MVCとは？

Djangoは、「MVCアーキテクチャー」と呼ばれる考え方に基づいて設計されています。

|  |  |
| --- | --- |
| Model | Webアプリケーションとデータベースとの間のやりとりを担当するもの |
| View | 画面表示関係を担当するもの。画面に表示させるWebページの部分 |
| Controller | 全体の制御を担当するもの。 |

プログラムの中心となるのは、コントローラーです。コントローラーの中で、必要に応じてモデルを呼び出しデータを受け取ったり、ビューを呼び出して画面に表示するWebページを作ったりする。モデルやビューを必要に応じて呼び出しながら、全体の処理をコントローラーで進めていく。

ユーザー

データベース

画面表示

モデル

ビュー

コントローラー

Webにアクセス

結果を表示

Djangoは、MVCがセットになってプログラムが構築されます。アプリケーションを作成すると、そのためのMVC関係のプログラムがセットで追加されます。

　このアプリケーションはプログラムの中にいくつでも追加することができます。つまり、MVCのセットをいくつでも用意することができます。１つのプロジェクトの中にいくつでも異なるプログラムを組み込むことができる。

＊例

|  |
| --- |
| オンラインショップ(プロジェクト) |
| カート管理アプリケーション  在庫管理アプリケーション  ユーザー管理アプリケーション |

アプリケーションを作成するコマンド

python manage.py startapp 名前　　　　　＊「名前」に作成したいアプリケーション名

Djangoプロジェクトの中身

|  |  |
| --- | --- |
| \_\_initi\_\_.py | Djangoプロジェクトを実行するときの初期化処理を行うスクリプトファイル |
| asgi.py | ASGIという非同期Webアプリケーションのためのプログラム |
| settings.py | プログラムの設定情報を記述したファイル |
| urls.py | プロジェクトで使うＵＲＬを管理するファイル |
| wsgi.py | WSGIという一般的なWebアプリケーションのプログラム |

・manage.py

このプロジェクトで実行する様々な機能に関するプログラム。Djangoでは、コマンドでプロジェクトをいろいろ操作しますが、そのための処理がここに書いてある。

作成したアプリケーションの中身　(python manage.py startapp 名前)

|  |  |
| --- | --- |
| 「migrations」フォルダ | マイグレーションといって、データベース関係の機能のファイルをまとめる |
| \_\_initi\_\_.py | アプリケーションの初期化処理のためのもの |
| admin.py | 管理者ツールのためのもの |
| apps.py | アプリケーション本体の処理をまとめる |
| models.py | モデルに関する処理を記述するもの |
| tests.py | プログラムのテストに関するもの |
| views.py | 画面表示に関するもの |

よくあるエラー

<https://blog.kikagaku.co.jp/python-basic-error>