

PythonとMySQLの連携

ゾープ・ジャパン株式会社 柴田 淳

What's Python

What's Python

- オープンソースのオブジェクト指向スクリプト言語
 - クラス,関数などもオブジェクトとして扱うことができる
 - モジュール
 - 例外処理
 - オープンソース
 - Unicode文字列のサポート
- 最新バージョン 2.3.4
- 日本Pythonユーザ会(PyJUG)
 - <http://www.python.jp/>
- 日本語ドキュメント
 - http://www.python.jp/Zope/links/python_documents

What's Python

- 言語仕様がシンプルかつエレガント
 - 習得が容易
 - 可読性が高い
 - ブロックの範囲をインデントで表現
 - コーディングスタイルが安定する
 - ブロックの範囲を判別しやすい
- 拡張性が高い
 - オブジェクトの挙動の多くが開発者に対して「開かれている」
 - コンストラクタ/ディストラクタやオペレーターだけでなく、リスト、アトリビュートアクセス用のメソッドなどをオーバーライドし、挙動を変更できる。
 - 動的な型チェック - オブジェクトの振る舞いを動的に変更できる。
 - C,C++による拡張/連携が容易

What's Python

- 対話モード(インタラクティブモード)
 - テストコードなどを手軽に実行可能
- Jython
- マルチプラットフォーム
- 単機能スクリプトからサーバーまで
 - Mailman,Zope,BitTorrent,GMailFS...

DBとの接続

- 多くのRDBZMSと接続可能
 - MySQL,PostgreSQL,Firebird,Oracle,DB2,etc...
- PythonのC拡張モジュールを使い対応
 - RDBMSのコネクションやデータ構造などをPythonのオブジェクトにマッピングする拡張モジュール

DBAPI(2.0)

- RDBMS接続の汎用性を高めるための仕様。
 - http://www.python.jp/pub/doc_jp/contrib/peps/pep-0249.txt
- 主要なRDBMS接続モジュールがDBAPIに準拠

MySQLdb

- Python - MySQLを接続するためのモジュール
 - <http://sourceforge.net/projects/mysql-python>
 - 4.1と合わせて使うならmysql-python-testを使う
 - 3.x系ならmysql-python(1.0.0)でOK
- ビルド方法
 - \$python setup.py build
 - #python setup.py install

デモ

- Blogのエントリを登録したblogpingというテーブルを利用

```
#モジュールをimport
import MySQLdb
#コネクションオブジェクトを取得
con = MySQLdb.connect(db='test')
#カーソルを取得
curs = con.cursor()
#SQLを実行
r = curs.execute("select * from blogping")
#一行取得
row = curs.fetchone()
for row in curs.fetchall():
    print row[0],row[1],row[2],row[3]
```

テンプレートによるSQLレンダリング

- Pythonのテンプレート機能を使ってSQLをレンダリング可能

```
curs.query("""INSERT INTO blogping(title,url)
VALUES('%s','%s')""" % (title,url))
```

SQLObject

Python Relational-Object Mapper

SQLObject

- ORM(Object-relational Mapper)
- RDBMSのテーブル、データをPythonのクラスにマッピングしてくれる
- 最新バージョンは0.6
- MySQL, Postgres, SQLite, Firebirdなどに対応
- <http://sqlobject.org/>
- c.f.
- HigherLevelDatabaseProgramming(Python製ORMのリスト)
 - <http://www.python.org/cgi-bin/moinmoin/HigherLevelDatabaseProgramming>

デモ

```
from sqlobject import *
# -*- coding: utf-8 -*-

from setup import *
__connection__ = conn

## 要素数の少ないPersonというクラス
class Person(SQLObject):
    __connection__ = conn

    firstName = StringCol()
    middleInitial = StringCol(length=1, default=None)
    lastName = StringCol()

## 次のようなテーブルが作成されます (for MySQL)
"""
CREATE TABLE person (
  id INT PRIMARY KEY AUTO_INCREMENT,
  first_name TEXT,
  middle_initial CHAR(1),
  last_name TEXT
);
"""
```

デモ(続き)

```
def reset():
    #既存テーブルをdrop, 新規にテーブルを作成
    Person.dropTable(ifExists=True)
    Person.createTable()

    #リセットする
    reset()

## Personインスタンスを作成, 表示
p = Person(firstName="John", lastName="Doe")
print p
#>> <Person 1 firstName='John' middleInitial=None lastName='Doe'>

#要素を表示
print p.firstName
#>> 'John'

#ミドルネームを設定 (UPDATE), 表示
p.middleInitial = 'Q'
print p.middleInitial
#>> 'Q'
```

デモ(さらに続き)

```
#新規にp2インスタンスをDBから取得
p2 = Person.get(1)
print p2
#>> <Person 1 firstName='John' middleInitial='Q' lastName='Doe'>

#pとp2の要素は同じかどうかを調べる
print p is p2
#>> True

## Snippet "simpleaddress-person1-use-set"
p.set(firstName='Bob', lastName='Dole')
## end snippet
```

ZopeとMySQLの連携

Zopeの紹介

- Webアプリケーションサーバ
- 多くの部分がPythonで記述されている
- HTTP,FTP, WebDAVサーバを内蔵
- オブジェクトデータベースを内蔵
- ZMI(Zope管理画面)
- ブラウザでZope内のオブジェクト(≠ファイル)を管理できる
- サーバでもあり, 開発環境でもある



二つのテンプレート言語

- DTML
 - 拡張タグ言語

```
<dtml-var standard_html_header>
<h2><dtml-var title_or_id> <dtml-var document_title></h2>
<p>
This is the <dtml-var document_id> Document
in the <dtml-var title_and_id> Folder.
</p>
<dtml-var standard_html_footer>
```

ZPT(Zope Page Template)

- HTMLのアトリビュート領域にロジックを記述
 - Dreamweaverなどのオーサリングツールとの親和性が高い

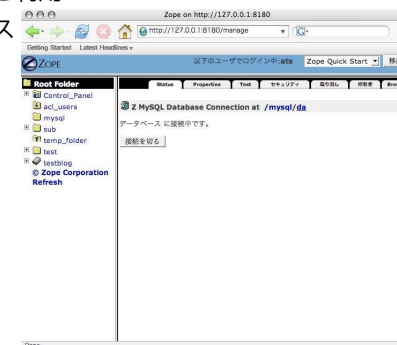
```
<html>
<head>
  <title tal:content="template/title">The title</title>
</head>
<body>

  <h2><span tal:replace="here/title_or_id">content title or id</span>
    <span tal:condition="template/title"
      tal:replace="template/title">
      optional template title</span></h2>

  This is Page Template <em tal:content="template/id">template id</em>.
</body>
</html>
```

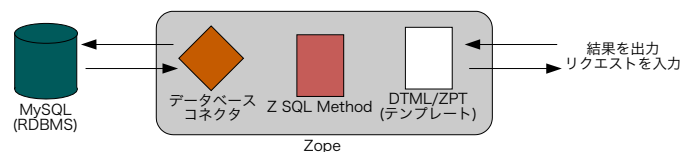
データベースコネクタ

- Zopeは多くのRDBMSと接続できる
 - MySQL,PostgreSQL,Firebird,Oracle,DB2,etc...
 - PythonのDB接続モジュールを利用
 - ZopeにDB接続のベースクラスが内蔵されている
 - ProductとしてDB接続機能を拡張可能
- コネクションプール



Z SQL Method

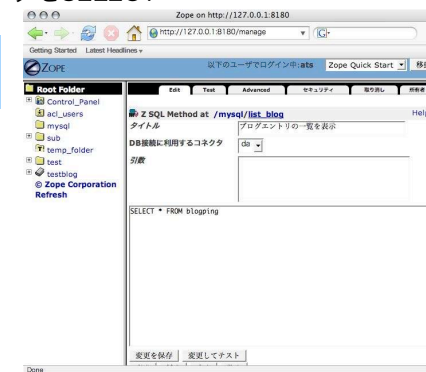
- Zope上でSQLを記述するためのオブジェクト
- SQLの動的なレンダリング
- DTMLを使い簡単なロジックを記述可能
- キャッシュ(リクエストキャッシュ)



デモ

- Z SQM Method(list_blog)
 - テーブルbloggingからデータをSELECT

```
SELECT * FROM blogging
```



デモ(続き)

- DTML Method(list_html) - エントリのリストを表示

```
<dtml-var standard_html_header>
<h2><dtml-var title_or_id> <dtml-var document_title></h2>
<p>
<table border="1">
<tr>
<th>タイトル</th>
<th>更新日</th>
<th>User Agent</th>
</tr>

<dtml-in list_blog>
<tr>
<td><a href="<dtml-var url>"><dtml-var title></a></td>
<td><dtml-var entrydate></td>
<td><dtml-var agent></td>
</tr>
</dtml-in>
</table>
</p>
<dtml-var standard_html_footer>
```

まとめ

- 「おはよう」から「おやすみ」まで暮らしを見つめるPython
- ローレベルAPIを使ってゴリゴリ書きたい人
- ORMを使って高級なことをしたいひと
- バズーカ砲(Zope)で手早く効率的に仕事を終えたい人
- Pythonは「癒し」を与える言語である

ありがとうございました