

# Django

Djangoの使い方

2024/02/23

# Django

## Djangoとは

- PythonでWebアプリを制作できる**Webフレームワーク**
- デフォルトでの**セキュリティ面が優れている**
- **InstagramやYouTubeにも使われているフレームワーク**



# Django

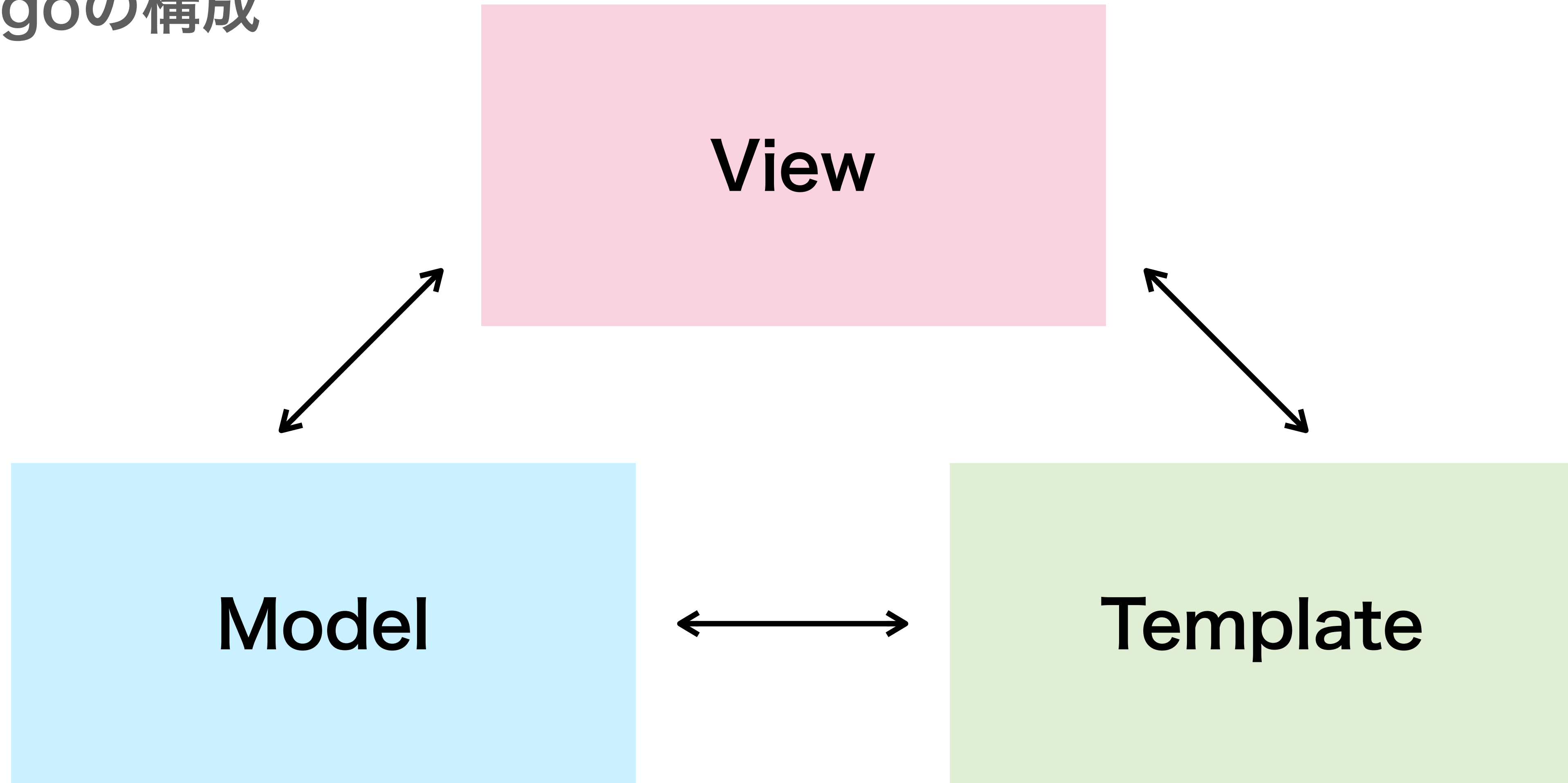
## Djangoとは

- ・ SSRまで可能な  
フルスタックフレームワーク



# Composition

Djangoの構成



# View

ビュー

- Djangoの**制御パート**
- **URLごとに実行する関数**を変更
- `views.py`という名称で存在する

# Model

## モデル

- Djangoのデータ保管パート
- クラスごとにテーブルを管理
- models.pyという名称で存在する

# Template

## テンプレート

- Djangoのページ表示パート
- **HTMLなどページのデザインを保存する**
- templatesというフォルダ内に存在する

# Virtual Environment

## 仮想環境

- ・ パソコン上全てのプロジェクトで同じ環境を使用していると、  
不要なライブラリを導入していたり、依存関係でエラーが生じることがある
- ・ 仮想環境を作成することでプロジェクトごとに環境を変えられる
- ・ 慣例上、仮想環境名は "venv" とすることが多い

A terminal window with a dark background and a title bar labeled "Terminal". The title bar has three colored window control buttons (red, yellow, green) on the left. The terminal displays the command `python -m venv [仮想環境名]` in a light-colored monospace font.

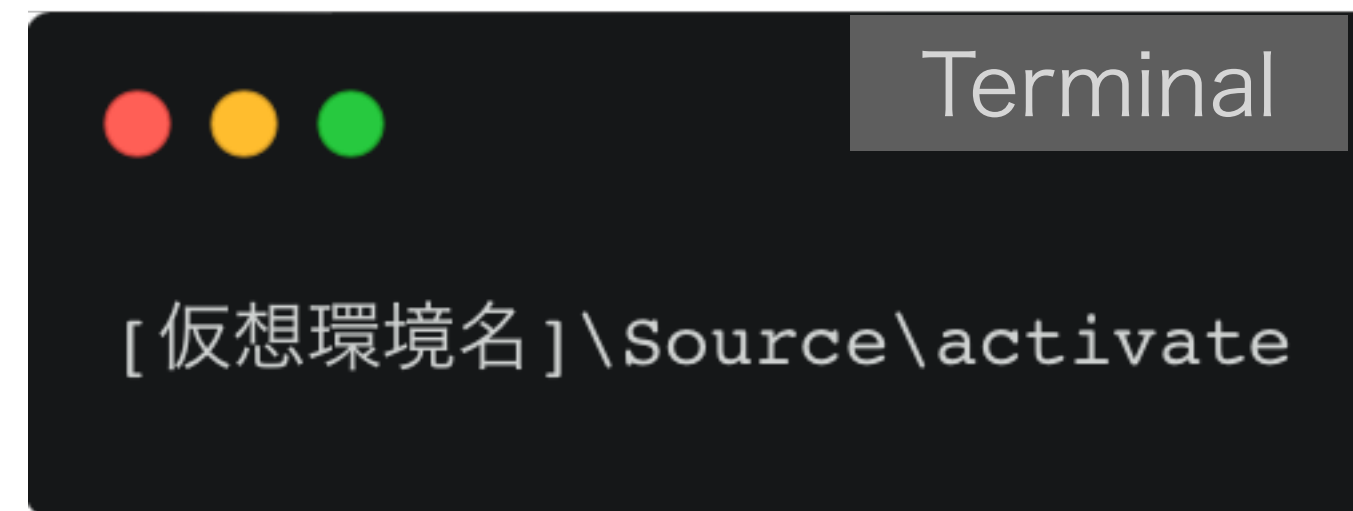
```
python -m venv [仮想環境名]
```



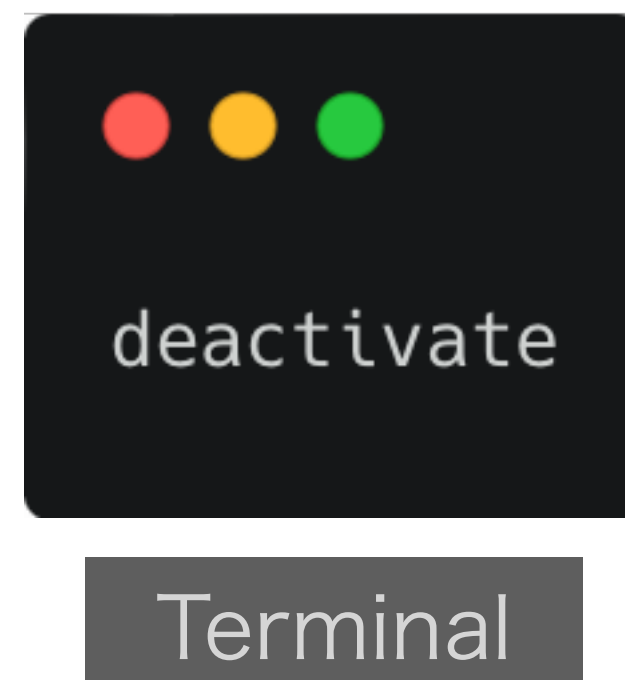
# Virtual Environment

## 仮想環境 (Windows)

- ・ 仮想環境の有効化



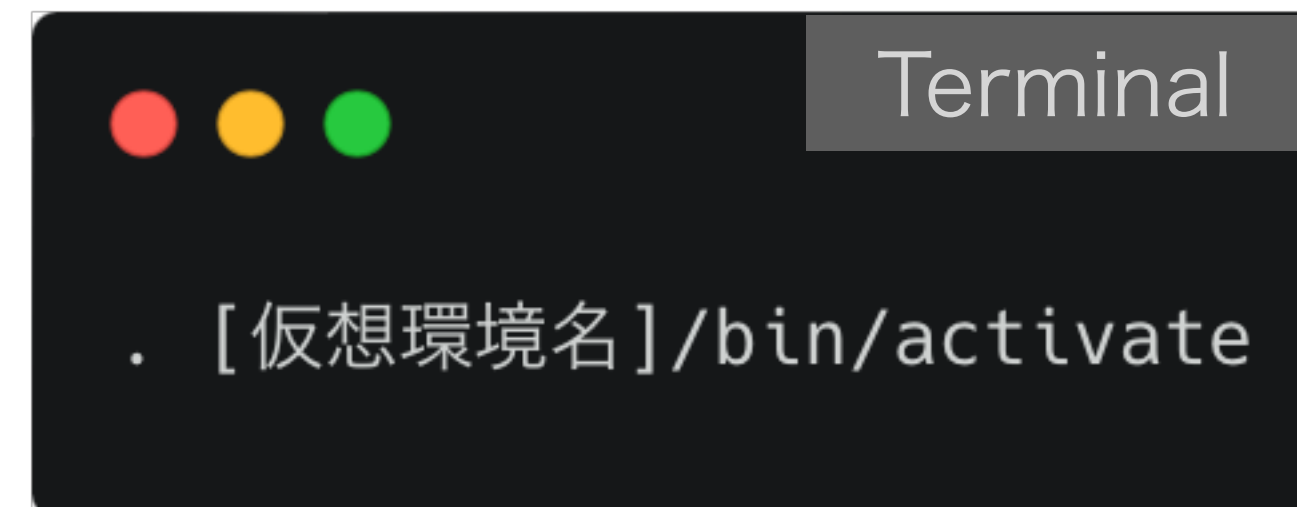
- ・ 仮想環境の無効化



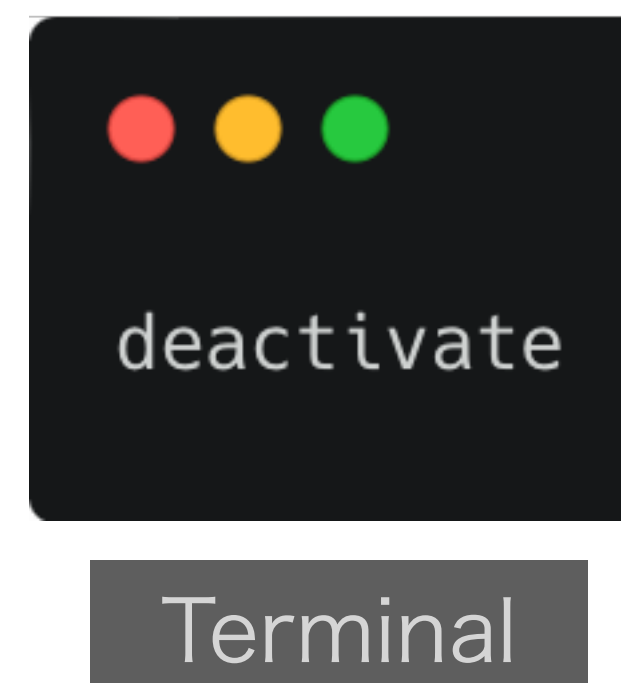
# Virtual Environment

## 仮想環境 (Mac/Linux)

- ・ 仮想環境の有効化



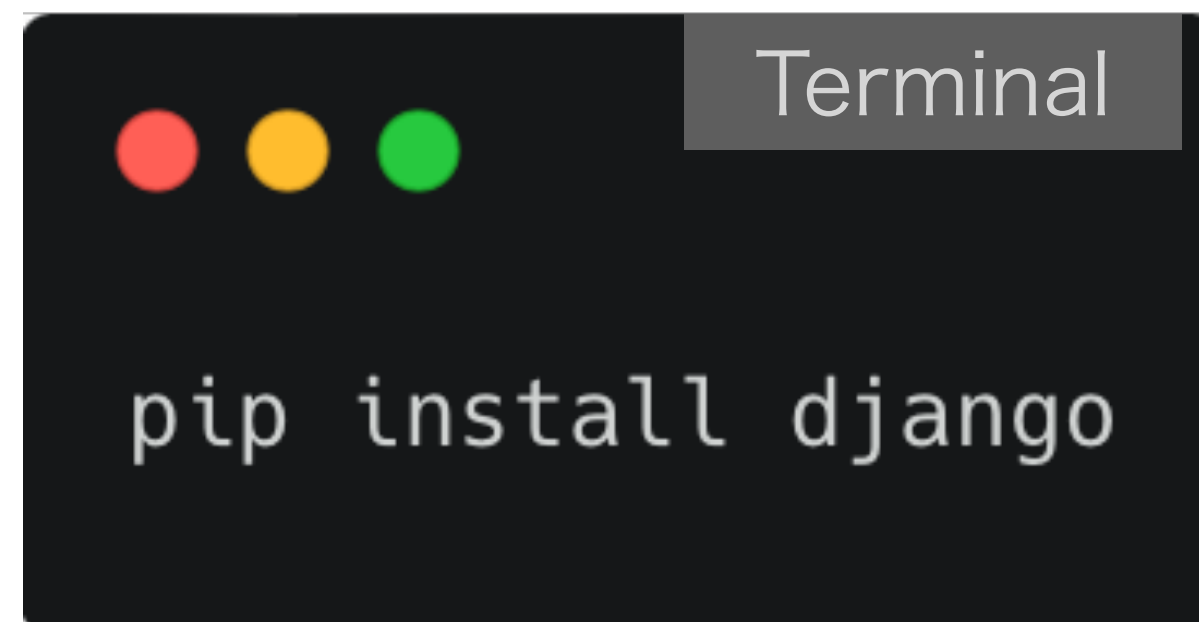
- ・ 仮想環境の無効化



# Install Django

## Djangoのインストール

- ・ 仮想環境を有効化した状態で、  
下記コマンドを実行し**Django**をインストールする

A terminal window with a dark background and a title bar labeled "Terminal". The title bar has three colored window control buttons (red, yellow, green) on the left. The terminal displays the command `pip install django` in a light-colored monospaced font.

```
Terminal  
pip install django
```

# Start Project

## プロジェクトのスタート

- Djangoがインストールされた仮想環境内で以下のコマンドを実行し、**Djangoプロジェクトを開始する**
- 保存先フォルダを "." と指定することで  
カレントディレクトリにプロジェクトを作成できる

A terminal window with a dark background and a title bar labeled "Terminal". The window contains three lines of text: the first line is the command "django-admin startproject [プロジェクト名]", the second line is "or", and the third line is the command "django-admin startproject [プロジェクト名] [保存先フォルダ]".

```
Terminal  
  
django-admin startproject [プロジェクト名]  
  
or  
  
django-admin startproject [プロジェクト名] [保存先フォルダ]
```

# Start Project

## プロジェクトのスタート

- 最近は予めプロジェクト名に対応したフォルダーを作り、プロジェクト名を "config" として開発することがある

A terminal window with a dark background and a title bar labeled "Terminal". The window has three colored window control buttons (red, yellow, green) in the top-left corner. It contains three lines of text: `mkdir [プロジェクト名]`, `cd [プロジェクト名]`, and `django-admin startproject config .`.

```
Terminal

mkdir [プロジェクト名]

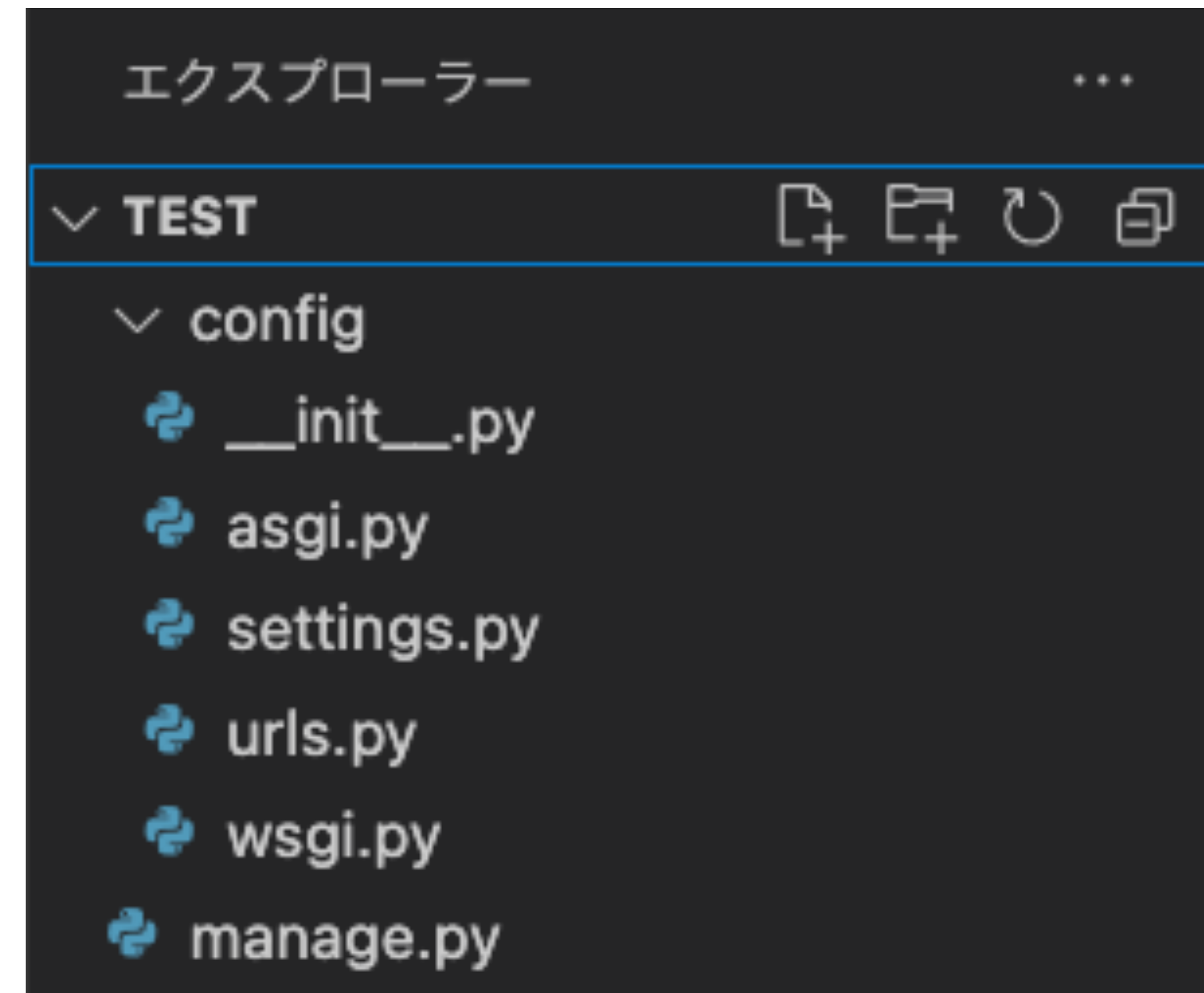
cd [プロジェクト名]

django-admin startproject config .
```

# Files composition

## ファイル構造

- config/settings.py は  
プロジェクトの設定を行う
- config/urls.py は  
URLルーティングを行う
- manage.py は  
サーバーホストやdb反映などを行う



# Run server

## ローカルサーバーの起動

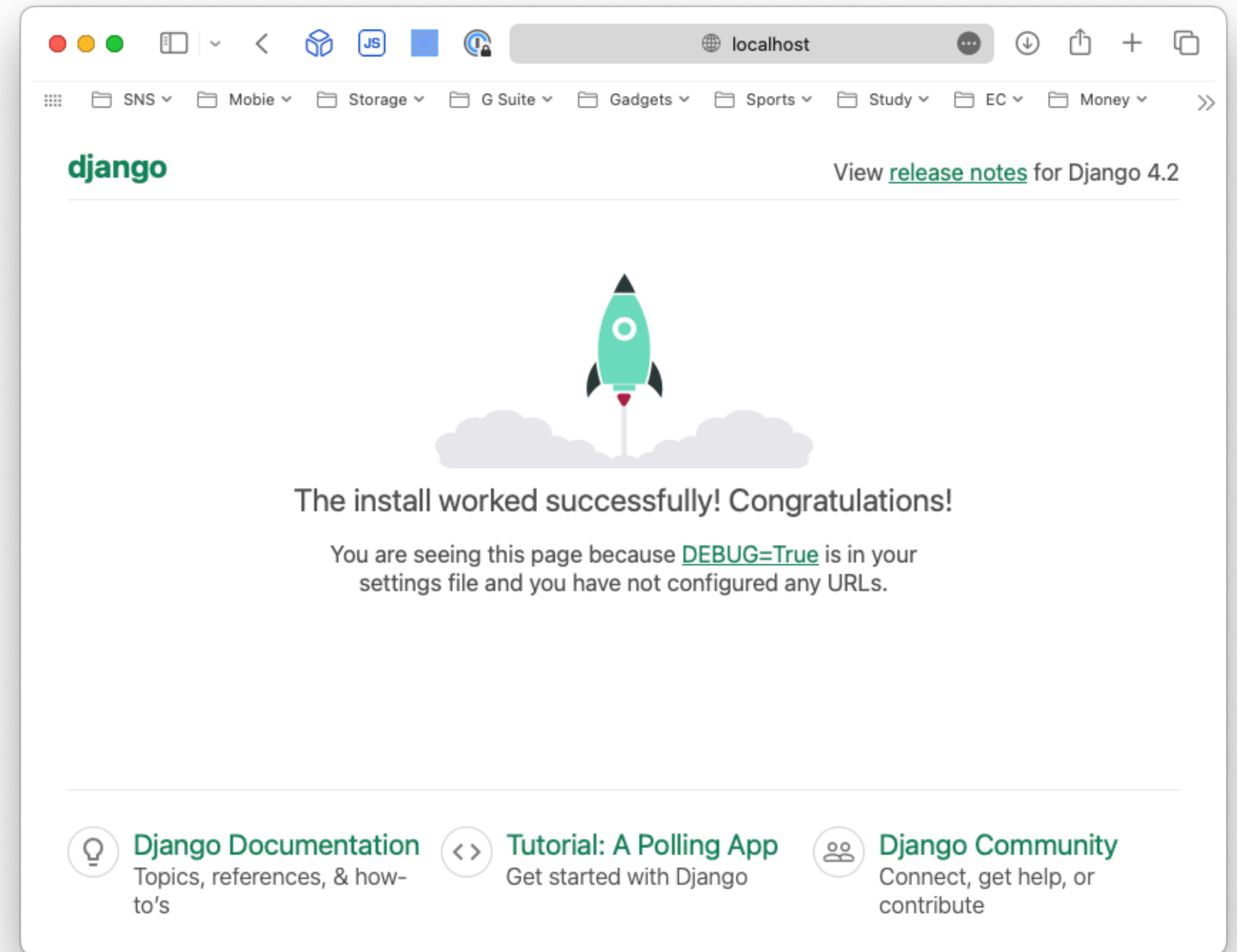
- 以下のコマンドを実行して、**Djangoサーバーを起動**する
- 何も指定しない場合は  
**Djangoデフォルトの8000番ポート**が解放される

```
Terminal

python manage.py runserver

or

python manage.py runserver [ポート番号]
```



localhost:8000 | Safari



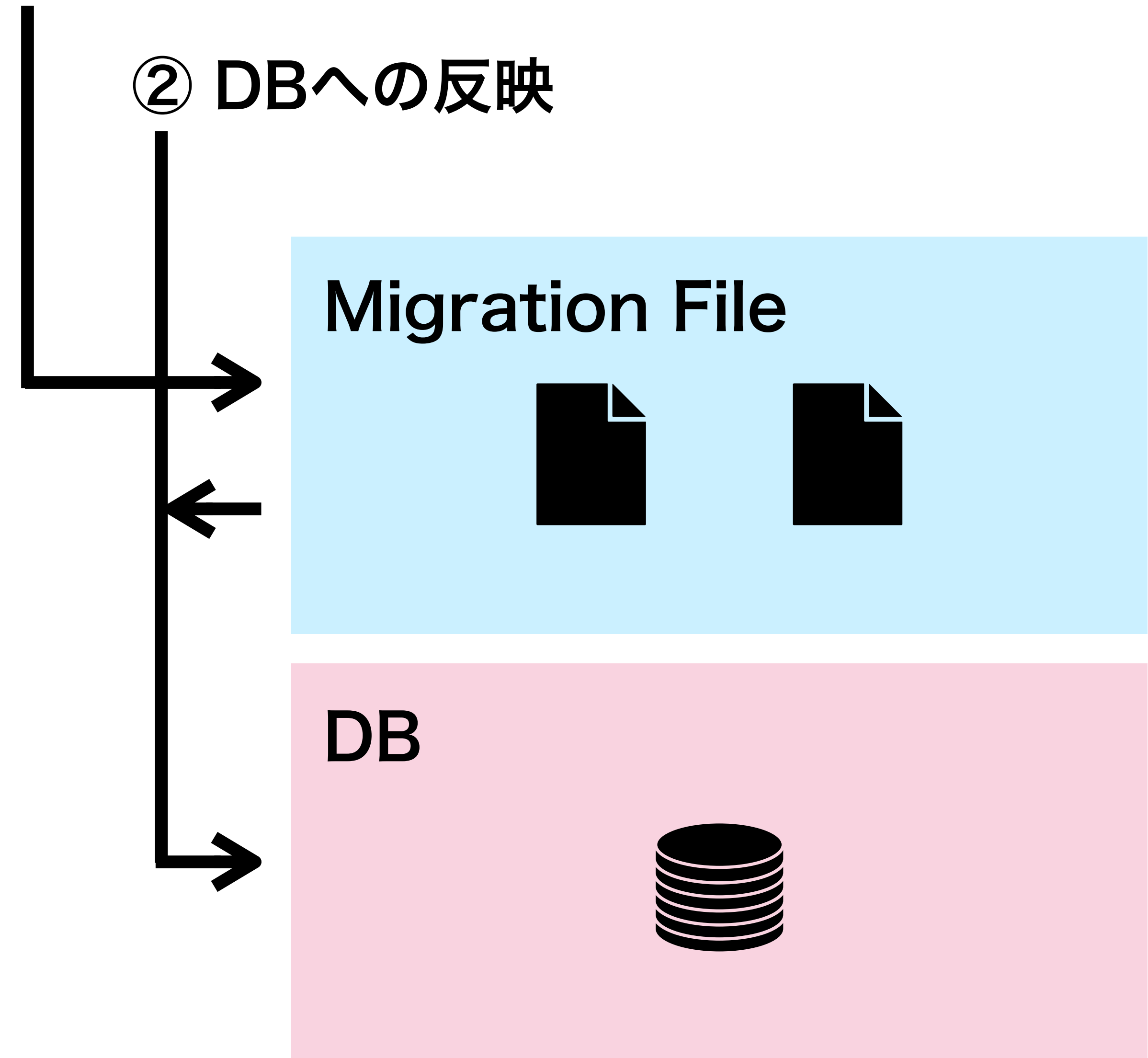
# Migrate

## マイグレート

- ・コード内で指定したモデルからDBに**反映させる処理**
- ・まずは**Migrate File**を作成し、その**ファイル**を元に**反映**する

### ① Migration Fileの作成

### ② DBへの反映



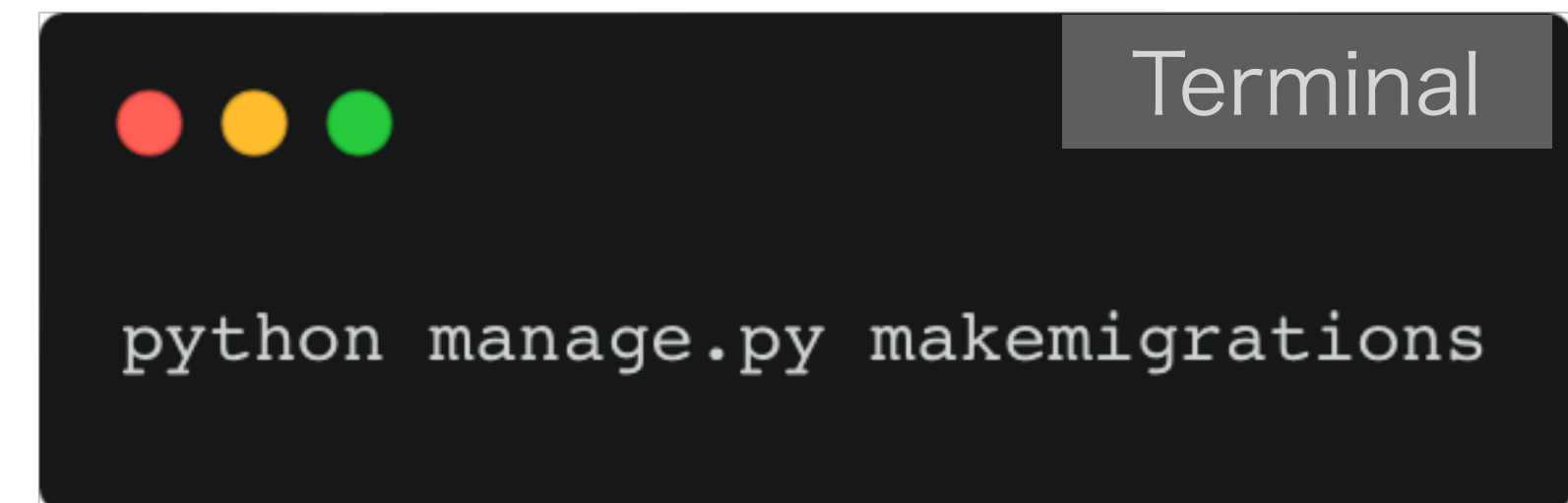


# Migrate

## マイグレート

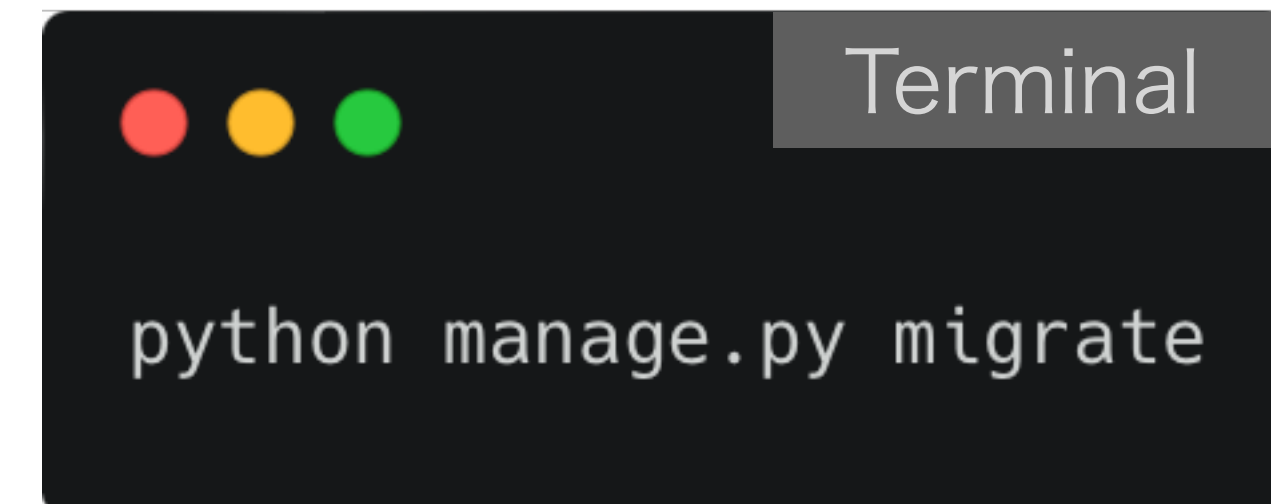
- ・コード内で指定したモデルからDBに**反映させる処理**
- ・まずは**Migrate File**を作成し、**そのファイル**を元に反映する

### ① Migration Fileの作成



```
python manage.py makemigrations
```

### ② DBへの反映



```
python manage.py migrate
```

# Create Application

## アプリケーションの作成

- Djangoではプロジェクト内に**複数のアプリケーション**を作れる
- 新規アプリケーションを作るには下記コマンドを実行する
- 例では **"tasks"** アプリを作成する

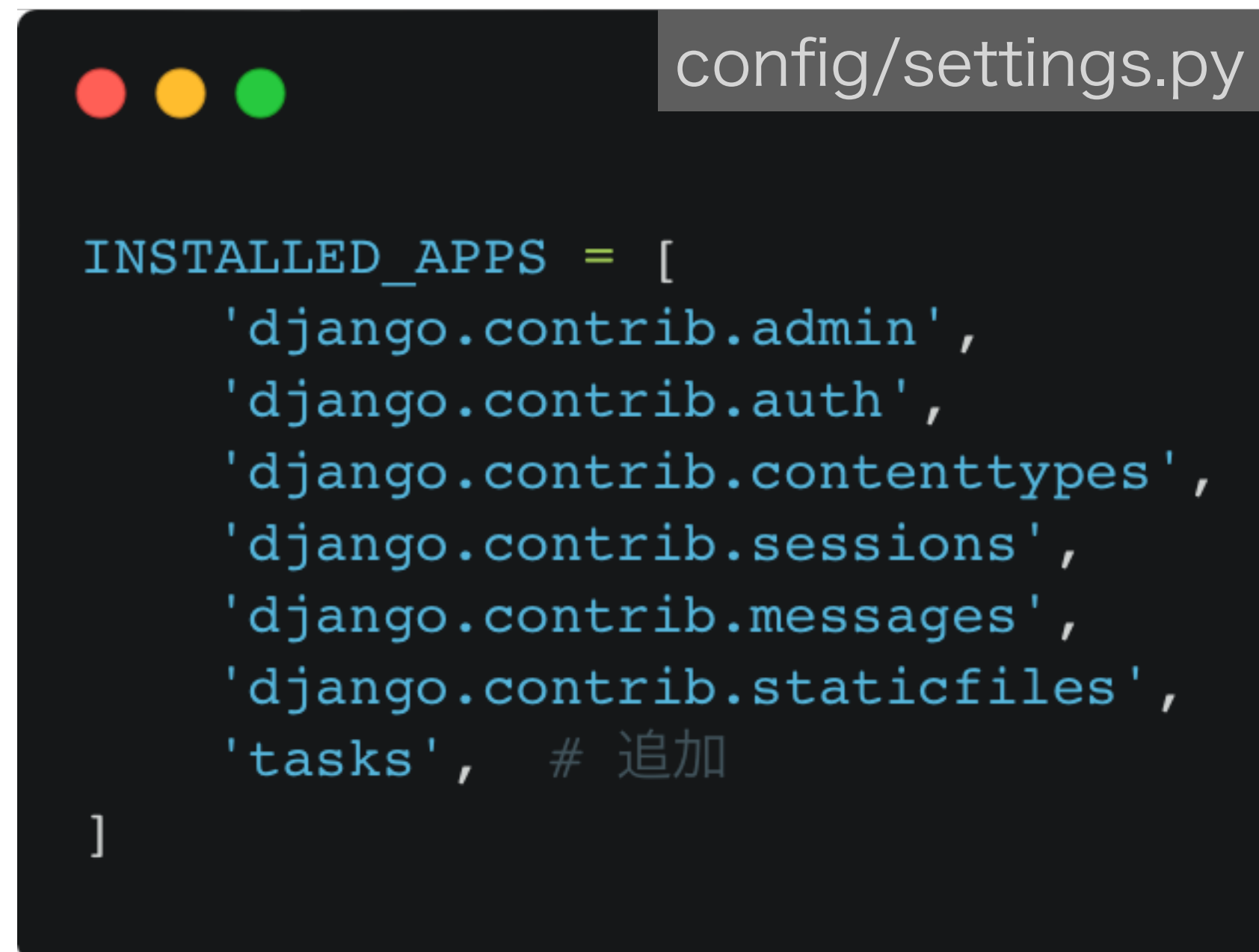
A terminal window with a dark background and a title bar labeled "Terminal". The title bar has three colored window control buttons (red, yellow, green) on the left. The terminal displays the command `python manage.py startapp [アプリケーション名]` in a light gray monospaced font.

```
python manage.py startapp [アプリケーション名]
```

# Create Application

## アプリケーションの作成

- settings.py 内にある INSTALLED\_APPS に  
作成したアプリケーションを追加する

A screenshot of a code editor window with a dark background. The title bar at the top right shows the file path 'config/settings.py'. The code is in Python and shows the 'INSTALLED\_APPS' list. It contains several Django core apps and a custom app named 'tasks' which is commented as '# 追加' (added).

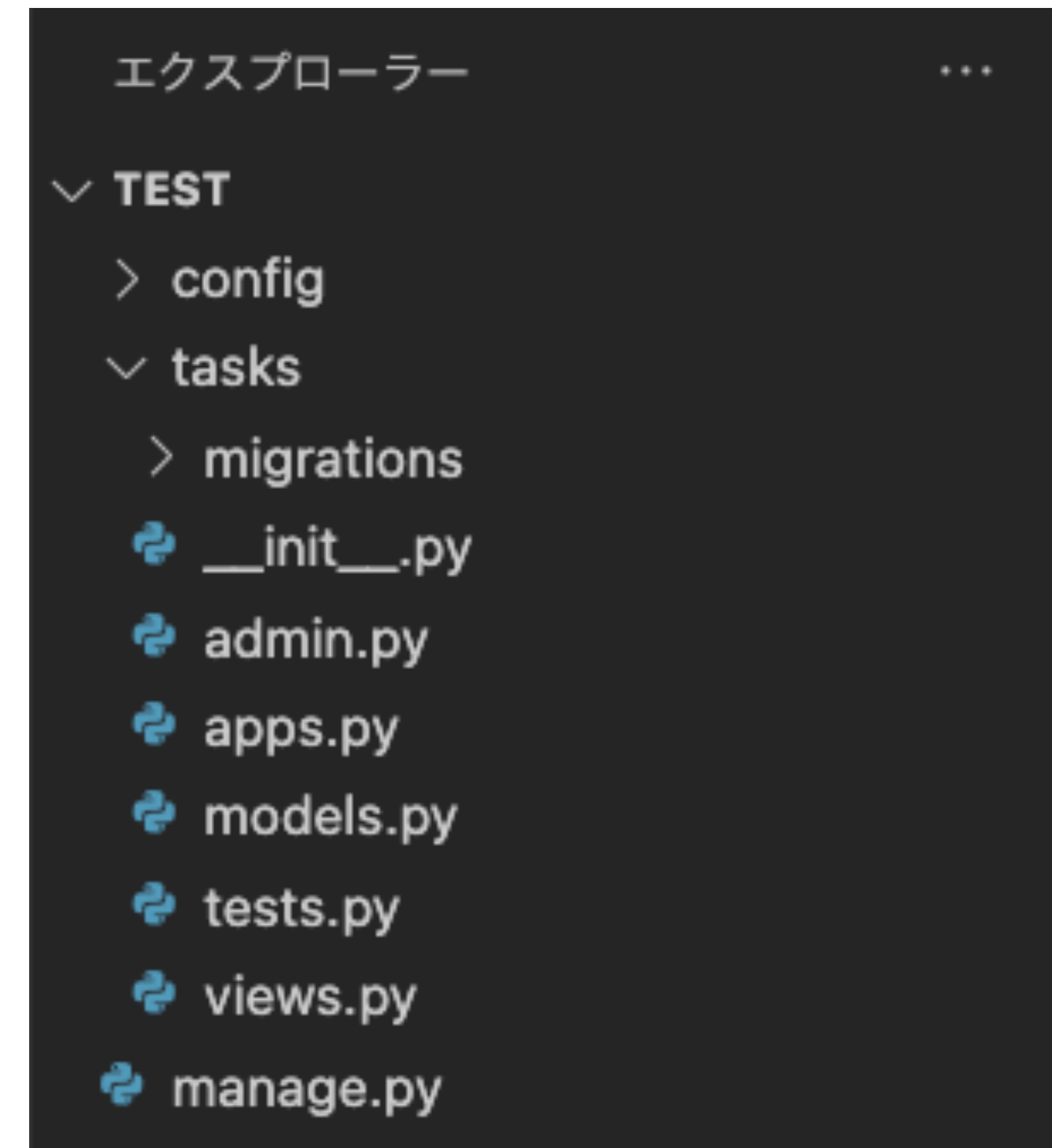
```
config/settings.py

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

# App composition

## アプリケーション構造

- 今回はアプリ「tasks」を作成
- tasks/models.py は  
**モデル管理**をするファイル
- tasks/views.py は  
**アプリ制御**をするファイル



# Create secret file

## 非公開ファイルを作成する

- settings.pyに記載された**公開してはいけない情報を非公開にする**
- 暗号化キーなどを非公開にし、安全性を高める
- GitHubから追加できるPythonデフォルトの.gitignoreより、**Djangoにおいてlocal\_settings.pyを非公開ファイルにすることが推奨**
- この操作は**Gitにコミットする前**に行う

# Details secret file

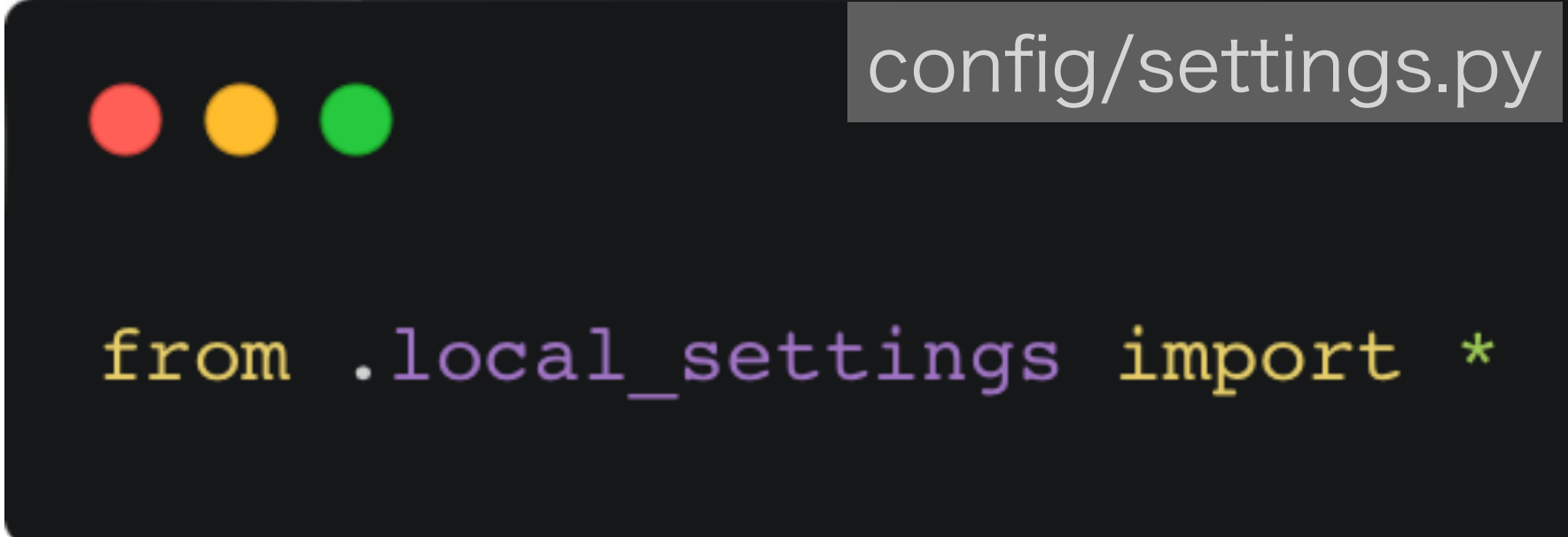
## 非公開ファイルの内容

- 以下の情報を**非公開**として扱うことが**推奨**される
  - BASE\_DIR
  - SECRET\_KEY
  - INSTALLED\_APPS
  - DATABASES
  - ALLOWED\_HOSTS

# Change settings file

## 設定ファイルの変更

- ・ 非公開ファイルに設定した項目を削除し、  
以下のように**非公開ファイルを読み込む**

A code editor window with a dark background. The title bar at the top right shows the file path 'config/settings.py'. On the left side of the title bar are three colored circles: red, yellow, and green. The main area of the editor contains a single line of Python code: 'from .local\_settings import \*'.


```
config/settings.py

from .local_settings import *
```

# Change URL Route

## ルーティングの変更

- ・ デフォルトのプロジェクトでのルーティングではなく、アプリごとに切り替える

A screenshot of a code editor window titled 'config/urls.py'. The code is written in Python and shows the Django URL configuration. It imports 'path' and 'include' from 'django.urls', and defines 'urlpatterns' as a list containing a single path that includes 'tasks.urls'.

```
from django.urls import path, include

urlpatterns = [
    path("", include("tasks.urls")),
]
```

config/urls.py

A screenshot of a code editor window titled 'tasks/urls.py'. The code is written in Python and shows the Django URL configuration for the 'tasks' app. It imports 'path' from 'django.urls' and 'views' from the current directory. It defines 'urlpatterns' as a list containing a single path that points to a view function named '[関数名]' (function name).

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

urlpatterns = [
    path("", views.[関数名]),
]
```

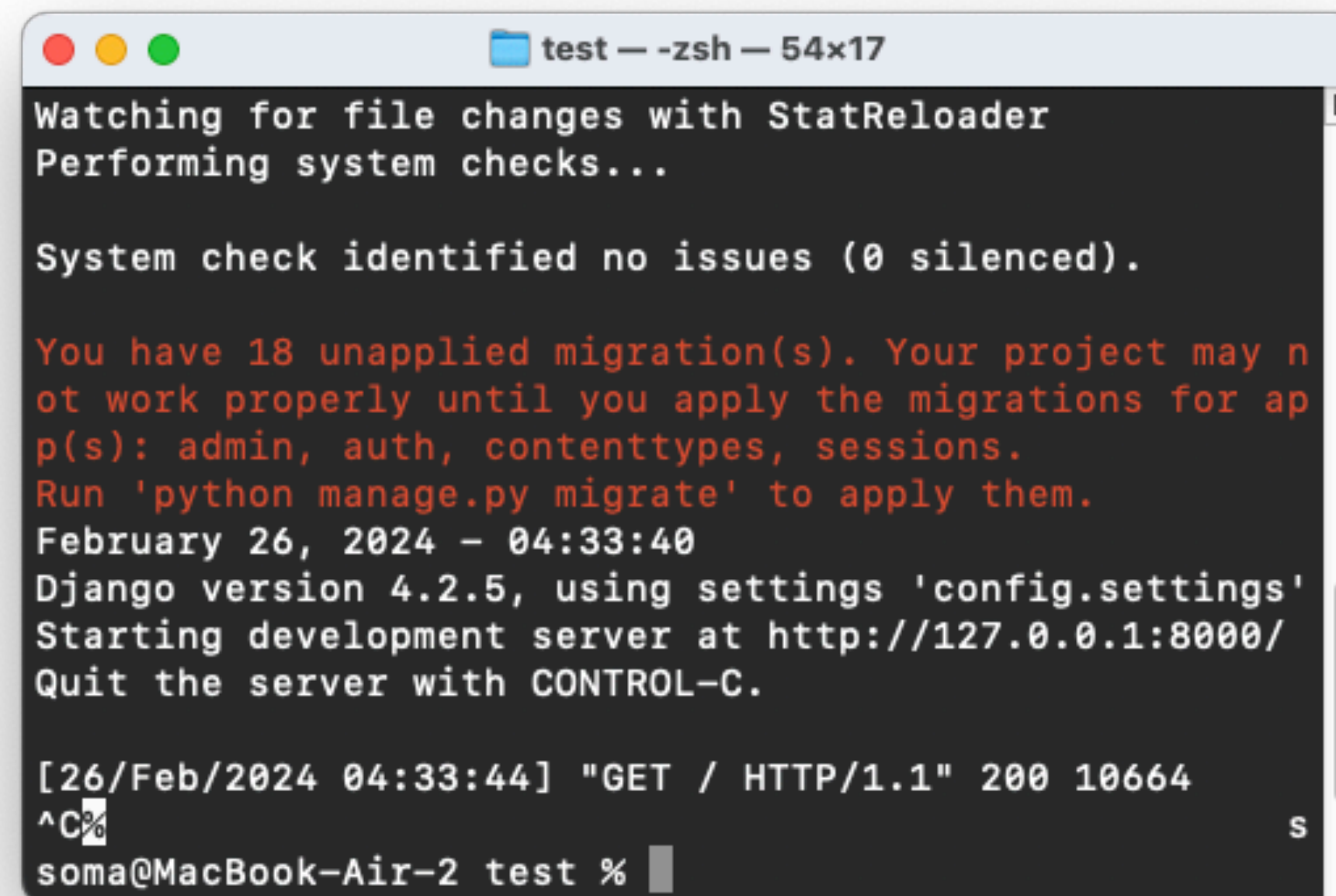
tasks/urls.py



# Stop server

## ローカルサーバーの停止

- 作業終了時にはターミナル上でMac OS/Windows両OSとも  
"control + c" を押して、サーバーを停止する



```
test — -zsh — 54x17
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
February 26, 2024 - 04:33:40
Django version 4.2.5, using settings 'config.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.

[26/Feb/2024 04:33:44] "GET / HTTP/1.1" 200 10664
^C%
soma@MacBook-Air-2 test %
```