Python2.x的编码问题

richardzhiming 662 11月19日 发布

推荐 0 推荐

收藏 2 收藏，145 浏览

好像写Python的人经常遇到这个问题,如果是一个新手,对网上的解答简直头都大了.如果又是Python2.x,简直了都(历史原因,不赘述)
但是,字符串编码是经常要面对的问题,不可不察.

> Unicode与各种编码格式

我的理解是:Unicode是一种`规则`,`法则`,`抽象的`,`飘在空中的`;而各种编码则是`工具`,`把Unicode字符捣鼓成我们想要的东西`.在Python中,Unicode充当着解决各种字符编码问题的`桥梁`

`数据(字符串)与Python程序无非两种关系: 数据流到Python程序中(输入),Python程序流出各种数据(输出).`

> 数据从外部(文件,网络等地方)-->Python程序时

先不急,用一个unicode接住它们,然后再进行之后的各种操作

```
content = unicode(originalContent, 'src_data_encoding')  #此处必须要
```

> 数据从Python-->外部时

Unicode字符是不能随便写的,要先把我们的Unicode编码成具体编码格式,然后再写出

```
content = unicodeContent.encode(encoding) #必须要清楚目的地能接受的编
```

> 其他需要知道的

1 Python有时会「自作主张地」转换我们的unicode(以ascii格式),简直`stupid`,ascii就那么几个字符,肯定经常抛UnicodeEncodeError啊

2 Python有时能猜到`目的地`的编码,猜到万幸,没有猜到就抛错误
3 使用Windows_中文版的人需要知道它的cmd控制台是gbk编码的
4 Linux等就好多了,我总是「如果不清楚该使用什么编码,那就是utf-8了」
5 文件头加一个`coding=utf-8`什么的/或者通过`sys模块`
6 `always be utf-8-no-bom`总是好的,所有IDE/editor都默认编码为`utf-8-no-bom`,可以省去许多麻烦(其中Windows的记事本少用,一不小心就忘记了`记事本会插入BOM`这一事实)
7 如果能用Python3就绝不使用Python2,3修补了字符串的许多坑(不只是unicode这一项)

> 举例

1 Windows_cmd控制台打印unicode

```
Windows8中文版-Python2.7
s = u'中国人民'
print s  #ok,看来Python知道应该使用什么编码,隐式转换了
print s.encode('gbk')  #ok,目的地能接受gbk
print s.encode('utf-8') #fail,看来万能的utf-8不好用啦,因为目的地不接受
```

2 有时候从网页获取来的数据可能是utf-8的,但是打印到Windows_cmd控制台出错了

```
content = unicode(contentFromHtml, 'utf-8')  #当然也有可能是其他编码
print content.encode('gbk')  #转换层控制台能接受的编码
```

> 总结

unicode是一座`桥梁`,连接这桥这头与那头