Python での DB の扱い方（SQLAlchemy編）

postgresql の導入 - Pythonでの接続テストまで

参考1 : sqlalchemy の使い方（http://symfoware.blog68.fc2.com/blog-entry-1373.html）
参考2 : Homebrewを使ったPostgreSQLのインストール
（http://qiita.com/tstomoki/items/0f1a930bd42a8e1fdaac）

brew で postgresql を導入

shell
```
# install
$ brew install postgresql
# DBの初期化
$ initdb /usr/local/var/postgres -E utf8
```

postgresql の起動

ひとつのシェルで postgres を起動しておく

shell-A
```
$ postgres -D /usr/local/var/postgres
```

もうひとつのシェルを立てて，以下のコマンドでデータベース一覧が取得できるか確認する.
一覧が取得できれば，インストール成功かつDBは正常に動作している.

shell-B
```
$ psql -l

    Name    |   Owner   | Encoding |   Collate   |    Ctype    |
Access privileges
------------+-----------+----------+-------------+-------------+------
-------------------
 postgres   | Yoshiharu | UTF8     | ja_JP.UTF-8 | ja_JP.UTF-8 |
 template0  | Yoshiharu | UTF8     | ja_JP.UTF-8 | ja_JP.UTF-8 |
=c/Yoshiharu           +
            |           |          |             |             |
Yoshiharu=CTc/Yoshiharu
 template1  | Yoshiharu | UTF8     | ja_JP.UTF-8 | ja_JP.UTF-8 |
```

```
=c/Yoshiharu            +
            |           |           |             |             |
Yoshiharu=CTc/Yoshiharu
(3 rows)
```

postgresql へのログイン & DB作成

```shell
# login
~$ psql -U Yoshiharu -d postgres

# login後は shell の表記が "postgres=#" になる
# test 用の DBを作成する
postgres=# create database pythontest;
# 確認
postgres=# \l
```

python での postgresql の操作

作業用ディレクトリとpython仮想環境の作成

```shell
$ mkdir python_postgre
$ cd python_postgre
# 仮想環境の作成
$ pyenv virtualenv 2.7.11 python_postgre
New python executable ...
Installing setuptools, pip, wheel...done.
# 仮想環境をローカルに適用する
$ pyenv local 2.7.11/envs/python_postgre
```

SQL操作用の package のインストール

```
$ pip install psycopg2 Flask-SQLAlchemy Flask-Migrate
```

インストールできているかを確認

```shell
$ python
>> import sqlalchemy
>> print(sqlalchemy.__version__)
1.1.11
```

さきほど作成したテスト用DB(pythontest)にテーブルを追加してみる
だいたいの流れは以下のとおり.

1. sqlalchemy を import する
2. DB 接続用の url を生成する．最小限だと "postgresql:///" の url で接続できる
3. url を引数に渡して，engine を生成する．engine は DB とのインターフェースの役割をするオブジェクト

今回は，localhost (=server_name) の pythontest (=db_name) なので，url は以下のソースコードの通りになる．

sample
```
import sqlalchemy
url = 'postgresql://localhost/pythontest'
engine = sqlalchemy.create_engine(url, echo=True)
```

以下のソースコードを実行して，postgresql に User table ができているかを確かめる．

connectdb.py
```
import sqlalchemy
import sqlalchemy.ext.declarative

Base = sqlalchemy.ext.declarative.declarative_base()

class User(Base):
    __tablename__ = 'user'
    id = sqlalchemy.Column(sqlalchemy.Integer,
primary_key=True)
    name = sqlalchemy.Column(sqlalchemy.String)

url = 'postgresql://localhost/pythontest'
engine = sqlalchemy.create_engine(url, echo=True)

Base.metadata.create_all(engine)
```

postgresql で pythontest に接続し，"\dt"でtableを確認する

shell
```
$ psql -U Yoshiharu -d pythontest
psql (9.6.3)
Type "help" for help.

pythontest=# \dt
         List of relations
 Schema | Name | Type  |   Owner
--------+------+-------+-----------
 public | user | table | Yoshiharu
(1 row)
```

Flask 上で postgreSQL を操作する

参考 : Flask by Example - Setting Up Postgres, SQLAlchemy, and Alembic (https://realpython.com/blog/python/flask-by-example-part-2-postgres-sqlalchemy-and-alembic/)

app.py で, sqlalchemy を import し DB への接続を行う

app.py

```
from flask import Flask
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] =
'postgresql://localhost/pythontest'
db = SQLAlchemy(app)

from models import User
```

models.py で User model を class として定義する

models.py

```
from app import db
from flask_sqlalchemy import SQLAlchemy

class User(db.Model):
    __tablename__ = 'users'

    id = db.Column(db.Integer, primary_key=True)
    username = db.Column(db.String(80), unique=True)

    def __init__(self, username):
        self.username = username

    def __repr__(self):
        return '<User %r>' % self.username
```

manage.py で migration 処理を設定する.

manage.py

```
from flask.ext.script import Manager
from flask.ext.migrate import Migrate, MigrateCommand

from app import app, db

app.config['SQLALCHEMY_DATABASE_URI'] =
'postgresql://localhost/pythontest'
```

```
migrate = Migrate(app, db)
manager = Manager(app)

manager.add_command('db', MigrateCommand)

if __name__ == '__main__':
    manager.run()
```

以下の3行を実行し，DB の migrate を行う．

shell
```
$ python manage.py db init
$ python manage.py db migrate
$ python manage.py db update
```

DB の検証

shell
```
$ python
>>> from app import db
>>> from models import User
>>> admin = User('admin')
>>> db.session.add(admin)
>>> db.session.commit()
>>> User.query.all()
[<User u'admin'>]
```