Python学习笔记--字符串操作string

Python中的字符串可以使用单引号，双引号，三引号表示。

单引号(')与双引号(")代表的意思相同，但要注意配对着用。并不分成单引号代表一个字符，双引号代表一个字符串，实际上Python中没有char。

三引号('''...''' or """...""")相比于前两者的的优势是，三引号内部可以自由使用单引号与双引号，可以用于描述一段对话中（因为里面很多引号），或者对一个函数的描述（因为可以指示一个多行的字符串，可以直接敲回车，而不用手写\n）。

转义符及'\'（表示当前行未结束，下一行继续）与C语言相同。

值得注意的是自然字符串与Unicode字符串。自然字符串可以将转义符取消掉，如：str=r'1 \n 2', print的结果为 “1 \n 2”，其中'\'与'n'表示两个字符，len(str)=6。 Unicode用于对非英文字母的处理，虽然str=u'1 \n 2', print的结果也为 “1 \n 2”，但'\n'还是表示换行符，只是并没有被直接转义，len(str)=5。

Python中，字符串也是不可以直接通过赋值改变的，想要对字符串进行某些修改时，要用到字符串相关的一些函数（注意：修改完后，原字符串仍未改变，需要将改变后的结果存到另一个字符串变量中）：

len(str): 获取str的长度。

enumerate(str): 返回str的index与value组成的元组。

str.append(value): 在字符串str最后添加数据value。与str.pop()结合可以实现stack与queue

str.pop(index): 从str中弹出位于index的数据，默认为最后一个，实现stack时，使用str.pop()表示栈顶数据弹出，实现queue时，使用str.pop(0)表示队列数据弹出。

str.upper(), str.lower(), str.swapcase(): 全部大写，全部小写，大小写转换。

str.find(subStr, start, end), str.rfind(subStr, start, end): 字符串搜索函数，分别表示从左往右与从右往左搜索，start与end是可选参数，默认值分别为0与-1，find()是从start到end, rfind()是从end到start. 当可以找到substr时，函数返回值为发现的第一个subStr的下标（不论是find()还是rfind()函数，字符串下标均是从左往右），如果未找到，则返回-1，不是0（这点，写if-else的时候需要注意）。

str.count(subStr): 也是搜索函数，找到str中的substr的数目。

str.replace(oldSub, newSub, maxReplaceTimes): 字符串替换函数，如果可以在str中找到oldSub，则将oldSub替换为newSub，否则不替换。maxReplaceTimes为可选参数，表示替换的次数，默认为替换所有可找到的oldSub。

str.strip(subStr): 从字符串开始与结尾处分别搜索，直到未遇到subStr中的任何单个字符，停止；如果遇到，则删除，并继续搜索。如str='ababcabab'; str.strip('abc') 结果为''。subStr也是可选参数，默认值为空白符（'\n', '\r', '\t', ' '）。类似的有lstrip()与rstrip()函数，分别表示从左往右与从右往左搜索。经常用于读文件中：

for line in file.readlines: # 或者 for line in file:

line\_new=line.strip() #由于readlines函数不去除每行结束时的换行符，需要手动去除。将line.strip()改为line.strip().split()则可得到每个单词

str.split(subStr): 字符串截断函数，将字符串以subStr为分界线分开变为一个由子字符串组成的list（subStr分开后消失）。subStr默认值为空格。

str.join(list): 将str加在list每相邻的两个item之间，返回一个新的string，list中的item必须都是string。一个主要用处是对list的items加空格，并变成一句话。

还有一些判断函数（如以is开头，返回bool型）：str.islower(), str.isupper(), str.startswith(subStr), str.endswith(subStr), str.isalpha(), sub.isdigit() 等。

最后，Python也支持八进制(oct)，十六进制(hex)，及ascii码转换（chr(), ord()）. 这两个函数并不是字符串方法，而是Python的内建函数，所以使用的时候，是传递参数，而不是像str.method() 这样来使用。如：

>>> chr(112) #ascii码转数字或字符，输入为0<=value<=255

'p'

>>> ord('p') #数字或字符转ascii码，参数为str

112

>>> hex(11) #十进制转16进制，结果为str

'0xb'

>>> oct(11) #十进制转8进制，结果为str

'013'

>>> unichr(112) #unicode码值转数字或字符

u'p'

每一门语言都离不开字符串以及字符串操作，Python也为字符串定义了很多函数。

1、字符串切片

sub = string[起始位置:结束为止:步长]

获得子字符串

string = "http://www.jianshu.com/p/2cb6159d3f13"

print("%s 子字符串:%s"%(string, string[7:15]))

获得所有字符串

string = "http://www.jianshu.com/p/2cb6159d3f13"

print("%s 打印全部是:%s"%(string, string[0:len(string)]))

print("%s 打印全部是:%s"%(string, string[0:]))

print("%s 打印全部是:%s"%(string, string[:]))

print("%s 打印全部是:%s"%(string, string[::]))

反向打印字符串

string = "http://www.jianshu.com/p/2cb6159d3f13"

print("%s 反向打印字符串是:%s"%(string, string[::-1]))

2、find查找前缀，rfind查找后缀

string = "http://www.jianshu.com/p/2cb6159d3f13"

findIndex = string.find(":")

if findIndex == -1:

print("%s 没有找到的协议"%string)

else:

print("%s 找到的协议是:%s"%(string,string[0:findIndex]))

fileName = "/test/A/a.text"

findIndex = fileName.rfind(".")

if findIndex == -1:

print("文件%s 没有找到扩展名"%fileName)

else:

print("文件%s 找到的扩展名是:%s"%(fileName, fileName[findIndex+1:]))

3、如果使用index和rindex，需要注意，如果没有子字符串，则会出现异常

string = "http://www.jianshu.com/p/2cb6159d3f13"

print("%s 使用的协议是:%s"%(string, string[0:string.index(":")]))

print("%s 使用的协议是:%s"%(string, string[0:string.rindex(":")]))

4、count测试字符串重复出现的次数

一般用法：

string = "http://www.jianshu.com/p/2cb6159d3f13"

subString = "jianshu"

print("%s 在‘%s’重复出现了%d次"%(subString, string, string.count(subString)))

特殊用法：count测试的时候可以指定起始位置和结束位置

string = "http://www.jianshu.com/p/2cb6159d3f13"

subString = "jianshu"

print("%s在‘%s’的[%d,%d]区间重复出现了%d次"%(subString, string, 20, 30, string.count(subString, 20, 30)))

5、replace替换字符串中的子字符串

用replace替换字符串时，需要注意原字符串不改变，替换后的字符串以返回形式返回

string = "http://www.jianshu.com/p/2cb6159d3f13"

newString = string.replace("http", "https")

print("%s 替换协议后是:%s"%(string,newString))

用replace默认替换所有匹配的，也可以指定替换几个

message = "hello world! Ha Ha ha"

newMessage1 = message.replace("Ha", "ha")

print(newMessage1)

newMessage2 = message.replace("Ha", "ha", 1)

print(newMessage2)

6、split分隔字符串

string = "http://www.jianshu.com/p/2cb6159d3f13"

print(string.split("://"))

7、title:字符串每个单词首字母大写和capitalize:字符串首字母大写

message = "hello world! Ha Ha ha"

temp1 = message.title()

temp2 = message.capitalize()

print((message, temp1, temp2))

8、startswith和endswith

判断是否以某个字符串开头或结尾。

string = "http://www.jianshu.com/p/2cb6159d3f13"

if string.startswith("http"):

print("字符串是以:http开头的");

fileName = "/test/A/a.text"

if fileName.endswith("text"):

print("字符串是以:text结尾的");

9、lower和upper

将字符串以小写或大写形式输出。

message = "hello world! Ha Ha ha"

newMessage1 = message.lower()

newMessage2 = message.upper()

print("%s转换后为:%s====%s"%(message, newMessage1, newMessage2))

10、ljust、rjust和center

给定字符串宽度，输出按照左对齐，右对齐和中间对齐符串。

message = "hello world! Ha Ha ha"

print(message.ljust(50))

print(message.rjust(50))

print(message.center(50))

11、lstrip、rstrip和strip

输出去掉字符串前、后或者前后空格的字符串。

message = " hello world! Ha Ha ha "

print(message.lstrip(" "))

print(message.rstrip(" "))

print(message.strip(" "))

12、partition和rpartition

将字符串分成三部分，区别与split。

string = "http://www.jianshu.com/p/2cb6159d3f13"

print(string.partition("://"))

print(string.rpartition("://"))

13、join链接字符串

strings = ["hello", "world", "ha"]

strTemp = "\*"

strTemp = strTemp.join(strings)

print(strTemp)