

# アプリケーションで機能を作ろう

1つ前のレッスンではプロジェクトを作りました。続いて、アプリケーションを作成していきましょう。

プロジェクトが「ウェブサービス全体に関わる設定」をしているのに対して、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部分」のことでしたね。

では、今回作るブログ管理サイトではどのような機能が必要でしょうか。

- 管理ページから記事を投稿できる
- トップページに、記事の一覧を投稿日時の降順で表示する
- トップページには、各記事の冒頭 100 文字のサマリーが表示される
- 記事詳細ページで記事全文を読める

これらの機能を実装してみたいと思います。非常にシンプルなサイトですが、まずはDjangoでサイトを作る工程を一通り体験してみましょう！

さっそく、アプリケーションを作成します。アプリケーションを作るときは `python manage.py startapp` コマンドを使います。

プロジェクトルートディレクトリで以下のコマンドを実行してください。

```
~/DjangoBros/django_blog/
```

```
$ python manage.py startapp blogs
```

このコマンドにより、ルートディレクトリ内に `blogs` というディレクトリができています。これがDjangoにおける一つのアプリケーションです。このblogsディレクトリの中には、すでにたくさんのファイルがありますが、これらのファイルにコードを書いていくことで、様々な機能を実現していきます。

## アプリケーションを登録する

`python manage.py startapp` コマンドでアプリケーションを作ることができましたが、アプリケーションを新しく作ったら最初にやらなければならないことがあります。それは、作成したアプリケーションをDjangoプロジェクトに登録してあげることです。

`django_blog/django_blog/settings.py` ファイルを開いて、blogsアプリケーションを取り込む必要があります。

`INSTALLED_APPS` の部分に、`'blogs'` を追加してください。

```
django_blog/django_blog/settings.py
```

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'blogs',  
]
```

こうすることによって、`blogs` アプリケーションが `django_blog` プロジェクトのアプリケーションとして扱われるようになります。これをしないと、この後アプリケーションをいくら編集してもプロジェクトに反映されませんので、アプリケーションを作ったら、`INSTALLED_APPS` に追加することを忘れないようにしてください。

## Templatesでページを作ろう

アプリケーションを登録することができたので、まず最初にシンプルなトップページをつくらせて表示してみましょう。ユーザーが最初に訪れるページのことです。

ページを表示するときは、通常のホームページ等と同様にHTMLファイルを作成してそれを表示させるのですが、Djangoで扱うHTMLファイルは、動的なコードを扱うことができます。つまり、HTMLファイル内でPythonコードを書くことができ、変数やfor文などを扱うことができます。とっても便利ですよ！ Djangoでは、これらHTMLファイルを `Templates` と呼びます。

では、トップページ用に最初のテンプレートを作成しましょう。blogsディレクトリに移動して以下のようにコマンドを打ってください。

※テンプレートを配置

```
~/django_blog/blogs/
```

```
$ mkdir templates  
$ cd templates  
$ mkdir blogs
```

上記のコマンドでは3つのことをしています。

- django\_blog/blogs/内に「templates」というディレクトリを作成
- 今作成した「templates」ディレクトリに移動
- django\_blog/blogs/templates/内にblogs/というディレクトリを作成

この一連の作業で、`django_blog/blogs/templates/blogs/` というディレクトリ階層ができました。

Djangoでは、このようにアプリケーション内に作成した `templates/<アプリケーション名>` ディレクトリ内にテンプレート（HTMLファイル）を配置していきます。

アプリケーションの中に、わざわざアプリケーションと同名のファイルを作って、そこにHTMLファイルを作成しています。煩雑でもっとシンプルにできるように思うかもしれませんが、何でこんなことをしなくてはいけないかについての説明は、別のチュートリアルで詳しく説明します。

次に、`django_blog/blogs/templates/blogs/`内に、`index.html`というファイルを作成してください。

```
~/django_blog/blogs/templates/blogs/
```

```
$ touch index.html
```

htmlファイルを作成できたらエディタでそのファイルを開いて少しだけHTML文書を書いてみましょう。文字が書けたら忘れずに保存してくださいね。

```
index.html
```

```
<h1>ブログサイト</h1>
<p>ここはトップページです。</p>
```

これでトップページに表示させるHTMLファイルを作ることができました！

しかし、ファイルを作成しただけでは、どのURLのときにこのファイルを表示していいのかをDjangoが把握できていません。ですので、今度はこのテンプレートファイルとURLを結びつけていきましょう。

## Viewとは？

トップページを表示させるためにまずは`View`というものを作成していきます。

DjangoにおけるViewでは、ユーザーから送られてきたリクエストをもとに、どのHTMLファイルを表示させるかや、こういった内容のデータを表示させるかなどを決める処理をしています。

ユーザーは、特定のURLにアクセスするとサーバーに対してリクエストを送っています。そのリクエスト情報をもとに、表示内容を決めているのです。

言葉で説明すると難しく感じますが、慣れるとそれほど難しいことはありません。むしろ、非常にシンプルかつ綺麗にプロジェクトの設計ができることがわかってくると思います。

まずは簡単なViewを書いて感覚を掴んでいきましょう。blogsアプリケーションの中にある`views.py`ファイルを開いてください。

ファイルを開いたら、このようにコードを書いてください。これはPythonファイルなので、インデントも重要です。インデントもちゃんとこの通りに書いてくださいね。

~/django\_blog/blogs/views.py

```
from django.shortcuts import render

def index(request):
    return render(request, 'blogs/index.html')
```

このコードが何をしているのか、何となくイメージができますか？

ここでは `def index()` という関数を作っていますよね。そしてこの関数は `request` を引数にとっています。requestとは、ユーザーがURLを入力してサーバーにアクセスする時に送られる情報のことです。requestには、例えばログインしている人のユーザー情報などの様々な情報が含まれています。

この関数は、ユーザーからの `request` 情報を引数にとって、`render` を `return` しています。`render` メソッドは、request情報を元にしてindex.htmlを表示することを意味しています。renderは[与える・提供する]という意味があります。

つまり、一言でいうとこのindex関数は「ユーザーからのrequest情報を元に、index.htmlを返す」関数となります。

## URLの設定

今度はindex関数を特定のURLと結びつけて、ユーザーがそのURLを打ち込んだ時にindex.htmlファイルがブラウザ上で表示されるようにしてみましょう。URLは、`urls.py` ファイルで設定します。

まず、`blogs` アプリケーション自体を、プロジェクト全体のURLに紐付けます。

`django_blog/django_blog/urls.py` を開いて、以下のように書き換えてください。

※プロジェクト全体のURL設定

django\_blog/django\_blog/urls.py

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('blogs.urls')),
]
```

2行目では、`include` と追記しています。これでinclude関数が使えるようになります。

`path('', include('blogs.urls'))` の部分で何をしているかというと、  
`http://127.0.0.1:8000/` にアクセスされた時は、`blogs.urls` ファイルを参照することを意味しています。。`blogs.urls` ファイルとは、この後自分で作成する`urls.py`ファイルのことです。

例えばここを、`path('blogs/', include('blogs.urls'))` と書いたとすると、  
`http://127.0.0.1:8000/blogs` にアクセスされたときに`blogs.urls` ファイルを参照するようになります。

続いて、blogsアプリケーション内のURLの設定を行います。blogsディレクトリに移動してtouchコマンドで`urls.py`というファイルを作ってください。

ファイルを作ったら、以下のようにコードを書いてください。

django\_blog/blogs/urls.py

```
from django.urls import path
from . import views

app_name = 'blogs'
urlpatterns = [
    path('', views.index, name='index'),
]
```

1行目では、DjangoのURL機能である`path`関数をインポートしています。

2行目の`from . import views`という部分では、同じ階層にある`views.py`ファイルをインポートしています。`ドット`は、同じ階層という意味です。

`path`関数では、第一引数で空の文字列を指定し、第二引数で、`views.index`を指定することで、URL (`http://127.0.0.1:8000/`) にアクセスした時は、`views.py`の`index`関数を実行するように設定をしています。

例えば、ここを`path('top/', views.index, name='index')`のように書き換えたとすると、URL (`http://127.0.0.1:8000/top`) にアクセスしたときに`index`関数が実行されることになります。このように、第一引数ではURLのパスを設定しています。

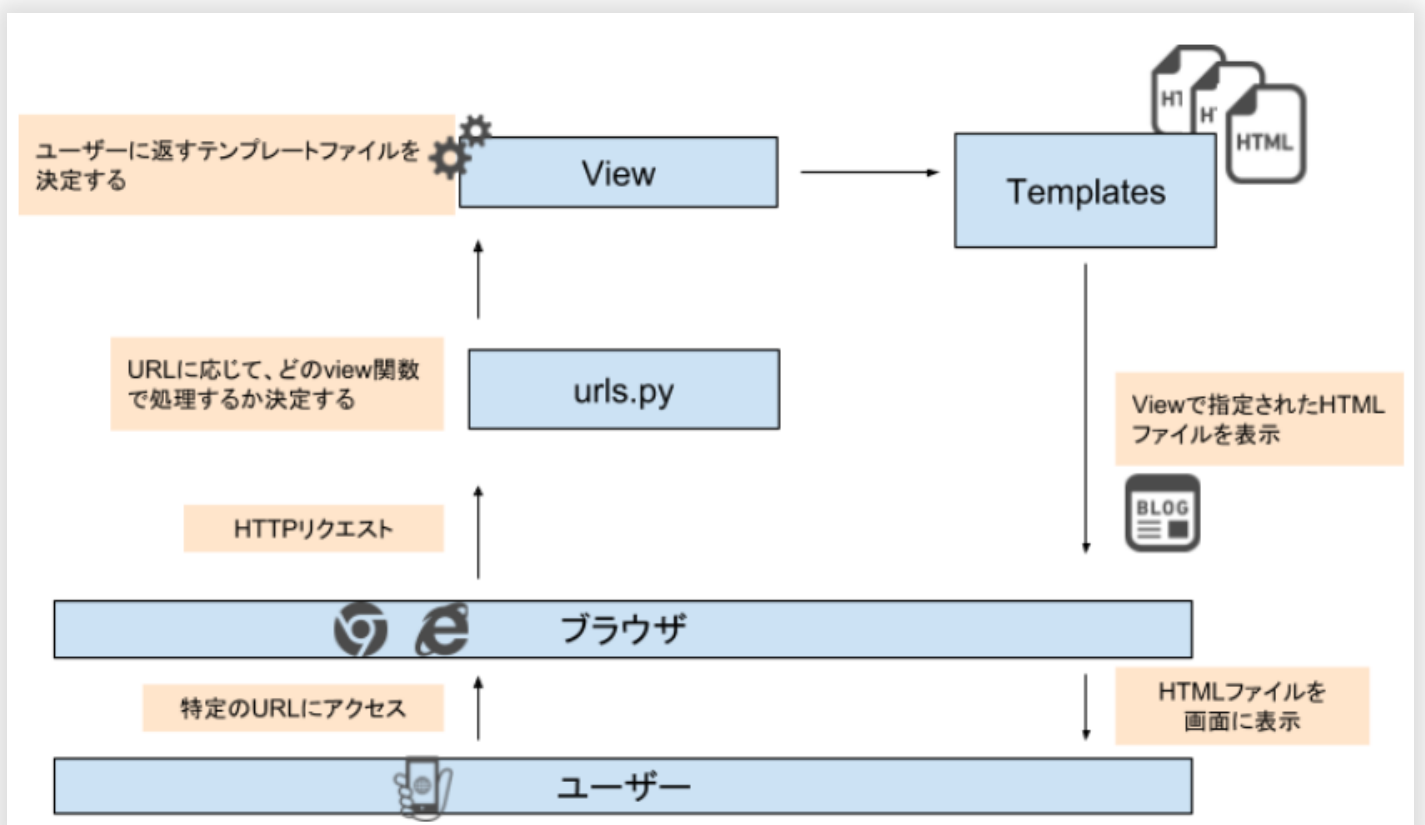
第三引数の`name='index'`という部分は、このURLパスに名前をつけてあげています。

これでページを表示する準備が整いました。ローカルサーバーを立ち上げて、  
`http://127.0.0.1:8000/` にアクセスすると、以下のようなページが表示されるはずです。



トップページをうまく表示することができましたね。

ここまで学習した流れを図で見るとこのようになります。



`URL→View→Template`の流れはDjangoアプリケーションにおいてベースとなるものですので、この流れをしっかりと抑えておきましょう。

少しずつウェブサイトっぽくなってきました。さあ、次のレッスンでは実際にブログを書いて投稿できるようにしていきます。