Python で日本語を使う

主に Python 2.x についての話題です。

ソースの書き方編

ソースコードの 1 行目か 2 行目でソースコード自体の文字コードを指定する。

```
# coding: utf-8
# coding: cp932
# coding: euc-jp
```

emacsen などとの互換性を考慮して以下のようにしてもよい。

```
# -*- coding: cp932 -*-
```

なお cp932 は Windows の機種依存文字を含む Shift_JIS のことである。それから以上のコメントは純粋にソースコードの文字コードを指定するだけであって、他の部分（入出力など）には一切の影響を及ぼさない。

複数の環境で動かすスクリプトの文字列は、次のように u を使って書く。

```
s = u"日本語"
```

文字コードの変換を行う場合は errors 引数を指定し、変換できない文字が見つかった場合の処理を明示する。

```
s.encode('utf-8')
s.encode('utf-8', errors='strict')
s.encode('utf-8', errors='ignore')
s.encode('utf-8', errors='replace')
```

外部とやり取りする時に文字コードを変換して、内部では unicode を使う。以下は utf-8 で入力、cp932 で出力する例。

```
src = open('src_filename', 'rb')
dst = open('dst_filename', 'wb')
for line in src:
    uline = unicode(line, 'utf-8')
    dst.write(uline.encode('cp932'))
```

代わりに codecs を使ってもいい。

```
import codecs
src = codecs.open('src_filename', 'rb', 'utf-8')
dst = codecs.open('dst_filename', 'wb', 'cp932')
for line in src:
    dst.write(line)
```

コンソールで化ける編

以下のコードをソースコードの先頭に記述する。

```
import sys
import codecs
enc = 'utf-8'
sys.stdin = codecs.getreader(enc)(sys.stdin)
sys.stdout = codecs.getwriter(enc)(sys.stdout)
sys.stderr = codecs.getwriter(enc)(sys.stdout)
```

LC_CTYPE をきちんと設定してもリダイレクトで化けるという場合はこれで直るはず。

Eclipse で化ける編

コンソールで化ける編に加えて、pydevd.py の execfile() の手前に次のコードを挿入する。

```
class EclipseStdout:
    def __init__(self, args):
        self._redirectTo = args
    def write(self, s):
        if type(s) is StringType:
            self._redirectTo.write(s)
        else:
            self._redirectTo.write(s.encode('cp932'))
stdout = EclipseStdout
execfile(file, globals, locals) #execute the script
```

コンソールでリストや辞書の文字列を日本語のまま表示する編

ASCII 以外の文字列はエスケープされた状態で表示される。以下のようにすればよい。

```
>>> [u'ふが', u'ホゲ']
[u'\u3075\u304c', u'\u30db\u30b2']
>>> print [u'ふが', u'ホゲ']
[u'\u3075\u304c', u'\u30db\u30b2']
>>> print repr([u'ふが', u'ホゲ']).decode('unicode-escape')
[u'ふが', u'ホゲ']
>>> ["はげはげ"]
['\x82\xcd\x82\xb0\x82\xcd\x82\xb0']
>>> print str(_).decode("string-escape")
['はげはげ']
>>> [u"はげはげ"]
[u'\u306f\u3052\u306f\u3052']
>>> print str(_).decode("unicode-escape")
[u'はげはげ']
```

Python 3.x では文字列が Python 2.x の unicode に相当するものになったので、そのまま表示される。

sys.setdefaultencoding はまず使わない

sys.setdefaultencoding は文字コードのこと考えていない古いソースコードのためにある設定項目なので、これから何か新しいものを書こうとするプログラマが設定する必要はありません。

Python は書けないが、海外で作られたアプリや MOD ユーティリティが UnicodeEncodeError/UnicodeDecodeError で動かないという方は、次のような設定で回避できる場合があります。

■ 方法１：sitecustomize.py を書き換える

・ Windows であれば C:\Python27\Lib\site-package\sitecustomize.py
・ Unix であれば /usr/lib/python2.7/site-package/sitecustomize.py

というファイルに以下の２行を追記してください。ただし、「********」は、お使いのファイルシステムの文字コード（Windows なら cp932、Unix ならたぶん utf-8）に置き換えてください。

```
import sys
sys.setdefaultencoding('********')
```

以上の設定は Python 全体で有効です。

■ 方法２：ソースコードを書き換える

該当ソースファイルの先頭に次の三行を加えます。

```
import sys
reload(sys)
sys.setdefaultencoding('********')
```

こちらは「書き加えたソースファイルの実行環境だけで有効」になります。

sys.setdefaultencoding を使わない場合の sys.argv の取り扱い (2.x 系)

以下は多言語のことが考慮されていない。これらの方法では文字の脱落などが生じてしまう。しかし解説を書く時間がない。

sys.setdefaultencoding を使わない場合、コマンドライン行の取得 (sys.argv[n]) を行うと、「wcs(UNICODE/wchar_t) ではなく、mbcs(char) として入力されて来る」ため、os.walk() や shutil.copyfile() の引数にそのまま渡すと問題になります。(ディレクトリ名やファイル名の末尾に「表」「申」が含まれる場合など。)

それらの引数に渡す前に wcs(UNICODE) への decode が必要です。

```
import os
import shutil
for pathname, dirnames, filenames in os.walk(sys.argv[1].decode('********'), topdown=True):
  for fn in filenames:
    print 'copying', fn
    shutil.copyfile(os.path.join(pathname, fn), os.path.join(pathname, fn + '.bak'))
```

ただし、「*********」は、お使いのファイルシステムの文字コード（Windows なら cp932、Unix ならたぶん utf-8）に置き換えてください。

結局、print が必要に応じて自動的に環境に合わせた encode で表示してくれるため、

```
print sys.argv[1]
```

とかで表示しても普通に表示されるのが混乱の原因なのですが、本来は、

```
print sys.argv[1].decode('********')
```

が正しい記述だと思っておいた方が、混乱が起きにくいと思います。

print で日本語出力

2.7 で端末に日本語出力する時は、 unicode で出力し、 setdefaultencoding は使わず、実行環境側で環境変数 PYTHONIOENCODING を設定しよう。

encode / decode が至る所に書かれてるようなコードは、保守性も悪くなるし、3.x への移植の際に手間になります。

お使いのファイルシステムの文字コード「********」を知る方法

sys.getdefaultencoding() の代わりに、下記の方法で文字コードを取得出来ます。

```
import sys, locale
enc = locale.getpreferredencoding()
print sys.argv[1].decode(enc)
```

参考

- UnicodeDecodeError の原因を知る
