションの設定	51
3.3	データモデルの設定.....	52
◆	作成するデータベース	52
◆	データベースの設定と作成	53
◆	モデルの作成	53
◆	テーブルの作成	56
◆	作成されたデータベース	58

## 第 4 章 管理画面 .....61

4.1	Django Admin.....	62
◆	管理ユーザーの作成	62
◆	テーブルの追加	63
◆	ログイン	63
◆	データの登録	65
4.2	シェルからの操作.....	68
◆	シェルの起動	68
◆	データの操作	68
◆	Django とデータベース	71

## 第 5 章 ビュー .....73

5.1	サイトのホームページ.....	74
◆	作成するビュー	74
◆	テンプレートの設定	75
◆	URL の設定	76
◆	ビューの作成	76
◆	サイトの index.html	77
5.2	アプリケーションのインデックスページ.....	79
◆	作成するビュー	79
◆	URL の設定	79
◆	ビューの作成	80
◆	用語一覧の index.html	81
5.3	用語ページ.....	82
◆	作成するビュー	82
◆	URL の設定	83
◆	ビューの作成	84
◆	用語のテンプレート	84



## 第 6 章 グルメサイトの作成 (1) .....87

6.1	グルメサイトの生成.....	88
	◆ サイトの概要 .....88	
	◆ プロジェクトの作成 .....90	
	◆ dish アプリケーションの作成 .....91	
	◆ accounts アプリケーションの作成 .....93	
	◆ URL の設定 .....94	
6.2	モデル.....	96
	◆ アカウト用のモデル作成 .....96	
	◆ 料理モデルの作成 .....98	
6.3	データの管理.....	100
	◆ 管理画面へのモデルの追加 .....100	
	◆ 管理ユーザーの作成 .....101	
	◆ データの確認 .....104	

## 第 7 章 グルメサイトの作成 (2) .....107

7.1	テンプレートの構造.....	108
	◆ ベーステンプレートとコンテンツ .....108	
7.2	ビューの作成.....	111
	◆ 料理のビュー .....111	
	◆ 料理のテンプレート .....116	
	◆ アカウトのビュー .....122	
	◆ アカウトのテンプレート .....123	

## 付 録 .....127

付録 A	メモランダム.....	128
付録 B	トラブルシューティング.....	136
付録 C	参考リソース.....	144
索引	.....	145



# 第 1 章

## Django と Python

この章では、Django の概要、使いはじめるために必要なインストールやその他の準備、Python の基本的な扱い方について解説します。

## 1.1 Django の概要

Django（ジャンゴ）は Python で Web アプリケーションを開発する際に使う Web フレームワークです。

### ◆ フレームワーク .....

Django は Web アプリケーションのためのフレームワーク（Frame work、枠組み、骨組み）です。

アプリケーションの目的や機能はアプリケーションごとに異なりますが、その本質的な構造は似通っています。たとえば、多くの Web アプリケーションは、投稿や商品の一覧やリストがあって、閲覧者はそこから詳細を表示してあるページに飛び、新規投稿や注文のフォームで情報をサーバーに送れるようになっています。また、たとえば、サイトにログインすると特定の操作ができるような仕組みは、アプリケーションの種類にかかわらずほぼ同じです。そうした大きな骨組みや共通する機能を容易に利用できるようにしたものがフレームワークです。

Django はさまざまな種類の Web アプリケーションを作成するための骨組みとなり、また詳細を作成するための共通した部品となるものやツールを提供します。

現在では要求が多様で高度になったため、ある程度の規模の Web アプリケーションをゼロからすべて自分で作ることはほとんどなく、たいていの Web アプリケーションがなんらかのフレームワークを使って作成されています。Django はそうしたフレームワークの中でも、最も便利で拡張性や柔軟性が高く、幅広く使われているもののひとつです。

### ◆ サーバーとクライアント .....

Web サーバーはクライアントからのリクエストによって Web ページやその他の情報をクライアントに送り返します。そのための適切な情報を作り出すのが Web アプリケーションの役割です。典型的なクライアントは Web ブラウザですが、それ以外のアプリケーションである場合もあります。

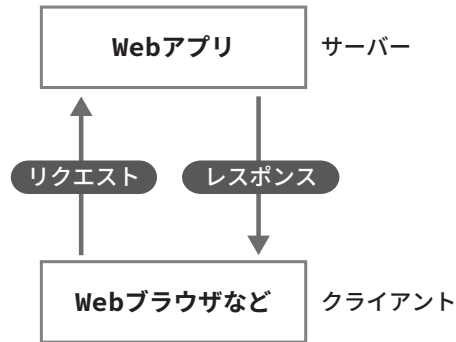


図1.1 ●サーバーとクライアント

Django で構築した Web アプリケーションも、サーバー上で稼働し、クライアントからのリクエストによって Web ページなどをクライアントに送り返します。

## ◆ Django の特徴 .....

Django には次のような特徴があります。

- Django は、実績があり安定した信頼できるブラックボックスです。フレームワークを利用するときには、背後で行われている詳細はそのフレームワークの開発者に任せ、フレームワークを信頼して、何をするとどうなるのかという点を理解して利用することになります。Django はそういった観点から信頼して利用できるツールです。
- Web アプリケーションの開発に使われる、「ユーザー認証」、「管理画面」、「サイトの案内ページ」、「RSS フィード」など、多くのサイトで必要になる要素がほぼすべて揃っています。  
このように、必要なあらゆるものが揃っているものをフルスタックともいいます。
- さまざまな目的に対応する非常に豊富なライブラリを持っているプログラミング言語 Python を使っています。たとえば、イメージ（画像）を含む多様な情報を扱ったり AI を実装するためのたくさんのライブラリが Python では利用できます。

- データベースを容易に利用することができます。Django では SQLite はデフォルトでインストールされます。また、PostgreSQL、MariaDB、MySQL、Oracle などをサポートします。
- ユーザー認証システムとして、ユーザーのアカウントとパスワードを安全に管理する方法が提供されています。Web アプリケーション開発者はログインと認証のための詳細を作成しないで、Django が提供する機能をほぼそのまま利用することができます。
- サイト全体の管理画面があります。データモデルを定義するだけで Web アプリケーション管理者が必要最低限のデータ管理を行うことができるようになります。また、サイトの案内ページを容易に作れます。
- 高度なセキュリティを実現できます。ユーザー認証システムをはじめ、さまざまな脅威に対する対策が施されているのでセキュリティ面で安心です。
- Django の各ファイルはそれぞれ役割が決まっています。そのため、目的に応じて独立して扱ったり拡張することが容易です。ほとんどどのような規模の Web アプリケーションにも柔軟に対応できます。そのため、用途が非常に広いといえます。



.....

Django の機能は強力で守備範囲も広範ですが、本書でそのすべてを取り上げるわけではありません。本書は Django で Web アプリケーションの開発をスタートすることに重点を置いています。

.....

## 1.2 インストール

Django を使うためには、システムに Python と Django および関連ツールなどがインストールされていることが必要です。

システムによっては必要なものがあらかじめインストールされている場合もあるので、インストールする前にそれがインストールされているかどうか調べ、インストールされているならバージョンを調べる必要があります。

### ◆ Python のインストール ◆

Django ではプログラミング言語としてだけでなく実行環境として Python を使います。

最初に Python をインストールする必要がありますが、システムにすでに Python がインストールされている可能性があります。オプション `--version` を付けて Python を実行してみると Python がインストールされているかどうかと、Python がインストールされている場合はバージョンを確認できます。

```
>python --version  
Python 3.11.1
```

Python を起動するコマンド名は、環境によっては `py` や `python3`、あるいは `python3.11` などの場合があります（数字はバージョンによって変わります）。その場合は、「`py --version`」や「`python3 --version`」のように有効なファイル名を指定して実行してください。

Django 4.x は Python3.8 以降でサポートされますので、Python がインストールされていない場合、および、Python のバージョンが 3.8 より古い場合は、新しいバージョンの Python をインストールしてください。

Python をインストールするには、Python の Web サイト (<https://www.python.org/>) の [Download] からプラットフォームとバージョンを選択してインストールします。選択したプラットフォーム / バージョンにインストーラーやインストールパッケージが用意されている場合は、それをダウンロードしてインストールする方法が最も容易なインス

ツール方法です。

Windows の場合、Microsoft Store からインストールすることもできます。

Linux や macOS の場合は、ディストリビューションに Python のパッケージが含まれている場合が多く、特に Python をインストールしなくても Python を使える場合が多いでしょう。ただし、インストールされているのが Python 3.8 より前のバージョンである場合は、新しいバージョンの Python をインストールする必要があります。

## ◆ pip のインストール ..... ◆

Python と Django そのほかさまざまなものをインストールしたり更新するために、最初に python 公式のパッケージ管理システムである pip がインストールされているか調べます。

次のコマンドで pip がインストールされているかわかります。

```
>pip --version
```

pip がインストールされていない場合は、以下の方法でインストールします。  
Windows では、次のコマンドで pip をインストールします。

```
>python get-pip.py
```

macOS では、次のコマンドで pip をインストールします。

```
$ sudo easy_install pip
```

apt をサポートする Linux (Ubuntu など) では、次のコマンドで pip をインストールできます。

```
$ sudo apt install python3-pip
```



pip がインストールされていても、pip のバージョンが古い可能性があるので、次のコマンドを実行してアップグレードします。

```
>python -m pip install --upgrade pip
```

または

```
>pip install --upgrade pip
```

## ◆ Django のインストール ..... ◆

システムによってはあらかじめ Django がインストールされている場合もあるので、Django がインストールされているかどうか、次のコマンドラインを実行してみます。

```
>python -m django --version
```

これで「No module named django」というエラーが出たら Django はインストールされていないでしょう。Django がインストールされていれば、インストールされている Django のバージョンが表示されます。

Django をインストールするときには、実際に使用する Python のバージョンに応じてインストールできる次のコマンドを使うことを推奨します。

```
>python -m pip install django
```

インストールが進行し、最終的に「Successfully installed」と表示されればインストールは成功しています。

## Note

Django でデータベースに SQLite3 を使う場合はデータベースに関して特に何もしなくてもかまいません。しかし、SQLite3 以外の「巨大な」データベースを利用する場合は、PostgreSQL、MariaDB、MySQL、Oracle などのいずれかのインストールと、データベースの設定が必要になります。その場合は、それぞれのデータベースの情報を取得して設定してください。初心者は最初は SQLite3 を使うことを推奨します。

## 1.3 Python との対話

Python のプログラムの主な実行方法には、2 種類あります。ひとつは、Python のインタラクティブシェル（対話型インタープリタ）を使って実行する方法です。もうひとつは、Python のプログラムファイル（スクリプトファイル）を作成して実行する方法です。ここでは簡単なプログラムの実行のしかたを学びます。

最初に、インタラクティブな方法（対話的方法）で Python を使い始めるために必要なことを説明します。

Python のインタラクティブシェルを起動して、Python を起動して確かめてみましょう。

Python が起動すると、Python のメッセージと一次プロンプトと呼ばれる「>>>」が表示されます。これが Python のインタラクティブシェルのプロンプトです。

```
>python
Python 3.11.1 (tags/v3.11.1:a7a450f, Dec 6 2022, 19:58:39) [MSC v.1934 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

これは Windows で Python 3.11.1 場合の例です。表示されるバージョン番号やそのあとの情報（Python をコンパイルしたコンパイラやプラットフォームの名前など）は、こ

の例と違っていても構いません。

Linux なら、たとえば次のように表示されることがあります。

```
$ python3
Python 3.10.6 (main, Mar 10 2023, 10:55:28) [GCC 11.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

いずれにしても、「Type "help", "copyright", "credits" or "license" for more information.」を含む Python のメッセージと Python のインタラクティブシェルのプロンプト「>>>」が表示されれば、インタラクティブシェルが起動したことがわかります。

インタプリタは「解釈して実行するもの」という意味、インタラクティブシェルは「対話型でユーザーからの入力を受け付けて結果や情報を表示するもの」という意味があります。

## ◆ プロンプト .....

Python のインタラクティブシェルのプロンプト「>>>」が表示されている環境では、入力された Python の命令や式などを Python のインタプリタが 1 行ずつ読み込んで、その結果を必要に応じて出力します。言い換えると、インタラクティブシェルのプロンプト「>>>」に対するユーザー（Python のユーザーはプログラムを実行する人）からの命令や計算式の入力を受け付けます。このようにプロンプトに対して命令や計算式などを入力することで、後で説明するようなさまざまなことを行うことができます。



Python を使っているときには、OS（コマンドウィンドウ、ターミナルウィンドウなど）のプロンプトである「>」や「#」、「\$」などと、Python のインタラクティブシェルを起動すると表示されるインタラクティブシェルのプロンプト「>>>」を使います。この 2 種類のプロンプトは役割が異なるので区別してください。なお、データベースのシェルを使う場合は、上記 2 種類とは異なるデータベースのシェルのプロンプトに対して入力することになります。つまり、3 種類以上のプロンプトを使うことがあります。

## ◆ 単純な加算 .....

Python のインタラクティブシェルに慣れるために、最初に Python で計算をしてみましょう。

Python のプロンプト「>>>」に対して  $2+3$  を入力し、Enter キーを押してみます。

```
>>> 2+3
5
>>>
```

上に示したように、 $2+3$  の結果である 5 が表示されたあとで、新しいプロンプトが表示されるはずですが（以降の例では、結果の後に表示される >>> は省略します）。



Web ブラウザや IDE のようなツールを使ってプログラムを実行するときには、プログラムコードを入力するための入力フィールドにコードを入力して、プログラムを実行するためのメニューコマンドやボタンをクリックします。なお、Web ブラウザや IDE を使う実行環境で実行するときには、`print(2+3)` のように `print()` を使わないと結果が出力されない場合があります。

引き算や掛け算、割り算を行うこともできます。引き算の記号は「-」（マイナス）ですが、掛け算の記号は数学と違って「\*」（アスタリスク）、割り算の記号は「/」（スラッシュ）です。

たとえば、 $6 \times 7 - 5$  を実行すると次のようになります。

```
>>> 6*7-5
37
```

もっと複雑な式も、もちろん計算できます。次の例は、 $123.45 \times (2+7.5) - 12.5 \div 3$  の計算例です。