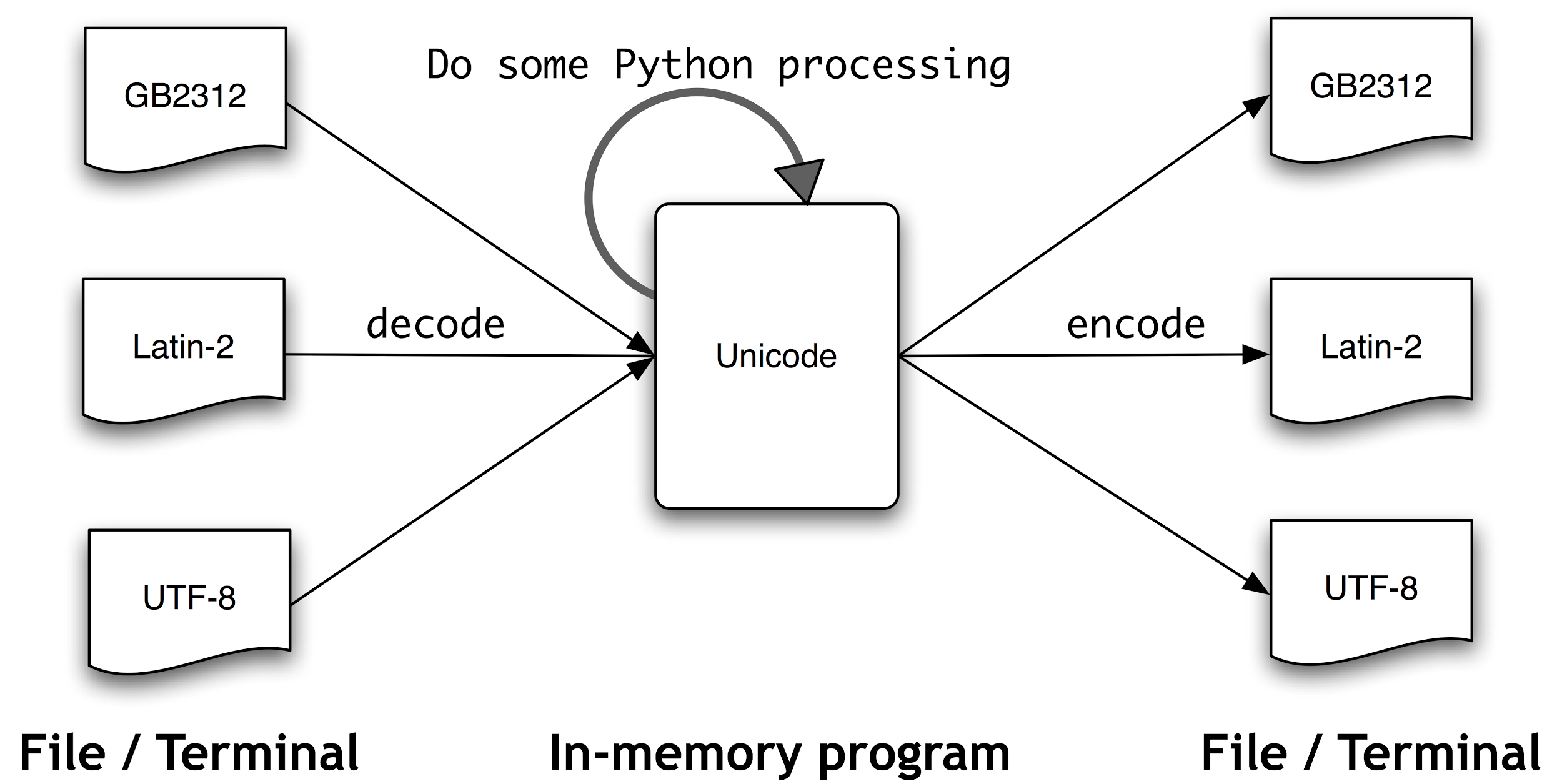
Unicode支持一百万种以上的字符。每个字符分配有一个编号，成为编码点。在Python里，编码点写作\uXXX的形式，其中XXXX是四位十六进制形式数。

在一段程序中，可以像普通字符串那样操作Unicode字符串。然而，当Unicode字符被存储在文件里或在终端上显示时，必须被编码为字节流。由于一些编码，如ASCII，Latin-2中的每个编码点都是用单字节，所以他们只需支持Unicode中的一个小子集就行了。其他编码，如UTF-8使用多个字节，可以表示全部的Unicode字符。

文件中的文本都是由特定编码的所以需要一些机制来讲文本翻译成Unicode——解码。想对的，要将Unicode写入一个文件或者终端，首先需要将Unicode装华为合适的编码———编码。



从Unicode的角度来看，字符是可以实现一个或多个字形的抽象的实体。只有字形可以在出现屏幕上或被打印在纸上。字体是字符到字形的映射。

codecs模块提供了将编码数据读入为Unicode字符串和将Unicode字符串以编码形式写出的含糊。codecs.open()函数有一个encoding参数来指定被读取或写入的文件编码。

|  |
| --- |
| path = nltk.data.find('corpora/unicode\_samples/polish-lat2.txt') |

import codecs

f = codecs.open(path,encoding='latin2')

for line in f:

line = line .strip()

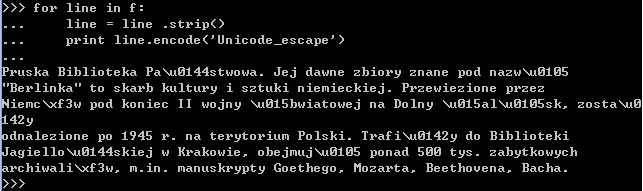
print line.encode('Unicode\_escape')

Python特定的编码unicode\_escape是一个虚拟的编码，它把所有的非ASCII字符转换成\uXXXX形式。编码点在ASCII 0~127范围以外但低于256的，使用两位数字的形式\xXX表示。

for line in f:

line = line .strip()

print line.encode('Unicode\_escape')



在python中，一个Unicode字符串常量可以通过在字符串常量前面加一个u来指定，如果想用unicode来表示a，可以写为u'\u0061’来表示。61是a对应的acsii 97的16进制表示。

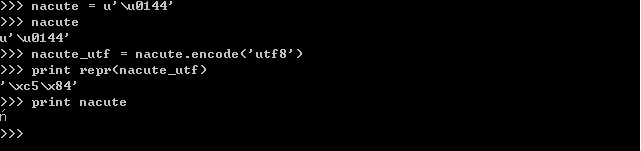


nacute = u'\u0144'

nacute

nacute\_utf = nacute.encode('utf8')

print repr(nacute\_utf)

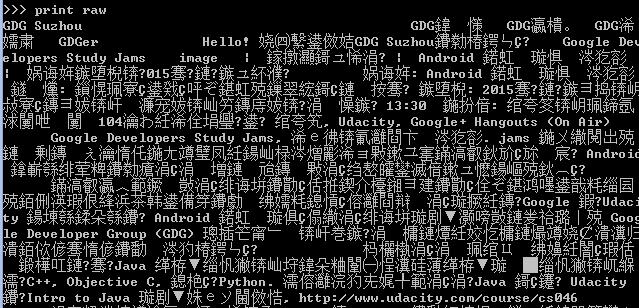


注意144在127和256之内。

dushu.hu去掉html标记，空格和换行之后就是如下壮观景象

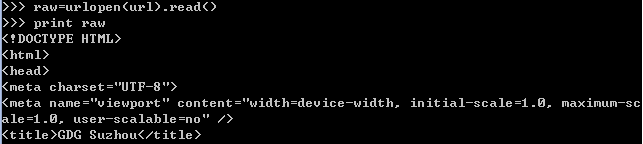


print一下。。。。



乱码。。。。

经过分析，我感觉这应该是unicode编码成为utf-8的结果，回头还原一下html标记，果然utf8

所以我对其进行了解码处理



我终于看到了亲切的unicode！！！！

来，print一下



文本提取出来了！！！

# <http://www.jb51.net/article/26543.htm> python 中文乱码问题深入分析

这篇讲得不错，建议看一下。