《源码探秘 CPython》19. 字符集和字符编码

原创 古明地觉 古明地觉的编程教室 2022-01-28 09:30

收录于合集
#CPython
97个 >

楔子

这一次我们分析一下Python的字符串，首先字符串是一个变长对象，因为不同长度的字符串所占的内存是不一样的；但同时字符串又是一个不可变对象，因为一旦创建就不可以再修改了。

而Python中的字符串是通过unicode来表示的，在底层对应的结构体是PyUnicodeObject。不过话说回来，为什么需要unicode呢？

首先计算机存储的基本单位是字节，由8个比特位组成，由于英文字母算上大小写只有52个，再加上若干字符，数量不会超过256个，因此一个字节完全可以表示。但是随着计算机的普及，越来越多的非英文字符出现，导致一个字节已经无法表示了。所以只能曲线救国，对于一个字节无法表示的字符，使用多个字节表示。

但是这样会出现两个问题：

- 因为每个国家都有自己的字符编码，所以不支持多国语言，例如中文的编码不可以包含日文，否则就会造成乱码；
- 没有统一标准，例如中文有GB2312、GBK、GB18030等多个标准；

到这里我们先不继续往下深入，我们先来理清楚一些概念。

字符集和字符编码

估计有很多小伙伴搞不清这两者的区别，我们先来解释一下所谓的字符集和字符编码是怎么一回事？

字符集：系统支持的所有字符组成的集合，像ASCII、GB2312、Big5、unicode都属于字符集。只不过不同的字符集所能容纳的字符个数不同，比如ASCII字符集中不包含中文，unicode则可以容纳世界上的所有字符；

字符编码：负责将每个字符转换成一个或多个计算机可以接受的具体数字，该数字可以理解为编号，因此字符编码维护了字符和编号之间的对应关系。而编码也分为多种，比如ascii、gbk、utf-8等等，字符编码不同，那么字符转换之后的编号也不同，当然能转化的字符种类也不同。比如ASCII这种字符编码，它就只能转换ASCII字符。

> 当然，ASCII比较特殊，它既是字符集、也是字符编码。并且不管采用什么编码，ASCII字符对应的编号永远是相同的。

将字符串中的每一个字符转成对应的编号，那么得到的就是**字节序列（bytes对象）**，因为计算机存储和网络通讯的基本单位都是字节，所以字符串必须以字节序列的形式进行存储或传输。

因此字符串和字节序列在某种程度上是很相似的，字符串按照指定的编码进行encode即可得到字节序列，也就是将每个字符都转成对应的编号；字节序列按照相同的编码decode即可得到字符串，也就是根据编号找到对应的字符。

比如我们写了一段文本，然后在存储的时候必须先进行编码，也就是将每一个字符都转成一个或多个系统可以接受的数字、即对应的编号之后，才可以进行存储。

```
s = "你好"
# 编码之后就是一串数字
print(s.encode("gbk"))  # b'\xc4\xe3\xba\xc3'
```

假设文本中只有**你好**二字，在存储的时候采用gbk进行编码，那么在读取的时候也必须使用gbk进行解码，否则的话就会无法解析而报错。因为字符编码不同，字符对应的编号也不同。

再比如每个国家都有自己的字符编码，你在日本的一台计算机上写好的文件拿到中国的计算机上打开，很有可能出现乱码。因为字符编码不同，字符和编号之间的对应关系也不同，采用不同的字符编码进行解析肯定会出问题。

但我们说，对于ASCII字符来说，由于不管采用哪一种编码，它们得到的编号都是固定的。所以编码对于ASCII字符来说，没有任何影响。

```
s = "abc"
print(s.encode("gbk"))  # b'abc'
print(s.encode("gbk").decode("utf-8"))  # abc

# 但如果是非ASCII字符，就不行了
try:
    s = "你好"
    s.encode("gbk").decode("utf-8")
except UnicodeError as e:
    # 报错了，无法解析
    print(e)
    # 'utf-8' codec can't decode byte 0xc4 in position 0: invalid contin
uation byte
```

这里我们再回忆一下bytes对象，我们创建的时候可以采用字面量的方式，比如**b"abc"**，但是 **b"憨"**却不可以。原因就是**憨**这个字符不是ASCII字符，那么采用不同的字符编码，其对应的编号是不同的，而这种方式Python又不知道我们使用哪一种编码，所以不允许这么做，而是需要通过**"憨".encode**的方式手动指定**字符编码**。

但是对于 ASCII 字符而言，不管采用哪一种字符编码，得到的编号都是一样的， 所以Python针对ASCII字符则允许这种做法，比如**b"abc"**。并且我们看到，对于汉字来说，在编码之后会对应多个编号，而每个编号占1字节，因此不同的字符所占的大小可能不同。

小结

以上就是字符集和字符编码，字符集就是字符组成的集合，不同字符集所能容纳的字符数量是有限的。字符编码是将字符转成对应的编号，比如将一个字符串中的所有字符都转成对应的编号之后，就得到了字节序列。

当然和字符集一样，字符编码能转换的字符种类也是有限的，像汉字我们可以使用 gbk 编码、utf-8 编码，但是不能使用 ascii 编码。

以上算是理清楚了一些概念，显然过于简单了，主要是为后面的内容做铺垫。那么下一篇，就来从Python的角度分析字符串的存储方式。

收录于合集 #CPython 97

‹ 上一篇
《源码探秘 CPython》20. Python是怎么存储字符串的?

下一篇 ›
《源码探秘 CPython》18. bytes 对象的缓存池

喜欢此内容的人还喜欢

python-字符串编码问题怎么破
一位代码

清华美女学姐编写的简明python编码规范，非常适合零基础入门

程序员森芋

CSS半透明属性介绍及代码实例

前端仿真