テンプレートとは？

Webページ(HTML)の中に変数などを埋め込んだもの。

Djangoはテンプレートを読み込むと、そこにある変数などに値を埋め込んでページを完成させたものをクライアントに送る。

Djangoを使うことで、表示するHTMLの内容をあれこれ操作できるようにすることができる。

テンプレート

|  |
| --- |
| **PAGE**  Welcome to  Today’s TODO : |

Django

　　　　　実際の表示

|  |
| --- |
| **INDEX PAGE**  Welcome to Taro  Today’s TODO :  XXに電話する。 |

Djangoに組み込まれているシステム(テンプレート)を利用して動かすようするためには、Djangoにどういったアプリケーションを動かすのかを登録しなければなりません。この設定を行っているのが、プロジェクトの「settings.py」というファイルですsetting.pyのINSTALLED\_APPにアプリケーションを登録していないと、Djangoのテンプレート機能がそのアプリケーション内にある「templates」フォルダを検索してくれません。

helloアプリケーションを作成したので、登録が必要となります。登録の方法はsetting.pyのINSTALLED\_APPという変数の配列にhelloアプリケーションを追加するだけです。

INSTALLED\_APPS = [

　　　‘django.contrib.admin’,

　　　‘django.contrib.auth’,

　　　‘django.contrib.contenttypes’,

　　　‘django.contrib.sessions’,

　　　‘django.contrib.messages’,

　　　‘django.contrib.staticfiles’,

　　　‘hello’,

]

エディターにテンプレートを追加しよう！

helloアプリケーションを先ほどコードで登録しました。しかし、実際にファイルを作成していないため機能しません。①で作成したhelloアプリケーションの中にtemplatesファイルを作成し、更にhelloフォルダを作成します。これでテンプレートを作成することができました。templateの中にhelloフォルダを作成する理由は他のアプリショーンとごっちゃになる可能性があるからです。

＊図にしますと、下記のディレクトリーになります。

django\_apps

↓

　hello

　↓

templates

　　↓

　　　hello

**HTMLでテンプレートを表示させてみよう！**

先ほどのtemplateフォルダの中のアプリケーション名にindex.htmlを作成します。

index.htmlの中身

<!doctype html>

<html lang=”ja”>

<head>

　　　<meta charset=”utf-8”>

　　　<title>hello</title>

</head>

<body>

　　　<h1>hello</h1>

　　　<p>This is sample page.</p>

</body>

</html>

次にURLのパスをlocalhost:8000/hello/にするため、helloフォルダのurl.pyを開き、urlpatternsのパスの設定を以下のようにします。

urlpatterns = [

　　　path(’’, views.index, name=’index’),

]

最後にhelloフォルダのviews.pyを開きindex関数のスクリプトを下記のようにします。

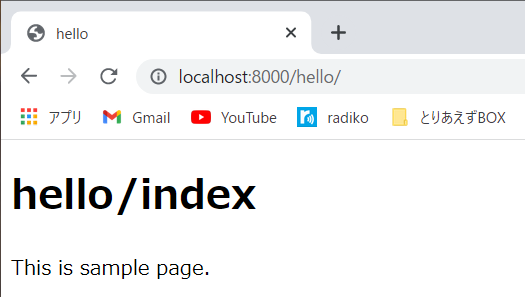
from django.shortcuts import render

from django.http import HttpResponse

Def index(request):

　　　return render(request,　’hello/index.html’)

runserverを起動させ、webブラウザからhttp://localhost:8000/hello/をアクセスしますと、下のように表示されます。



render関数とは？

templateタグを利用することなしでtemplateタグと同様にhtml文をブラウザに表示させることができる関数です。

**テンプレートに表示させることはできましたが、HTMLで表示させているだけです。このテンプレートを回しものにして表示させるためにテンプレートに値を渡してみよう。**

helloフォルダの中のindex.htmlの書き換える部分

titleタグの中を{{title}}と書き換える(<title>{{title}}<title>)

h1も同じく{{title}}と書き換える

pタグの中に{{msg}}と書き換える

ここでは、2つの変数を埋め込んでいます。{{title}}と{{msg}}です。テンプレートでは、このように{{変数名}}という形で変数を埋め込むことができるのです。{{}}という記号は、様々な値に埋める込むことができるので、変数だけに限りません。

次にindex関数で変数paramsに値を用意して、タイトルとメッセージとしてWebページに表示させるためにhelloフォルダのviews.pyのindex関数を下記のように変更します。

def index(request):

　　　params = {

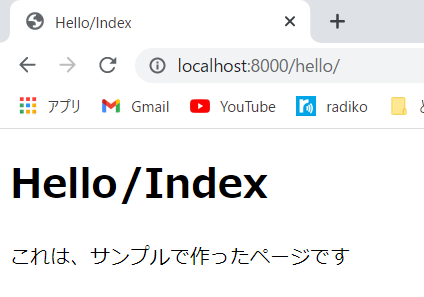
　　　　　　’title’:Hello/Index’,

　　　　　　’msg’:’これは、サンプルで作ったページです。’,

　　　}

　　　return render(request,　’hello/index.html’,params)

コードを打ったらもう一度<http://localhost:8000/hello/>にアクセスしてみよう



こんな具合に、「値を辞書にまとめる」「render時に第３引数で辞書を渡す」「テンプレート側で{{}}で値を埋め込む」という３つの作業で、ビュー関数側からテンプレート側に値を受け渡すことができるようになります。

**複数ページの移動**

Index.htmlのbodyの中にpタグをもう一つ作成します。その中に<a href=”{% url goto %}”>{{goto}}</a>を加え、helloフォルダのviews.pyを書き換えましょう。

from django.shortcuts import render

from django.http import HttpResponse

def index(request):

　　　params={

　　　　　　’title’:’Hello/Index’,

　　　　　　’msg’:’これは、サンプルで作ったページです。’,

　　　　　　’goto’:’next’,

　　　}

　　　return render(request, ’hello/index.html’,params)

def next(request):

　　　params = {

　　　　　　’title’:’Hello/Next’,

　　　　　　’msg’:’これはもう１つのページです’,

　　　　　　’goto’:’index’,

　　　}

　　　return render(request, ’hello/index.html’,params)

def next(request):

　　　params = {

　　　　　　’title’:’Hello/Next’,

　　　　　　’msg’:’これは、もう1つのページです。’,

　　　　　　’goto’:’index’,

　　　}

　　　return rendera(request, ’hello/index.html’,params)

次にhelloフォルダの中のurls.pyのurlpatterns文を以下のように修正します。

urlpatterns = [

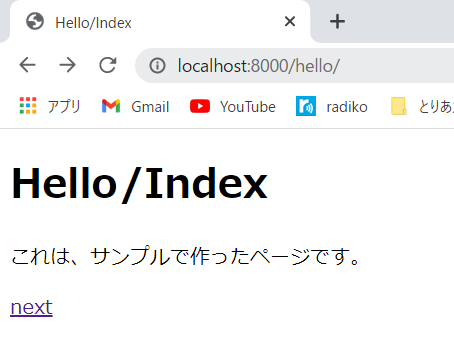
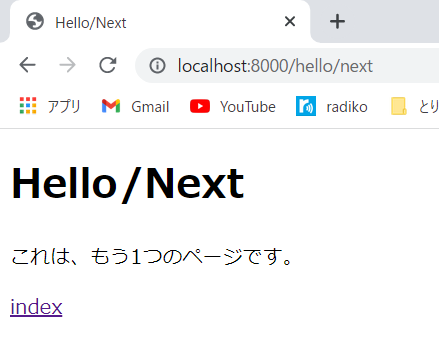
　　　path(’’,views.index,name=’index’),

　　　path(’next’,views.next,name=’next’),

]

コードが書けたら、runserverで実行し、<http://localhost:8000/hello/>にアクセスしてみよう

最初のページの「next」をクリックするとnextページに、「index」をクリックするとindexページに移動します。

ここでは、views.pyに２つの関数を用意して、それぞれindex.htmlに値を渡しています。

indexとnext関数では、それぞれgotoに’index’,’next’値を設定しています。

{% %}という記号は「テンプレートタグ」で、ここでは「url」というテンプレートタグです。

{% url 名前 %}はurlpatternsで用意した名前を指定しています。

path(’’,views.index,name=’index’)

path(’next’,views.next,name=’next’)