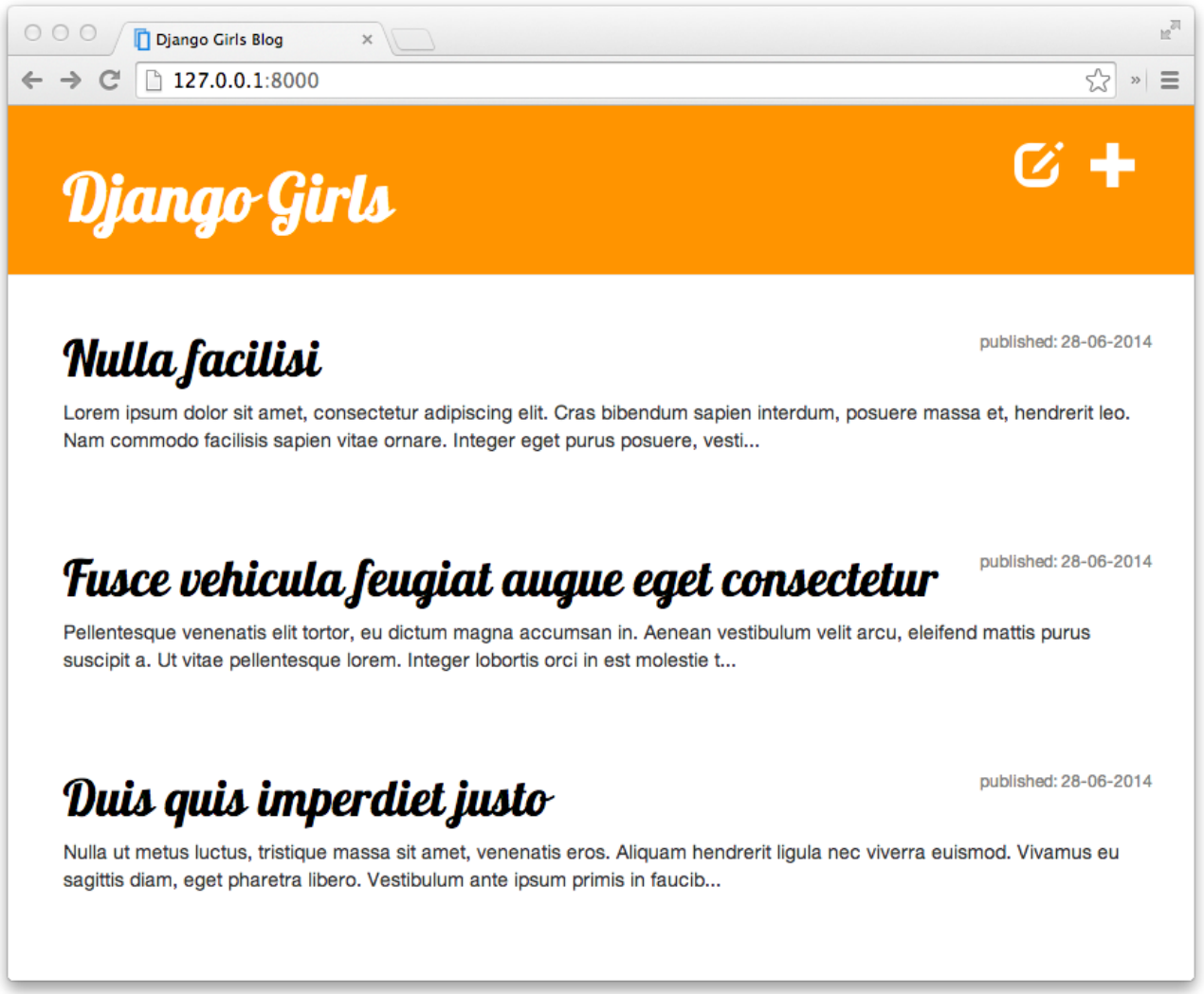
django

PythonのWebフレームワークから学ぶWebアプリケーションの作り方とイメージ

参考資料



Django Girls

こちらのWebサイトを元に本日のワークを行いたいと思います。

https://tutorial.djangogirls.org/ja/

Djangoでのアプリケーションの作り方

1. 環境設定をする settings.py

   Python、仮想環境の準備、Djangoのインストール、データベースとの紐づけ

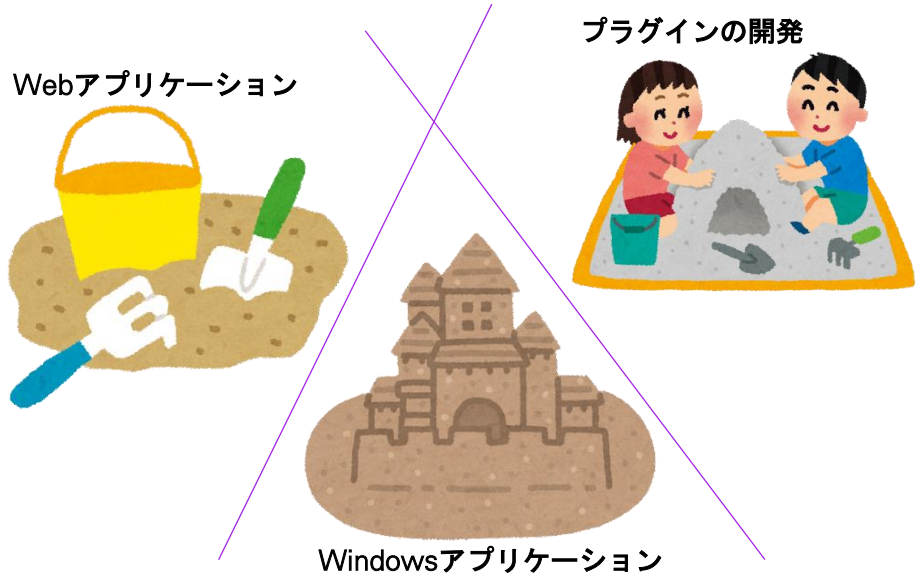
2. モデルを設定する models.py
3. URLによって処理が変わるように設定する urls.py
4. 処理を定義する views.py
5. 表示するページを作成する html
6. フォームの設定 forms.py

環境設定１

1. Pythonのインストール
   1. Pythonのダウンロード
      パソコンで使えるようにするための情報を取得する
   2. インストール
      パソコンで使えるようにする

2. フォルダを作って仮想環境を準備する

   仮想環境とは、砂場と思ってください。この時、砂場で何をしても、他の砂場と区切られているので、他の砂場に全く影響を及ぼしません。パソコンの中でも、領域を区切って開発を行います。



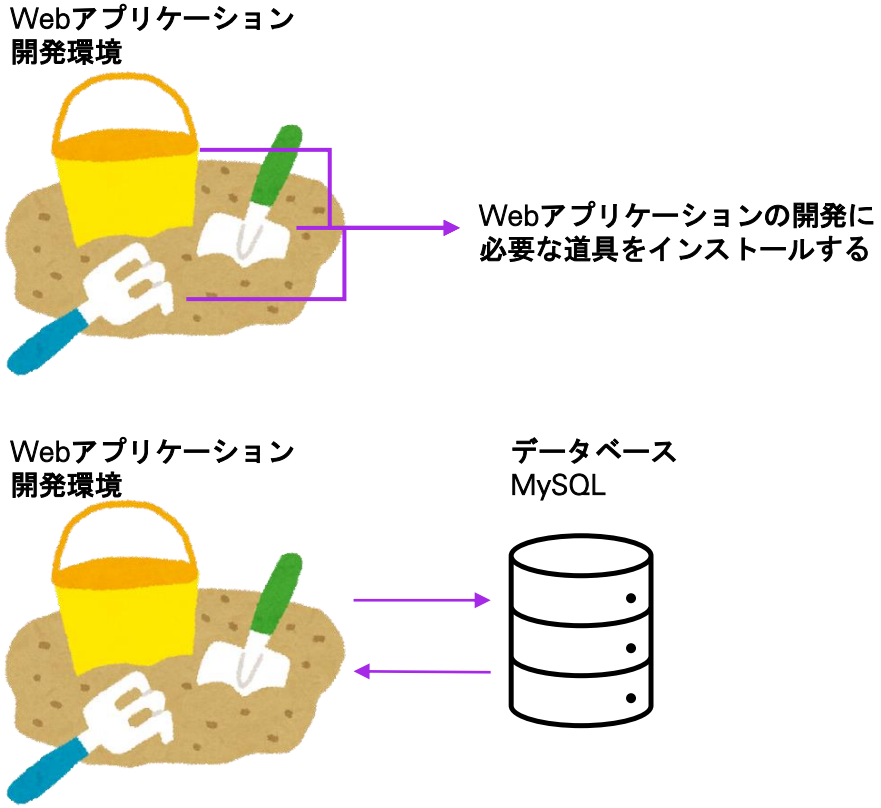
環境設定２

３．Djangoのインストール

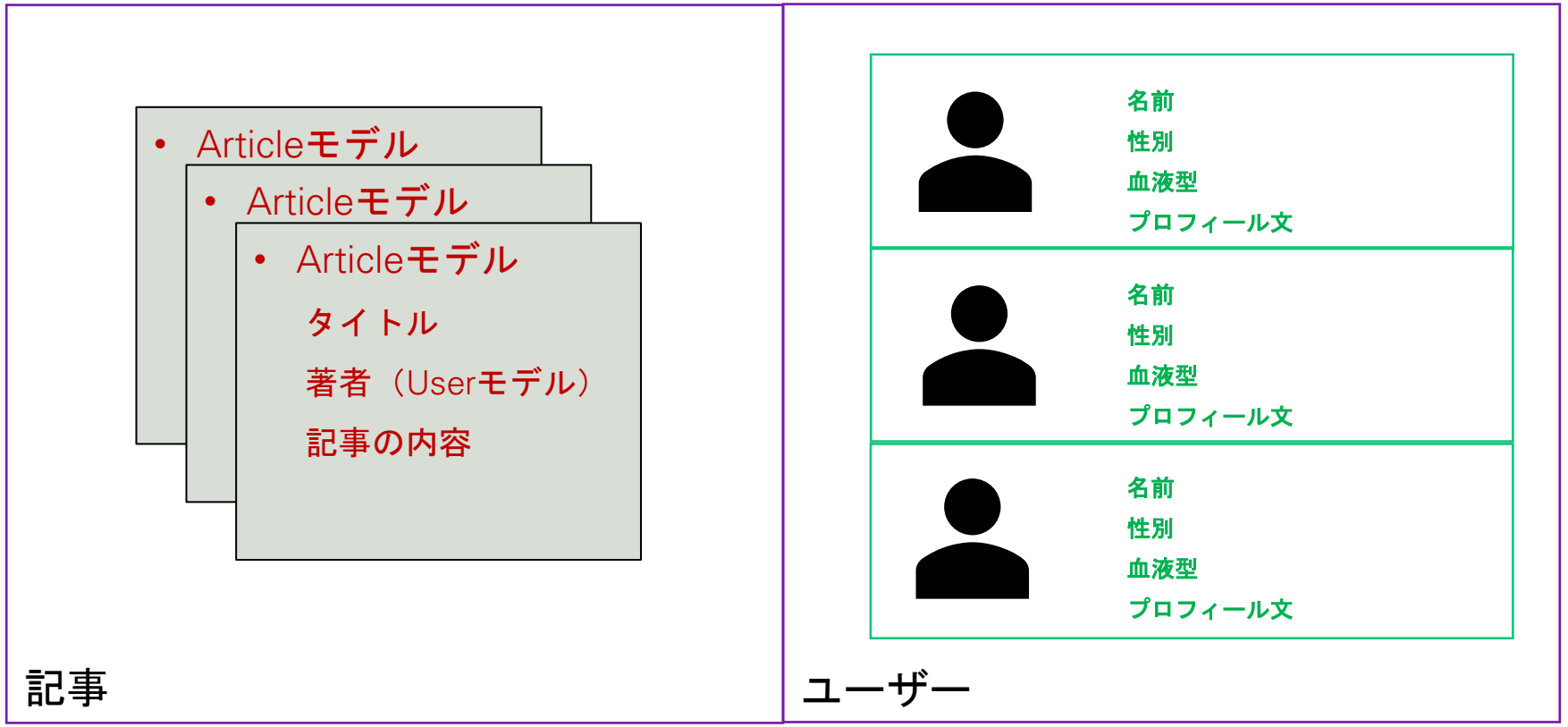
1. 用意した砂場（仮想環境）に入ります。
2. 開発に必要な道具をインストールしましょう。
   今回はWebアプリケーションに必要なDjango
   というライブラリをインストールします。

４．データベースとの紐づけ

Webアプリケーションではデータベースを使うことが多いです。データベースは一時的ではなく、ずっとデータを保持しておけるのと、オンラインのデータベースでは、他のユーザーとデータを共有することができます。

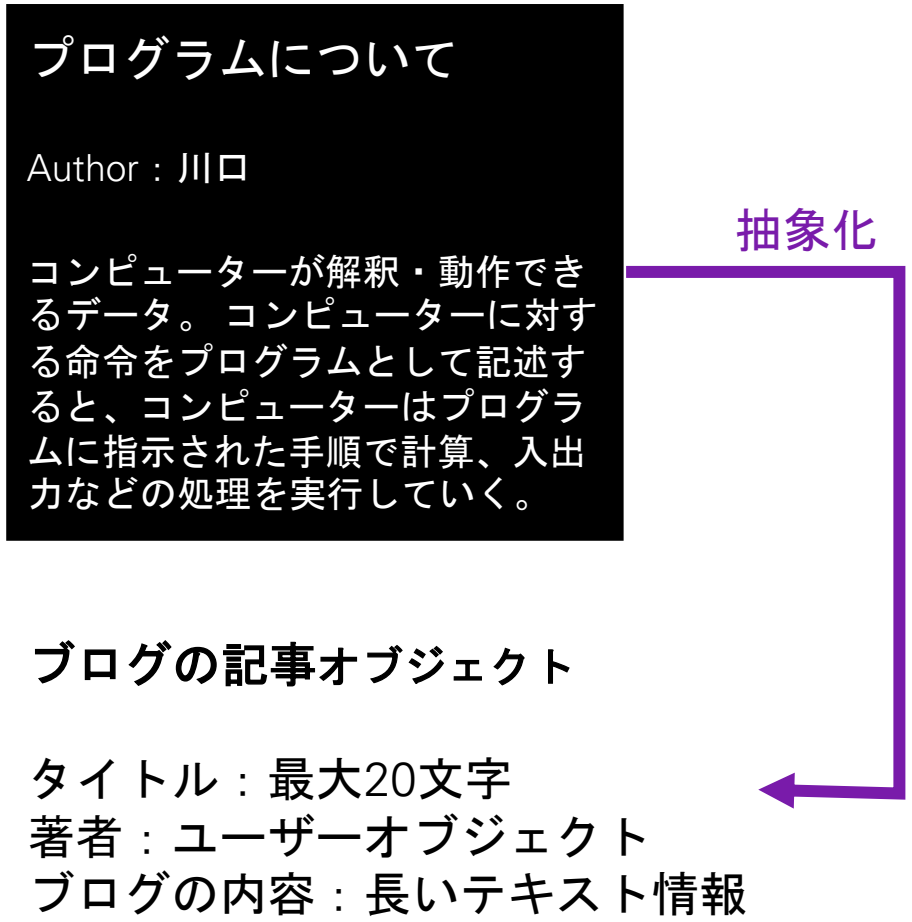


管理したい情報を考える



オブジェクト指向

- 現実でまとまりとして捉えている情報を、プログラムの中で表現すること。その考え方をオブジェクト指向と言います。
- 今回アプリケーションで扱う情報のまとまりは２つです。ブログの記事、ユーザーです。
- 早速、表現してみましょう。表現するには、それぞれがどのような情報を持っているか分解して考える必要があります。

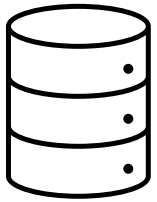


モデルを設定するメリット

- Articleモデル
  - タイトル
  - 著者（Userモデル）
  - 記事の内容

- Userモデル
  - 名前
  - 性別
  - 血液型
  - プロフィール文

データベース



テーブル１：Article

テーブル２：User

モデルの定義に基づいて自動でテーブルを生成してくれる

ブラウザ上での表示

タイトル

著者.name

ブログ記事の内容

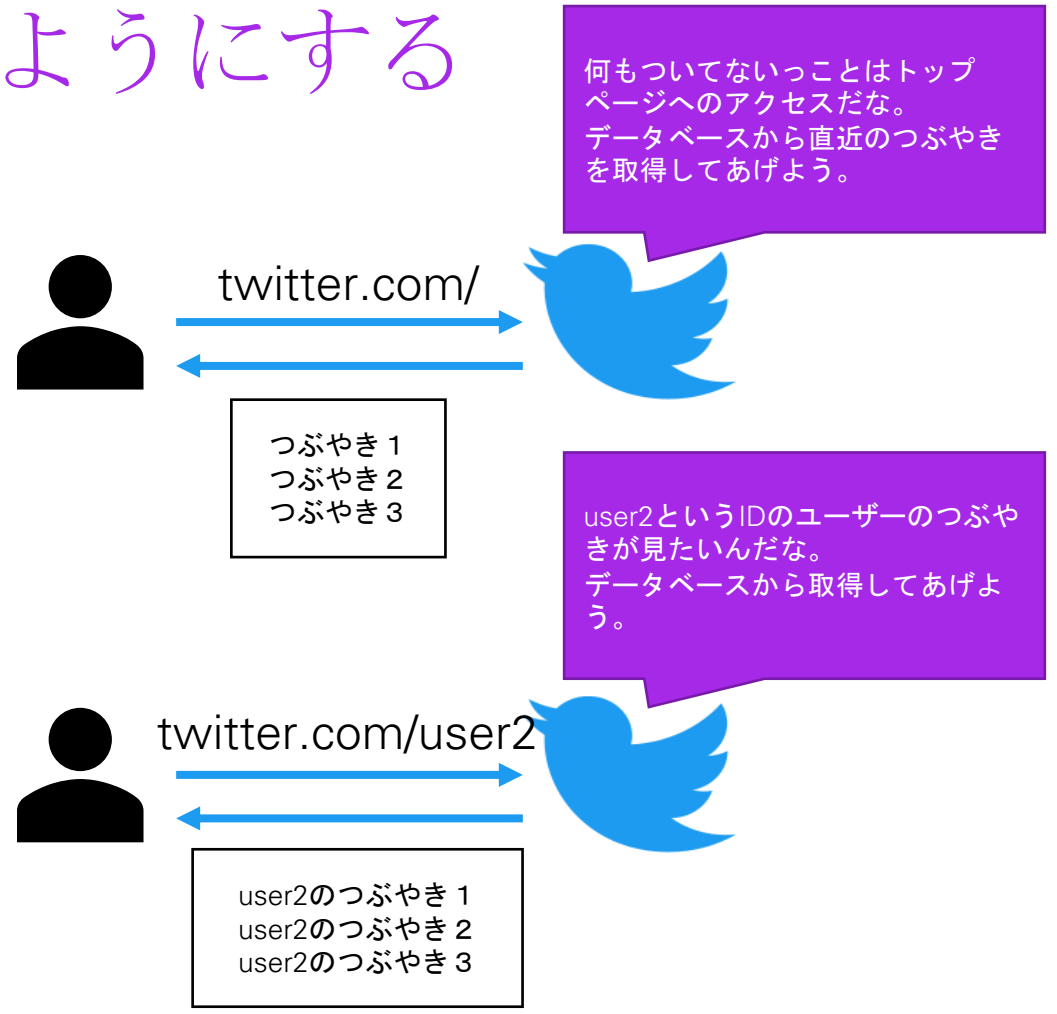
情報を取り出すときも、article.titleなど簡単に取り出すことができる

名前
性別
血液型
プロフィール文

ArticleとUserが結びついているので、クリックしたらユーザー情報が表示されるなどの連携がとれる

URLによって処理が変わるようにする

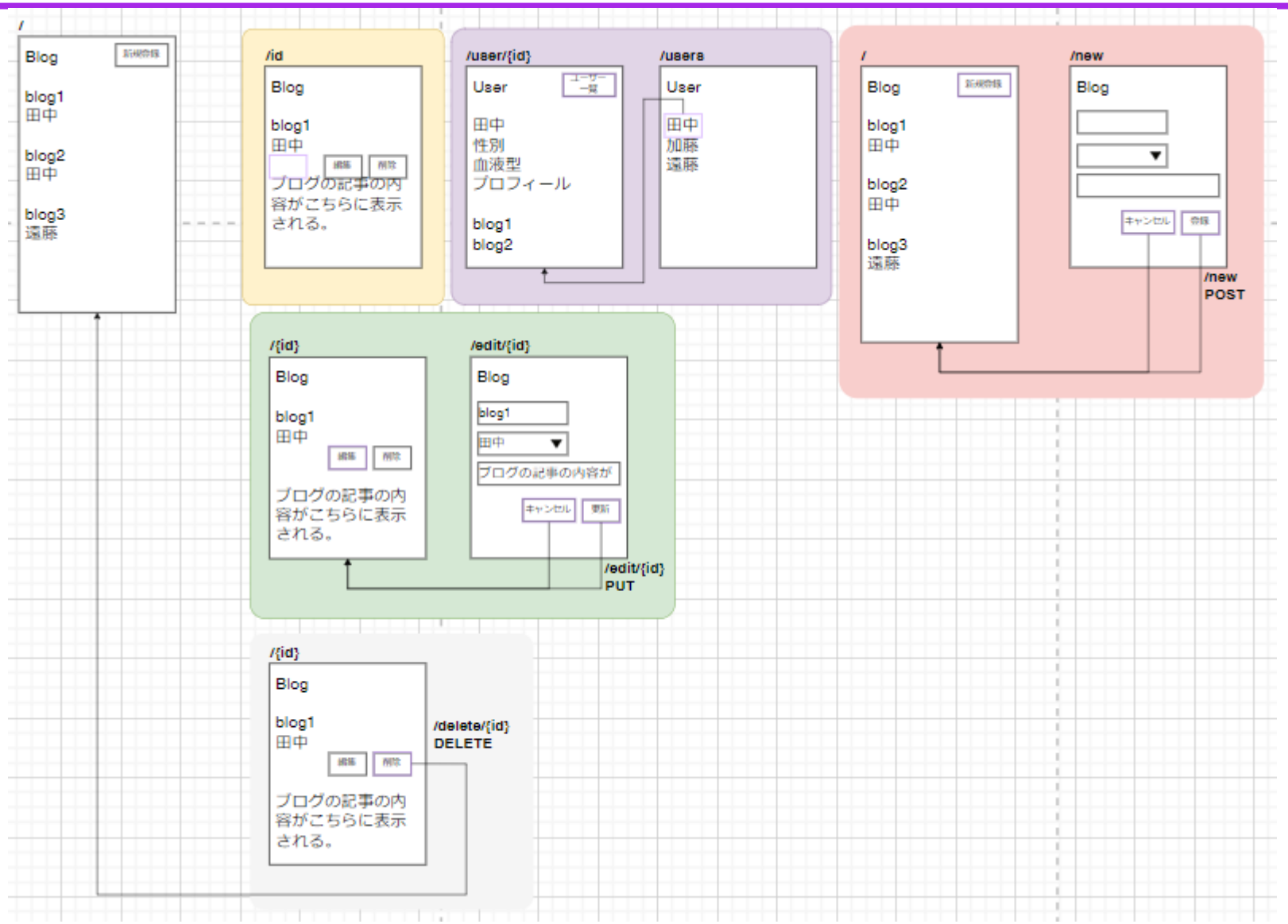
- 例えば、「twitter.com/」にアクセスすると、最新のつぶやきが一覧で表示されます。
- 「twitter.com/user2」にアクセスすると、user2のつぶやきが一覧で表示されます。
- このように、URLによってサーバーで動くアプリケーションから返ってくる情報は異なります。
- クライアントがアクセスしようとするURLによって、アプリケーションの動作を規定する必要がありますね。



URLによって処理が変わるようにする２

URL	機能	メソッド	通信
/	ブログが一覧で表示される	blogs	GET
/id	特定のブログ記事の内容を表示	blog	GET
/new	ブログの投稿フォームを表示	new	GET
/new	ブログの新規投稿	new	POST
/edit/id	ブログの記事の編集画面を表示	edit	GET
/edit/id	ブログの内容を更新	edit	POST
/delete/id	ブログの記事を消去	delete	POST
/users	ユーザーが一覧で表示される	user_list	GET
/user/id	特定のユーザーが表示される	user_detail	GET

画面遷移を考える

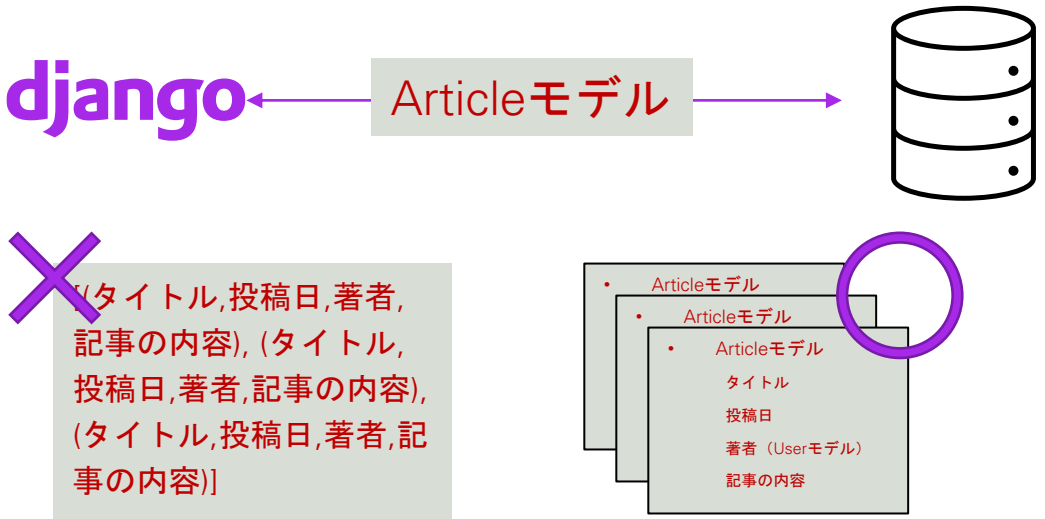

/
Blog
新規登録
blog1
田中
blog2
田中
blog3
遠藤
/id
Blog
blog1
田中
編集
削除
ブログの記事の内容がこちらに表示される。
/user/{id}
User
ユーザー一覧
田中
性別
血液型
プロフィール
blog1
blog2
/users
User
田中
加藤
遠藤
/
Blog
新規登録
blog1
田中
blog2
田中
blog3
遠藤
/new
Blog
キャンセル
登録
/new
POST
/{id}
Blog
blog1
田中
編集
削除
ブログの記事の内容がこちらに表示される。
/edit/{id}
Blog
blog1
田中
ブログの記事の内容が
キャンセル
更新
/edit/{id}
PUT
/{id}
Blog
blog1
田中
編集
削除
ブログの記事の内容がこちらに表示される。
/delete/{id}
DELETE

処理を定義する（リクエストの処理）

- アクセスされたURLによって処理を分ける所まで出来ました。
- ただし、GETやPOSTの判定や、パラメータの受け取りについてはまだ実装できていませんね。
- クライアントから来たリクエストについて詳しく分析し、処理を更に分割することができます。
- ここでは新規登録とブログの詳細画面の表示を例に説明したいと思います。

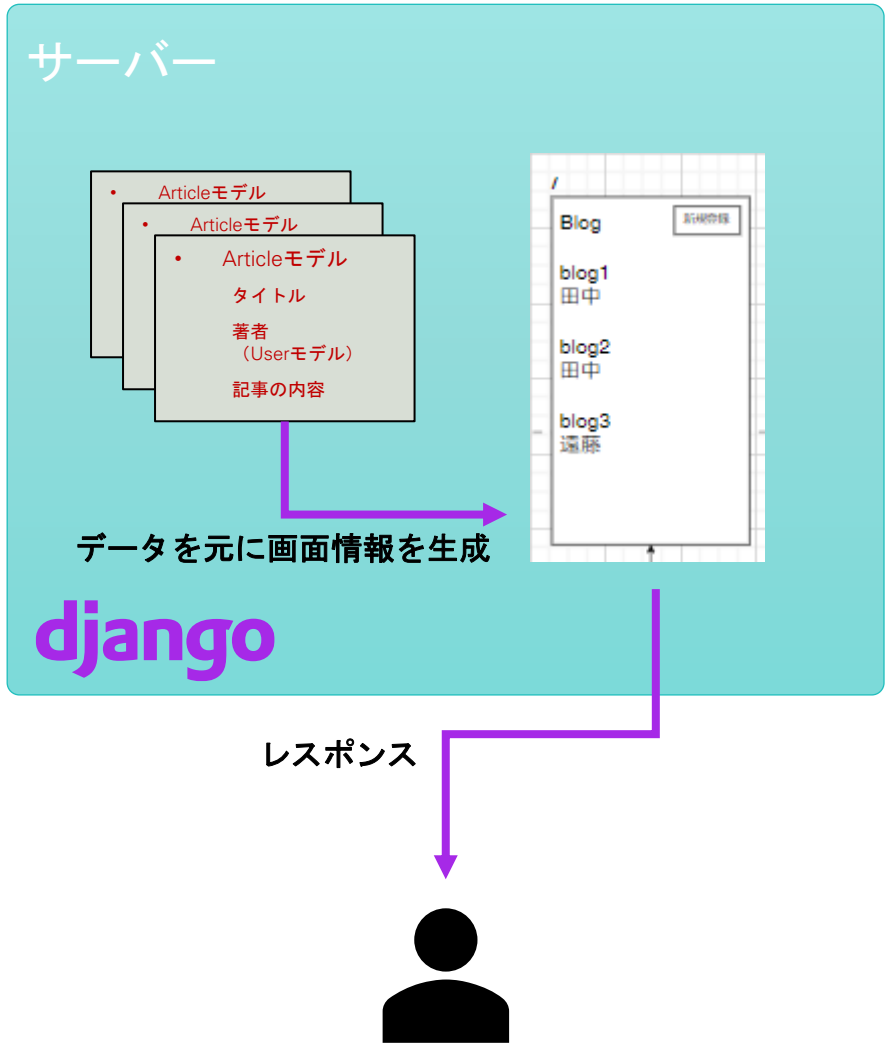
処理を定義する（モデルとデータベース）

- アクセスされたURLから情報を得ることができました。
- データベース（DB）とのアクセスには、SQLではなくORMを使います。
- DjangoではDBにアクセスする時、モデルを使ってアクセスします。
- 例えば、"/"にアクセスしたときはどのような処理が必要でしょうか？articlesテーブルからデータを全部取得する必要がありますね。



表示するページを作成する **html**

- 先ほどはデータベースからデータを取得するところまで進みました。
- しかし、それだけではユーザーにページの情報（HTML）の情報を返していないので、まだ何も見ることができません。
- 表示に必要な情報はtemplatesというフォルダの中にHTMLファイルを作って管理します。
- views.pyから、取得したデータをどのHTMLを使って表示するか指定してあげます。



フォームの設定

- 情報を取得して、表示することはできました。
- しかし、モデルの情報に合わせて、ユーザーから情報を受け取るにはどうしたらいいでしょう？
- フォームを通じて、ユーザーは文字を入力したり、選択肢から選ぶことができます。本来、表示するフォームは別途作る必要があります。
- しかしDjangoではモデルを使ってフォームを自動生成することができます。