目次へ

Web上のドキュメントをテキストファイルのように読み込む

今まで、データとなるテキストファイルを手元に置いて、こいつをまた別のスクリプトファイルを使って読み込むという練習を繰り返してもらいました。

今後もいろいろな操作のために似たことを繰り返していきますので、かなり基本的なワザと理解しておいてください。

基本のワザですので、こいつをちょっとひねって使うという応用ワザのための足場にもなっていきます。今回、ちょっと早い話題ではあるのですが、具体的にどんな応用ワザが存在するのだろう、ということを紹介してみたく、ここにこんな記事をはさむことにしました。

やることは、手元にあるテキストファイルでなく、まだWeb上にあるテキストファイルを扱うというワザです。（なんか、ワザ、ワザ、と連呼しすぎだな）

まず、下のスクリプトを確認してみましょう。「練習問題1」の回答例にあたるものです。

sample1.py

```
g=0
for s in open("leavetime.txt"):
  h=int(s[0:2])
  m=int(s[3:5])
  zangyo=h*60+m-1040
```

```
  g=zangyo+g
a=g/60
b=g%60
print a, "hours", b, "minutes"
```

データファイルは、あらかじめ手元にダウンロードして、スクリプトファイルの隣りにたぶん置いてある、leavetime.txt というテキストファイルです。必要なら、「練習問題1」のページから改めてダウンロードしておきましょう。 これはちゃんと動作して、残業時間の合計を最後に出力してくれます。実行結果は

```
10761 hours 31 minutes
```

になったでしょう。

次に下のスクリプトを見てみましょう。直前のとどこが変わっているかをまずは確認してみてください。

sample1_web.py

```
from urllib import urlopen
url = "http://giraffe.topaz.ne.jp/wiki/lib/exe/fetch.php/py:leavetime.txt"
g=0
for s in urlopen(url):
  h=int(s[0:2])
  m=int(s[3:5])
  zangyo=h*60+m-1040
  g=zangyo+g
a=g/60
b=g%60
print a, "hours", b, "minutes"
```

変わっている部分は3行分ですね。最初のimport、次にurl変数に文字列を入れている部分、最後に、今までファイルのopenだったものが、urlopenとかいうものにすりかわっている。

これが何を意味するかというと、今まではテキストファイルを開いていたのに対し、urlに入れたインターネット上のリンク(URL)から直接ファイルを取得しながら全く同じ処理をするということです。

ためしにこのスクリプトを手元にダウンロードして、普通に実行してみましょう。leavetime.txtはゴミ箱に捨ててもかまいません。それでもなお、同じ結果が出てきますね。

本質的には、openの代わりにurlopenを使った、というそれだけのことです。最初のimportは、urlopenを使うためにちょっと前もって宣言しておくというだけのことで、オマジナイみたいなものです。 url変数にあらかじめ文字列を入れておいたのは、一行を長くしすぎないためだけのことです。

「テキストファイルの読み込み」という方法について充分理解を深めておけば、こういった応用の機会が出てきたときも柔軟に対応できるようになりますよ。

さらに今のうちに言っておくと、今はたまたまWeb上にデータファイルが置いてあるだけですが、いつも普通に眺めているその他のWebページってのも、本質的にはただのテキストファイルと変わらないものなのですよ。HTMLという種類のテキストファイルです。

知っている人にはなんでもないことですが、もしそんなことを聞くのは初めてだという方がいたら、今使っているウェブブラウザで、「ページのソースを見る」とか「ソースを表示する」とかそんな機能がきっとあるはずですから、それを探して実行してみてください。下の例みたいなものがズラズラと表示されませんか。

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
 lang="en" dir="ltr">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>
    練習問題・残業時間    [kirinwiki]
  </title>
  <meta name="generator" content="DokuWiki Release 2009-12-25c &quot;Lemming&quot;" />
<meta name="robots" content="index,follow" />
... 下にずっと続く ...
```

これがHTMLと呼ばれる形式の「テキストファイル」で、これも今までと同じような方法でpythonに読み込ませて扱わせるということが可能です。可能というより、超カンタンです、というほうが適当ですね。（まあ、ファイルの中身を正確に解析しようと思えば結構大変ですが、ここでは「読み込む」だけなら、ね）

で、これは、なんらWebサーバーに気兼ねする必要のない、当たり前な仕事のひとつでもあります。Webサーバーにとっては、相手がウェブブラウザだろうとpythonのプログラムだろうと、要求されたとおりのドキュメントを返すだけが仕事で、別に何の違いもないわけですからね。もっとも、あんまり高い頻度でばんばんドキュメントを要求するのは、ウェブブラウザでリロードボタンを連打してサーバー側を混雑状態にしてやろうというのに似て、マナー違反になる可能性はありますから、そこだけ注意すればですが。

この回はオマケの読み物くらいに読み飛ばして結構ですが、ここで言っておこうかなと思ったのは、

- python（とかその他のプログラム言語）でウェブ上の情報を取得することはカンタン
- そしてそれ自体はは誰に気兼ねする必要もないもの

ってことです。

これって、googleとかyahooとかの「ウェブクローラ」っていう種類のものがやっていることと同じだったりもしますね。クローラってのは、ウェブサーバーから得られる全部の（またはできるだけ多くの）HTMLファイル等を自動的にかき集めていくという種類のプログラムです。そいつらがpython製かどうかは知りませんが、きっと部分的にはここで紹介したのと似たようなものが動いているんじゃないかな。

今勉強しようとしていることと、Webの世界の最先端の技術が案外近くにあるという実感を感じたりしませんか。しませんか、そうですか。