

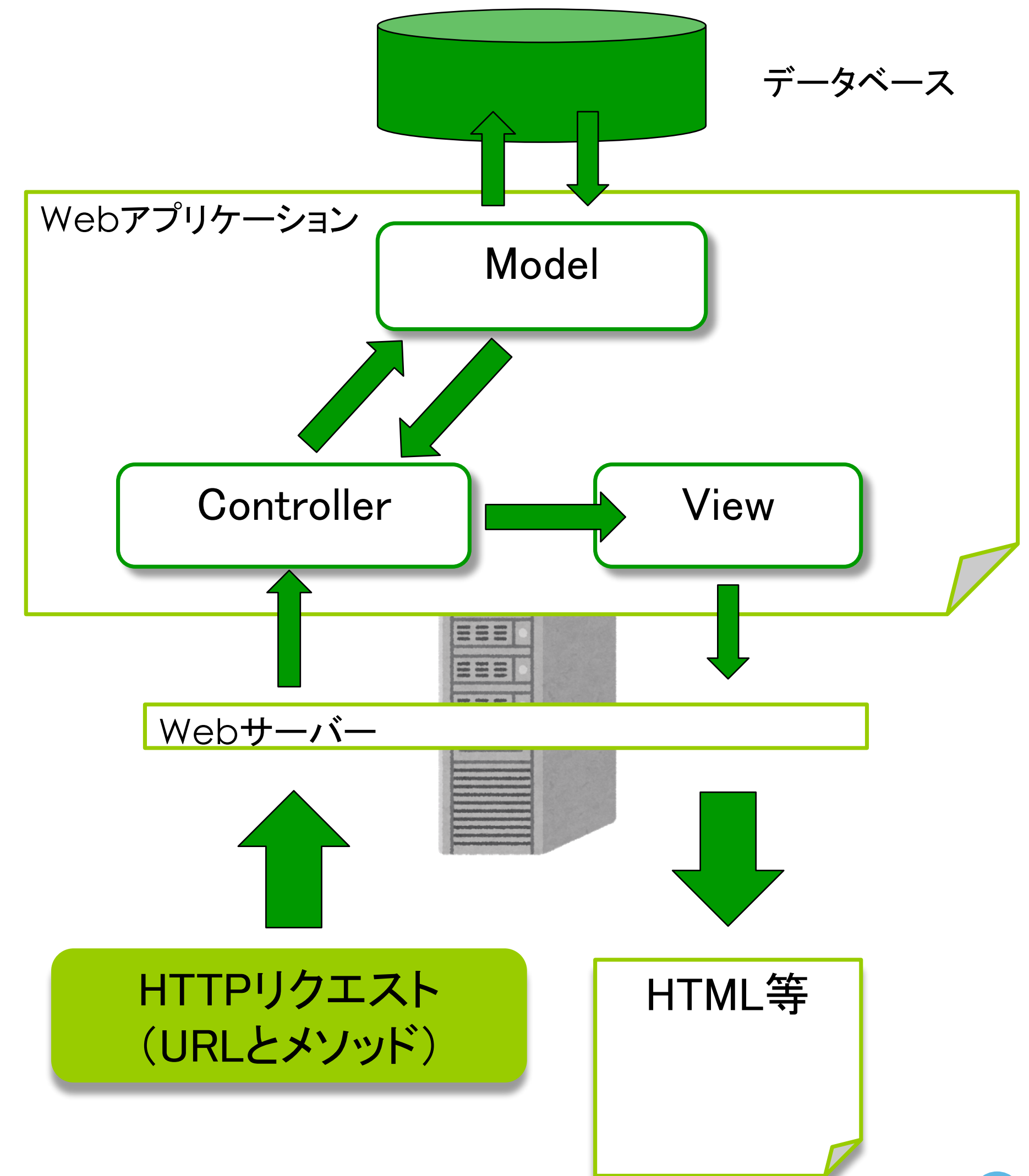
②

DjangoにおけるMVC

前回紹介したMVCの概念がDjangoでどのようなものかを紹介します。

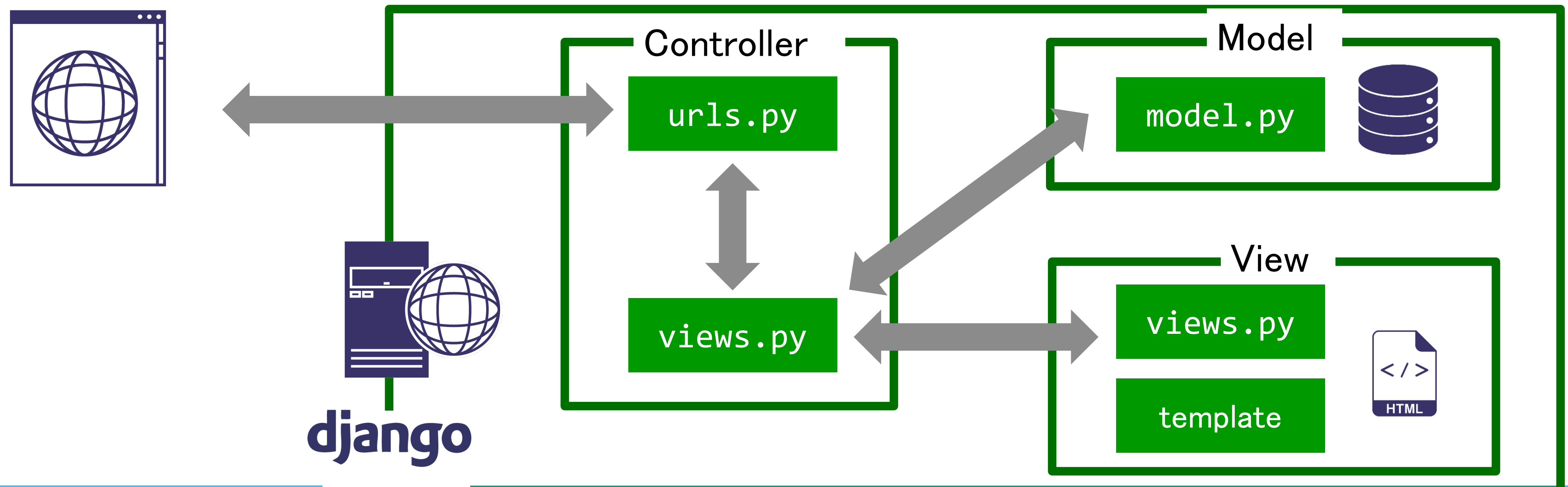
(復習) MVC (Model-View Controller)とは

- 以下の3コンポーネントからなるデザインパターン
 - Model : データと付随するロジック
 - View : 情報の表示・出力
 - Controller : 入力を受け付け、ModelやViewを制御
- Webサービスの場合であれば、右のようになる
 - ※概念は、Webサービス固有ではないことに注意



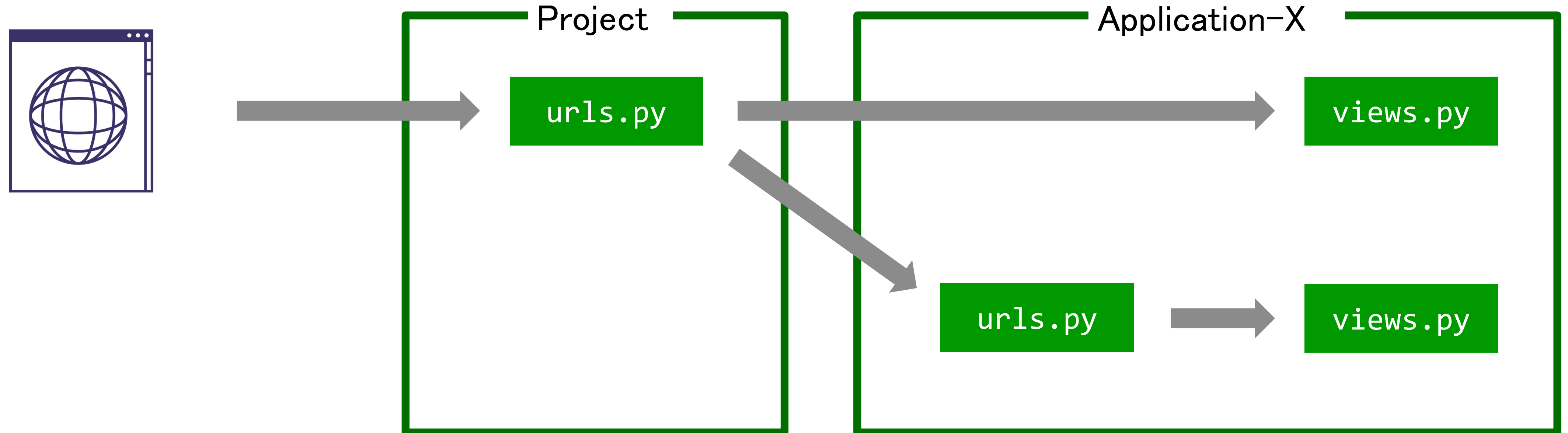
DjangoのMVC(Model, View, Controller)

- DjangoでのMVCは、以下のような分担で連携して動作します
 - Controller: `urls.py` がURLで振り分けて、`views.py` が処理をする
 - Model: `model.py` がデータベースのエントリをクラスとして抽象化
 - View: `views.py` (Controllerも兼ねる)がテンプレートからHTMLを作成
- (参考) Djangoでは、MVCをMTV (Model, Template, View)と呼びます
 - MVCのV がTemplateに、MVCのC がViewに対応する



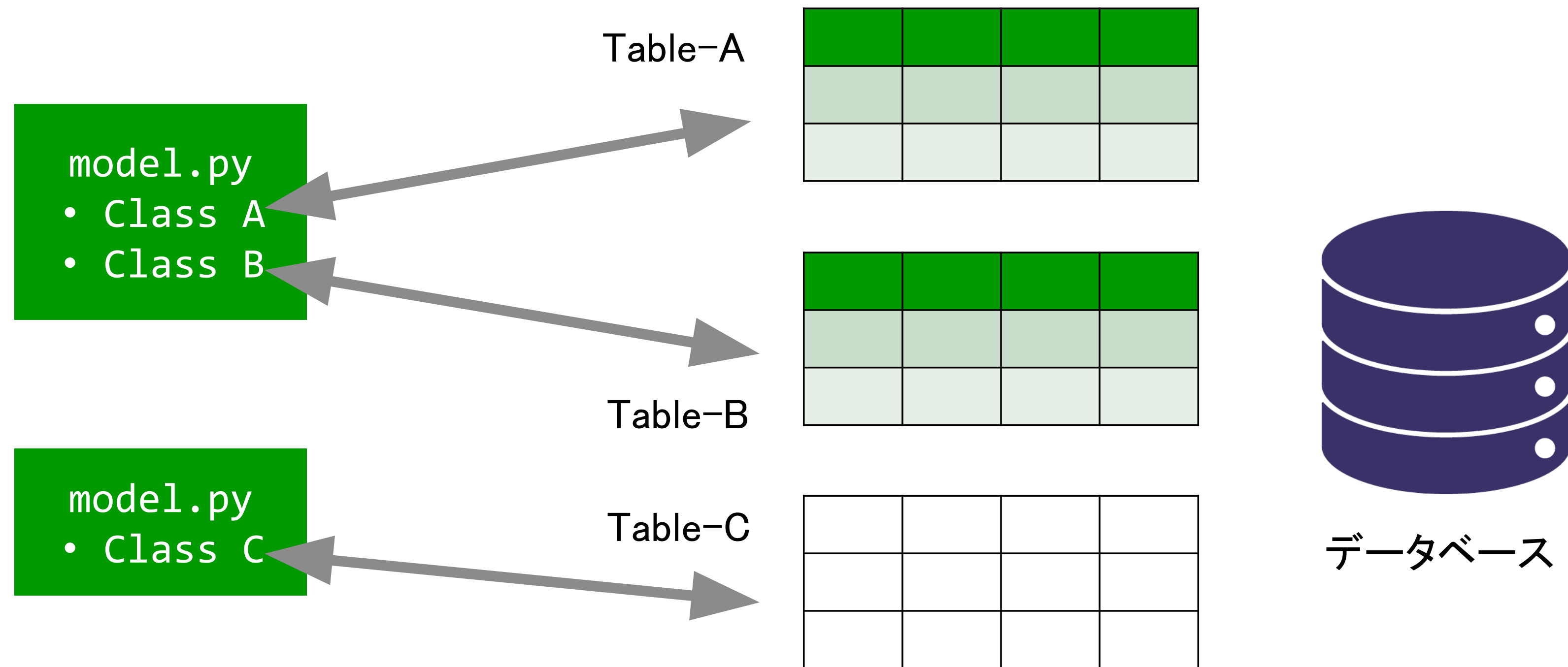
Controller

- Webアプリケーションにアクセスされると、最初にプロジェクトの`urls.py` が参照されます
 - アクセスされたURLにマッチするパスが定義されていれば、対応する `views.py` の関数かクラスのメソッドが呼び出されます
 - 設定によっては、アプリケーション毎に定義された `urls.py` に転送されることもあります



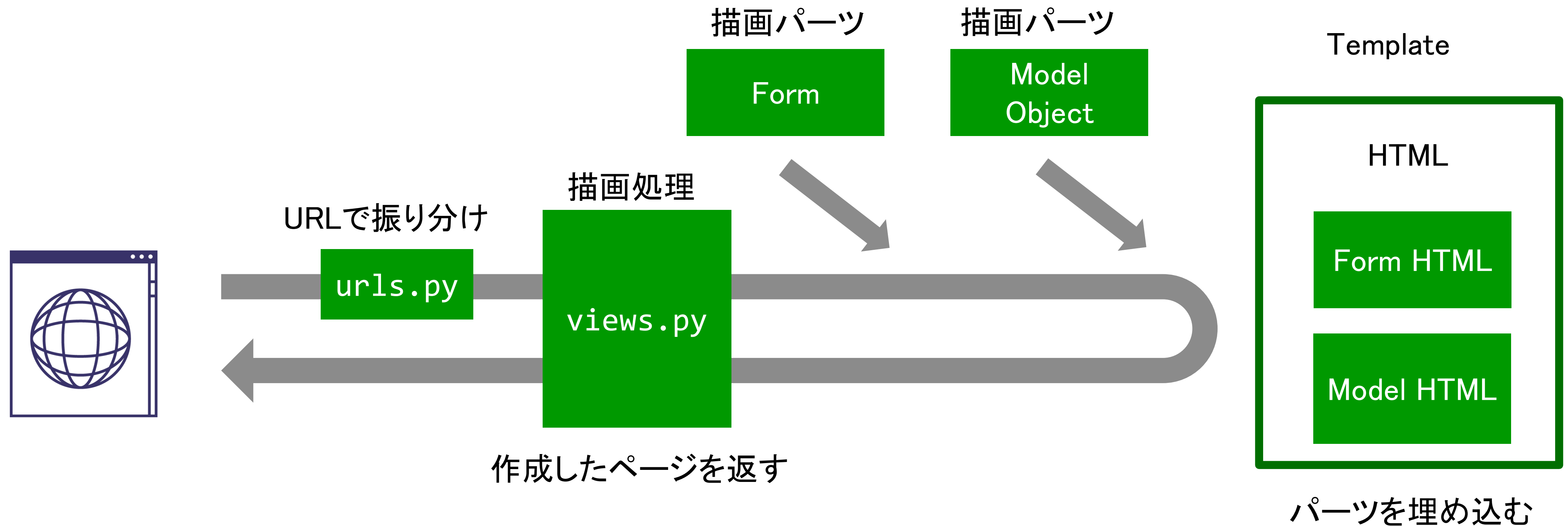
Model (詳細は次週)

- **models.py** には、データベースのスキーマが記述されており、**views.py** から利用されます
 - オブジェクト関係マッパ(Object Relation Mapper, ORマッパー)の機能で、データベース上のデータを、Pythonのオブジェクトのように扱うことができます
- データベースのテーブルも、Modelクラスの定義から生成します



View

- 最終的に、**views.py** がHTML等を生成してレスポンスとして返します
 - この時、テンプレートシステムを用いて、テンプレートにパーツを埋め込みます



MVCとファイルの対応

- (少し名前が分かりにくいですが) DjangoのMVCは、以下のように対応していると思われる
 - Model → `blog/models.py`
 - Controller → `blog/views.py`
 - View → `blog/templates/blog/***.html`

