Python字符串开头的b"、u"、r"与中文乱码

Python字符串开头的b"、u"、r"与中文乱码

智能决策上手系列教程索引

先看几个常见的中文乱码：

```
s = u'More更多请关注我'

print('--encoded---')
print('【utf-8】', bytes(s, encoding='utf-8'))
print('【utf-16】', bytes(s, encoding='utf-16'))
print('【gbk】', bytes(s, encoding='gb2312'))
print('【unicode-escape】', bytes(s, encoding='unicode-escape'))
print('--decoded--')
print('【utf-8=>utf-8】', s.encode('utf-8').decode('utf-8'))
print('【utf-8=>utf-16】', s.encode('utf-8').decode('utf-16'))
print('【utf-8=>ISO-8859-1】', s.encode('utf-8').decode('ISO-8859-1'))
print('【gbk=>ISO-8859-1】', s.encode('gbk').decode('ISO-8859-1'))
```

运行得到：

image.png

仔细看一下，找到下面两个规律：

- 要么是上面encode编码后的¥...¥...¥...，要么是错误解码decode之后的乱码。
- 除了两个最后输出utf-16的，其他的都能正常显示英文More。

u'和b'

第一行我们用了u'More...'，这个u是可以省略的，在python3里面所有字符串默认都是utf-8编码解码的，这个u就是指示要用utf-8编码，所以可省略。

上面四个encoded输出的开头都有b'...'开头，因为代码里面是bytes(s,...)，bytes是字节，字符串可以用很多种编码方式变为字节，就像密码电报一样，同一个信息可以用不同的算法加密成各种不同的乱码，如果你不知道是什么编码的，你就读不出。

> *python2*中只有字节形式的字符串，没有*u'*开头的*utf-8*编码的，所以*python3*才发明了*b'*开头这种表示字节的方法，兼容旧版本。

字符串经过**特定的方式**编码**coding成**为**字**节，**然后再通**过**正确的方式也可以把字**节还**原**为**字符串。关**键**就是要知道原来的**编码**方式**

.encode('utf-8').decode('utf-8')这句当然看起来没毛病，所以也输出了正确的文字。
而后面s.encode('utf-8').decode('utf-16')这种就不正常了，编码encode方式和decode解码方式不一样，就导致了乱码。

解决乱码问题

首先要想办法知道或者实验出原来是什么编码的，你可以对照上面输出的代码进行猜测。

如果是很多斜杠的bytes，那么就直接尝试不同的decode：

```
s=b'More¥xe6¥x9b¥xb4¥xe5¥xa4¥x9a¥xe8¥xaf¥xb7¥xe5¥x85¥xb3¥xe6¥xb3¥xa8¥xe6¥x88¥x91'
print(s.decode('utf-8'))
```

如果是被错误解码的乱码，那么就尝试反向encode再decode就能还原：

```
#乱码来自s.encode('gbk').decode('ISO-8859-1')
s='More¸ü¶àÇë'Ø×¢ÎÒ'.encode('ISO-8859-1').decode('gbk')
print(s)
```

r' 强制不转义

先看下这个问题：

```
s1=r'.¥folder¥new.file'
s2=b'.¥folder¥new.file'
s3=u'.¥folder¥new.file'
print('r:',s1)
print('b:',s2)
print('u:',s3)
```

它的输出是这样的，你肯定猜不到：

image.png

为什么会是这样？只有r开头的正常。
我们都知道代码里面字符串要用引号包裹，那么字符串里面要是也有引号怎么办？换行怎么办？

编程语言会用特殊方法来表示这些特殊符号，叫做转义字符。比如¥n表示回车换行，¥f表示换页，¥"表示双引号，¥' 标示单引号...

那么真的遇到了¥n怎么办？就像上面那个情况，b'把f编码成了¥x0,而u'直接把¥f弄没了，¥n变成了回车...

为了避免这种情况，就有了r'强制不转义，优点是明显的，不会乱变，缺点也是有的，那就不能真的显示回车换行了啊。

一般在目录地址和正则表达式中我们常用r'避免混乱。

智能决策上手系列教程索引

每个人的智能决策新时代

如果您发现文章错误，请不吝留言指正；
如果您觉得有用，请点喜欢；
如果您觉得很有用，欢迎转载~

END