目次へ

shelveモジュール

仕事でコンピュータ使う用事といえば何か、と聞いたら、「データベース」という答えが相当上位に来るでしょう。データベースといっても色々な種類がありますが、まずはおおむね、「ほしいデータが取り出せる。探しやすいように格納もしておける」という条件を満たせば、そいつをデータベースと呼んでも差し支えないんじゃないでしょうか。

で、データベースの割と単純な姿を実現してくれるshelveモジュールを今回紹介することにします。

今まで、スクリプトの中で色々と処理したデータは、ほとんどの場合、pythonを終了したらその場でなくなっちゃうようなものばかりでした。データを処理した結果は、メモリの中で処理されて、最後にプロンプト上に表示されるなどして終わりでした。テキストファイルやHTMLファイルを書き出したりすることはしましたが、それ自身はデータベースとは呼びがたいものです。単なる一覧表みたいなものでしたしね。

データベースと聞いて、「ああ、SQLとかを使うアレだな」と思いつく方もいるでしょう。そいつは一般に「リレーショナルデータベース」というデータベースの一種で主に使われるものです。今回はもうちょっと単純なデータベースです。とはいえ単なるテキストファイルよりは面白く使えるようなものですよ。(SQLを使うようなやつは、この後でやりましょう。)

で、shelveというのがどんなデータベースかというと、pythonの「辞書」と同じ使い勝手を持つデータベースです。「辞書」は色々な使いかたがあって面白いものでした。「キー」とそれに対応する「値」を格納しておいて、キーをもとにしてその「値」を取り出せるというのが基本的な機能です。

ここまで説明したら、もう例に移ってしまうのがいいでしょう。下のスクリプトを実行してください。

```
import shelve
s = shelve.open('firstdb')
s['tyama'] = 'Yamamoto Tetsuya'
print s['tyama']
```

```
(実行結果)
'Yamamoto Tetsuya'
```

(この程度のスクリプトなら対話シェルで行ったほうが早いのですが、カレントディレクトリを調整するのがちょっと面倒だな、という程度の理由で、ここはあえてスクリプトを実行することにしています)

- カレントディレクトリって用語ははじめて出現するようですので、説明。コマンドプロンプト上で、「現在のフォルダ」を移動したりしてスクリプトの実行環境を整えたりしていますね。この「現在のフォルダ」と同じ意味です。対話シェルではこの「現在のフォルダ」がどこか確認しにくいので、どこにデータベースファイルが生成されるか、見失ってしまうとアレでして。

スクリプト内でやっていることは、shelveというモジュールを使うためにimportして、そこで用意されているshelve.open関数を実行してその結果を s に入れるという処理です。ここまでやった後は、sを単に辞書を使うときとまったく同じように扱っていることがわかるでしょう。tyamaというキーに実際の名前を入れて、直後にそれを呼び出してprintしています。説明するまでもない感じです。

このスクリプトが正常に実行できたら、同じフォルダ内に firstdb という名前のファイルができていることを確認してください。こいつがshelveデータベースのファイルです。 shelve.openの引数で、使うデータベースを指定したというわけです。すでにこのデータベースがあればそいつを使いますが、なかった場合はこんな風に勝手に新しく作ってくれますから、とても扱いが楽でよいです。

firstdbというファイルは、テキストファイルではありません。pythonのshelveモジュールでしか使えない、専用の形式で作られています。試しにテキストエディタで開いてみてもいいですが、ぐちゃぐちゃな中身に見えるでしょう。でも使えるから大丈夫。

さて、このデータベースが残っていれば、次にpythonを実行しなおしたときにもさっきの辞書を引き続き使えます。下のスクリプトがそれを確かめる例。

```
import shelve
s = shelve.open('firstdb')
print s['tyama']
```

```
(実行結果)
'Yamamoto Tetsuya'
```

出ましたか。

もしキーに対応する値がないときは、辞書を使うときと同じように KeyError という例外が発生します。try, catch でうまく扱わない限り、pythonが異常終了してしまいます。で、has_key や setdefault などといったメソッドも辞書のときと同じように使えますから、そいつを使って色々処理するということもできます。キーの一覧を keys を使って取得もできます。とにかく、辞書と同じです。ファイルに残る、という特徴以外は、感動的に辞書と同じです。

で、辞書と同じように使えるんだから、「値」の部分には単なる文字列だけじゃなくてリストや別の辞書が入っていてもいい。

```
import shelve
s = shelve.open('firstdb')
s['tyama'] = {'name': 'Yamamoto Tetsuya', 'age': 37, 'job': 'Yami Programmer'}
```

だからこんな風に、'tyama' というキーと、このYamamoto氏の個人情報をさらに辞書（こっちはshelveじゃない単なる辞書）に入れて格納することもできます。この場合、この情報を参照するためのスクリプトは下のような感じになりますね。

```
import shelve
s = shelve.open('firstdb')
p = s['tyama']
print p['job']
```

'tyama' というキーでまずはshelveから値を取り出します。その値自体も辞書なので、一旦 p に格納しておいてからさらにその中身を調べています。

実行結果は省略しますが、コードネーム 'tyama' であらわされるYamamoto氏の職業は闇プログラマーであることが、データベースを検索することで判明してしまいました。マンガとか映画でよくあるシーンですね。ハードボイルドな感じでかっこいいですね。さらにこのデータベースへのアクセスがパスワードとかで守られていたら完璧ですね。まあ、闇プログラマーってのがなんだかは知りませんけど。

shelveは面白いので、ちょっと練習問題もやってみようかと思います。

まずこのファイルを手もとにダウンロードしてください。

100poems

こいつはあらかじめ作っておいた、百人一首のshelveデータベースです。Windows用のpython2.6と、それに標準添付のshelveモジュールで作りました。他のOS用だったりすると、うまく動かないかもしれません。そのときはスイマセン。shelveモジュールはこういう微妙な依存の問題があるようなんです。

- 百人一首の元データは、http://ja.wikisource.org/wiki/[http://ja.wikisource.org/wiki/]小倉百人一首 からいただきました。

こいつがちゃんと使えそうか、下のような感じのスクリプトでテストしてみてください。スクリプトと同じ場所にshelveファイルを置いて実行します。

```
import shelve
s = shelve.open('100poems')
print len(s.keys())
```

473っていう数字が無事に表示されれば、ちゃんと使えているとみてよいでしょう。おかしなエラーメッセージ(bsddbがなんちゃら…)が出たら、今日のところはあきらめて、残りは適当にこの記事を読むだけにしてください。

で、このshelveファイルは、大きく二種類のキーと、それに対応する値を持っています。

ひとつは、歌番号に対応する歌。キーは、'poem001' から 'poem100' という文字列です。poemっていうのが適切な語なのかは知りませんが、まあそんなルールで作りました。数字部分は必ず三ケタです。数字の前にゼロをつけて決まった文字数にそろえる方法、ありましたね。思い出せるでしょうか。

ともかく、たとえば 'poem001' に対応する値は下の通り。

```
秋の田の かりほの庵の 苫をあらみ 我が衣手は 露にぬれつつ（天智天皇）
```

テキストは、シフトJISエンコードされた文字列として入っています。

もうひとつのキーの種類は、'秋' とか '田' とかの漢字一文字でそれぞれ持っていて、この値は、「その漢字が入っている歌のリスト」です。あ、この漢字一文字も、シフトJISです。つまり正確には２文字分のキーなんですけど、まあ細かいところはこだわらない。シフトJIS(cp932)で普通にスクリプトを書けばいいってことです。

たとえば、'天' に対応する値は下の通り。

```
[1, 2, 7, 12, 15, 60]
```

ここでは、リストとして入っています。1とか2とかは数字で、歌番号に対応しています。リストが六つの数字でできていますから、全部で六首に「天」が入っていることが分かるという仕掛けです。詠み人に含まれるときもあります。

辞書(shelveファイル)のキー総数が473ってことは、100首のデータと、373種類の漢字について出現する句のリストがいっしょくたに入っているということです。

で、こいつを参照しながら下の課題を解いてもらうというのが、今回の練習問題です。

練習問題08

【練習問題:08】百人一首データベースを調べ、「春」という文字が(詠み人も含めて)含まれる句をすべて抜き出して表示せよ。そののちに「夏」「秋」「冬」「花」「鳥」「風」「月」についても同様に調べよ。「春夏秋冬花鳥風月」の中でもっとも多く含まれるのはどれか。

調べるまでもなくそんなのわかる、という方もいらっしゃるでしょうが、まあ、スクリプト書いてウラを取りましょうよ。一番多かった漢字は、全部出してから目視で調べればよいです。