

Djangoチュートリアル編

Django導入まで

pip導入

```
$ sudo apt-get update -y  
$ sudo apt install -y python3-pip
```

virtualenv導入

```
$ sudo pip3 install virtualenv
```

virtualenv環境作成

virtualenvコマンドで仮想実行環境を作成し、activateコマンドで有効化します。

ここではユーザーのホームディレクトリの下へ virtualenv というディレクトリで作成します。

```
$ virtualenv ~/virtualenv  
$ source ~/virtualenv/bin/activate
```

Pythonが切り替わったことを確認します。

```
$ which python  
/path/to/virtualenv/bin/python  
$ python -V  
Python 3.5.2
```

無効化する場合は deactivate コマンドを実行します。

```
$ deactivate
```

Django導入

Djangoのパッケージをインストールします。

virtualenvが有効な状態でコマンドを実行してください。

```
$ pip3 install Django==1.10.3
```

pip freezeコマンドで表示されるパッケージの一覧にDjangoがあることを確認します。

```
$ pip3 freeze  
Django==1.10.3
```

Djangoアプリケーションの作成

Djangoチュートリアルに沿って、ECサイトアプリケーションを作成しましょう。

<https://docs.djangoproject.com/ja/1.10/intro/tutorial01/>

プロジェクト作成

Djangoプロジェクトを作成します。今回は ecommerce_website という名前で作成します。(公式チュートリアルでは mysite となっています)

```
$ cd ~  
$ django-admin startproject ecommerce_website
```

アプリケーション作成

プロジェクト上にアプリケーションを作成します。今回は ecommerce という名前で作成します。(公式チュートリアルでは polls となっています)

```
$ cd ecommerce_website  
$ ./manage.py startapp ecommerce
```

作成したアプリケーションをsettingsモジュールに追加します。

ecommerce_website/settings.py

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

アプリケーションでHello world

Hello worldを実装します。Pythonは空白や改行が意味を持つので、余分な文字が入らないよう注意して実装してください。

ecommerce_website/urls.py

```
from django.conf.urls import url, include # ←include を追加  
from django.contrib import admin  
  
urlpatterns = [  
    url(r'^ecommerce/', include('ecommerce.urls')), # ←追加  
    url(r'^admin/', admin.site.urls),  
]
```

ecommerce/urls.py (新規作成)

```
from django.conf.urls import url  
from . import views  
urlpatterns = [  
    url(r'^$', views.index, name='index'),  
]
```

ecommerce/views.py

```
from django.http import HttpResponse  
  
def index(request):  
    return HttpResponse("Hello, world. You're at the ecommerce index.")
```

確認

開発用webサーバを起動します。

```
$ python manage.py runserver 0.0.0.0:8000
```

別のシェルから確認してみましょう。

```
$ wget -q -O - 'http://localhost:8000/ecommerce/'  
Hello, world. You're at the ecommerce index.
```

template

template機能を使ってみましょう。

<https://docs.djangoproject.com/ja/1.10/intro/tutorial03/> を参考にしてください。

はじめにアプリケーションディレクトリの中に templates ディレクトリを作成します。

```
$ mkdir ecommerce/templates
```

views.indexをテンプレートを使うように変更します。

ecommerce/views.py

```
from django.shortcuts import render  
  
def index(request):  
    context = {  
        'foo': 'bar',  
    }  
    return render(request, 'index.html', context)
```

ecommerce/templates/index.html (新規作成)

```
fooの値 {{foo}}
```

動作させてみましょう。

viewから渡した変数が反映されています。

```
$ wget -q -O - 'http://localhost:8000/ecommerce/'  
fooの値 bar
```

テンプレートからはviewでアサインした変数のほかに、リクエストの内容も参照できます。

たとえばURLパラメータは request.GET から参照できます。

下のテンプレート記述はURLのbaz=パラメータの値を表示する例です。

ecommerce/templates/index.html

```
{{request.GET.baz}}
```