【Python小课堂】第二课-趣玩Python字符串

Original 2016-10-03 哈哈小 菜猿观世界



注:为了方便,本文用python表示Python。

又到了跟哈哈小学python的时间了,哈哈~

今天我们来对python中的一种重要的数据结构-字符串进行讲解。

字符(character)是各种文字和符号的总称,包括各个国家的文字,标点符号,图形符号,数字等。那字符串(string)因此就好理解了,就好比由字符连成的一串文字或符号。例如:

<!--~~@#\$%^&luanqibazaodiwoyebuzhidaoshurulexieshenmehahahah.笑??!!~~-->

好吧,我们来看看在python中,关于字符串,哪些需要我们了解或注意的?



●● 原始字符串,长字符串,拼接及切片 ●●

首先,假如我要打出一文件路径字符串: "C:\nihao", 看下示代码:

```
>>> str='C:\nihao'
>>> str
'C:\nihao'
>>> print(str)
C:
ihao
>>> str="C:\\nihao"
>>> str
'C:\\nihao'
>>> print(str)
C:\nihao
>>> print(str)
```

哈哈小按码说人话:

单双引号字符串没区别,但是不能开始用单引,结束用双引。

当用print 打印str的时候,\n被当做转移字符换行输出,我们用\\n方式解决了此问题。

既然提到字符串转义字符,哈哈小觉得有必要附加一张转移字符含义表,大家可以照表尝试哦。

○ 字符串转义字符含义

符号 描述

\"	双引号
\n	换行
\a	系统发出响铃声
\b	退格
\t	横向制表符TAB
/r	回车符
\v	纵向制表符
\f	换页符
\0	八进制代表的字符
\x	十六进制代表的字符
\0	表示一个空字符
//	反斜杆
/,	单引号

说到这里,哈哈小表示有反对意见了,如果我要输出一串老长老长的带巨多巨多斜杆n的字符串,每个地方都去加\\n, 那还不得累死个人啊。

于是哈哈小要就此立下flag,决心要东天取经,寻找大法(好像这个也挺累的额)。

苦苦求索,蓦然回首,我遇见了raw character

什么鬼?原始字符串,长这幅模样:

```
>>> str =r'C:\nihao\women\kaishi\jiang\zifuchuan'
>>> str
'C:\\nihao\\women\\kaishi\\jiang\\zifuchuan'
```

如果后面再调用print打印str,想必大家能立马想到答案。

取法归来,哈哈小感慨万千,思忖良久,遂赋诗一首:

迎着黎明还未到来,破晓之际,我希望紧随时间的步伐,及早踏上我心向神往的归途。

云卷云舒,又有何惧何畏?

花开花落,又有何忧何扰?

淡定从容,是泥垢缠身的勇士指向远方的一记永恒清尘。

哈哈小表示想把这首打印出来,于是我就这样做了:

>>> str="迎着黎明还未到来,破晓之际,我希望紧随时间的步伐,及早踏上我心向神往的归途。

云卷云舒,又有何惧何畏?

花开花落,又有何忧何扰?

淡定从容,是泥垢缠身的勇士指向远方的一记永恒清尘。"

SyntaxError: EOL while scanning string literal

ouch, 这样好像貌似可能不行。

我就不信了, 多加两个引号试试。

>>> str="""迎着黎明还未到来,破晓之际,我希望紧随时间的步伐,及早踏上我心向神往的归途。

云卷云舒,又有何惧何畏?

花开花落,又有何忧何扰?

淡定从容,是泥垢缠身的勇士指向远方的一记永恒清尘。"""

sas ctr

'迎着黎明还未到来,破晓之际,我希望紧随时间的步伐,及早踏上我心向神往的归途。\n\n云卷云舒,又有何惧何畏?\n\n花开花落,又有何忧何扰?\n\n淡定从容,是泥垢缠身的勇士指向远方的一记永恒清尘。'

>>> print(str)

迎着黎明还未到来,破晓之际,我希望紧随时间的步伐,及早踏上我心向神往的归途。

云卷云舒, 又有何惧何畏?

花开花落,又有何忧何扰?

淡定从容,是泥垢缠身的勇士指向远方的一记永恒清尘。

>>>

Bingo, 行了。

哈哈小可以偷偷告诉你,刚才我要输出一首诗(有很多行),在python可以叫做长字符串,如果要顺利地输出它,我们可以用三个引号去表示。

赋诗之后,哈哈小觉得酣畅淋漓,不虚此行。

所以呢?人总是要有目标的,哪怕小目标也是好的。(这个转折,我只能说,额(⊙o⊙)...,没有什么想说的)

因此, set a small goal for myself,

比如, such as earning 100B yuan next year.

所以:

```
>>> 'set a small goal for myself,''such as earning 100B yuan next year.'
'set a small goal for myself,such as earning 100B yuan next year.'
>>> 'set a small goal for myself,'+'such as earning 100B yuan next year.'
'set a small goal for myself,such as earning 100B yuan next year.'
>>>
```

上图给大家展示的是python中字符串的拼接, so easy~

讲到这,哈哈小表示很过瘾,很带劲啊。再来。比如,我觉得那个数字挺招人喜欢的,那就单独把它切出来 玩玩吧,哈哈~。

```
>>> str='set a small goal for myself'+'such as earning 100B yuan.'
>>> str
'set a small goal for myselfsuch as earning 100B yuan.'
>>> len(str)
53
>>> str[:-10]
'set a small goal for myselfsuch as earning '
>>> str[-10:]
'100B yuan.'
>>> str[43:]
'100B yuan.'
>>>
```

有必要解释一下字符串中的切片:len()是获取字符串长度的方; str [:x]表示截取从开始(也就是0)到

x位置的字符串;负数表示倒着数;可以利用切片轻松实现字符串的插入。

Ok, 这一啪就stop here了。



●● 内置方法向右看齐,立定! ●●

Attention! 接下来哈哈小要讲一些关于字符串操作的内置方法(built-in functions):

在这里要说明的是,python 字符串的内置方法有很多,哈哈小只是挑选一些常用的进行讲解,不可能一一涵 盖。

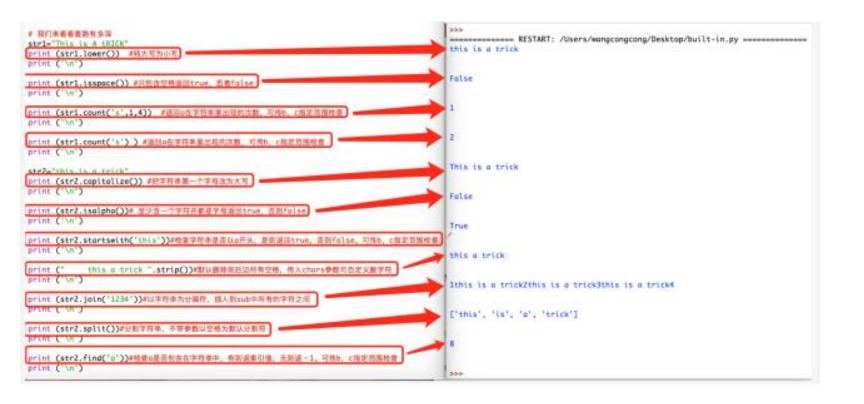
免得被打, 我觉得还是先上图为妙。

○ 字符串的一些内置方法

方法 描述

lower()	转大写为小写
count(a,[b,[,c]])	返回a在字符串里出现的次数,可传b,c指定范围检查
capitalize()	把字符串第一个字母改为大写
isalpha()	至少含一个字符并都是字母返回true, 否则false
isalnum()	至少含一个字符并都是字母或数字返回true, 否则false
center(width)	居中,并使用空格填充至长度width的新字符串
islower()	至少包含一个区分大小写的字符,并都是小写,返回true,否则false
isnumeric()	只包含数字字符返回true, 否则false
isspace()	只包含空格返回true, 否者false
join(sub)	以字符串为分隔符,插入到sub中所有的字符之间
strip([chars])	默认删除前后边所有空格,传入chars参数可自定义删字符
startwith(a,[,b[,c])	检查字符串是否以a开头,是则返回true,否则false。可传b,c指定范围检查
find(a,[,b[,c])	检查a是否包含在字符串中,有则返索引值,无则返-1。可传b,c指定范 围检查

再用一些代码来解释,一图以蔽之:



有了这些奇葩的内置方法,对字符串的操作是不是感觉方便多了呢?

更多方法请参看python说明文档。



●● 今天最后一啪-格式化 ●●

各位观众,听众,嗅众,吃瓜群众。请注意了,last but not least,哈哈小最后要给大家讲的是python中关于字符串的格式化。

讲到格式化,我们首先要提到一个函数,format()

怎么个用法呢?

比如, 我要输出这么一段字符串:

just hate those speaking blindly the truth in front of others like you. (就讨厌像你这种在别人边瞎说大实话的人,哈哈~)

可以这样写:

```
>>> "just hate {0} speaking {1} the truth in front of {2} like {3}".format("those","blindly","others","you")
'just hate those speaking blindly the truth in front of others like you'
>>> "just hate {a} speaking {b} the truth in front of {c} like {d}".format(a="those",b="blindly",c="others",d="you")
'just hate those speaking blindly the truth in front of others like you'
>>> '{0:.5f}{1}'.format(6.666664, 'awesome')
'6.666666awesome'
>>>
```

说明一下:

● 第一句用到的是<u>位置参数</u>,大括号放置一个数字,数字值表示位置索引值(记住索引值是从0开始哦)。 后面format函数传入的参数按顺序依次对应,最后打印出来。

- 第二句用到的东东叫<u>关键字参数</u>,前面字符串中出现的字母a,b,c,d可以说是一些未知数,后面for-mat函数传入的参数以名字对应好就行,不必强调顺序。相较位置参数而言,是不是随心所欲多了呢?
- 最后输出的6.66666awesome,是位置参数的一个实例。对于 {0:.5f} 这个奇怪的东西有点懵c啊。其实很简单,0表示位置数,:表示我要在后面对这个位置的字符串定制一些格式,这里哈哈小给定的格式 (.5f) 的意思是,输出保留小数点后五位的定点数(在这里,大家姑且把定点数当做浮点数看就行)。

为了更多地了解字符串格式化,我们需要对以下的字符串格式化符号含义稍加以理解应用。

○ 字符串格式化符号含义

符号	描述
----	----

%с	格式化字符及其ASCII码	
%s	格式化字符串	
%d	格式化整数	
‰	格式化无符号八进制数	
%x	格式化无符号十六进制数	
%X	同上	
%f	格式化定点数, 可指定小数点后精度	
%e	用科学计数法格式化定点数	
%E	同上	
%g	根据值的大小决定使用%f或%e	
%G	同上	

打几个比方:

如果你在命定提示行写下这么一句: '%c %c %c'%(97,98,99)

没错,最后输出的结果讲会是: 'a b c', 因为a,b,c字符对应的ASCII码值分别为97,98,99.

再比如: 十进制数以八进制输出 >>> '%o'%10 '12'

十进制数以十六进制输出(去试试大写的X,看看有什么区别)

>>> '%x'%12

'C'

做加法运算, cool~

>>> 1%d+%d=%d'%(5,6,5+6)

'5+6=11'

>>>

除了要了解上面基本的字符串格式化符,掌握一些辅助符号,对我们格式化字符串可大有帮助哦。

〇 格式化辅助操作符

符号

描述

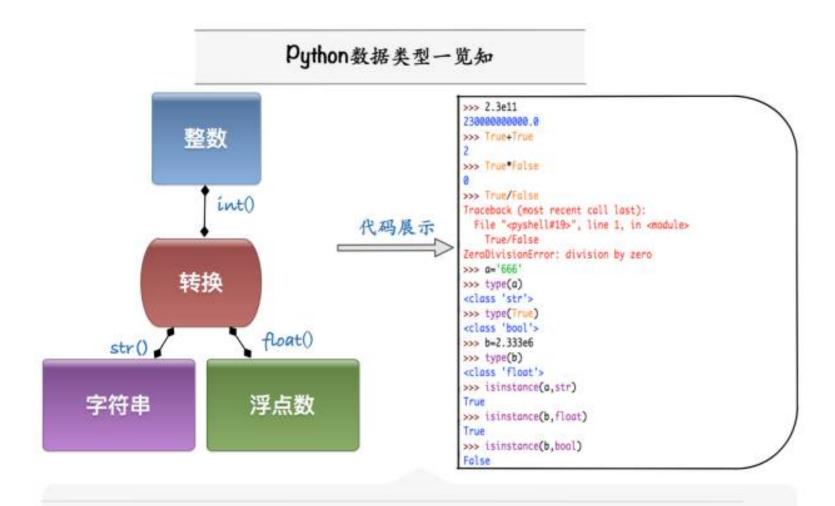
a.b	a为显示的最小总宽度, n是小数点后的位数
-	用于左对齐
+	在正数前加+
#	在八进制数前显示0o,十六进制前显示0x或0X
0	显示数字前面填充'0',取代空格

说明:想必大家对第一个a.b应该很熟悉了,前面输出定点数的时候也多次用到。

像另外三种可以这么用:

```
>>> '%-5d'%10
'10 '
>>> '%#o'%11
'0o13'
>>> '%#x'%11
'0xb'
>>> '%010d'%11
'00000000011'
```

字符串可以算作python的一种数据类型,前面有讲到整型,也有提过浮点型。最后,哈哈小以一张图来对这三种数据类型做一个概括总结吧。



有上图可知, 三种类型之间可以互相转换

由代码可知, 2.3Ell表示科学计数法, 属于浮点数

Python中的布林类型很奇葩,可以直接做数学运算,True为1,False为0可用type(),isinstance()两种方法去判断数据类型,前者返回类型,后者返回判断结果是(True)与不是(False)

好的,今天跟哈哈小深入浅出学python的时间就到此结束了。最后,哈哈小还想啰嗦一句,要想扎扎实实学

好一门编程语言,如果光靠哈哈小在这讲,你们光看不练,那可是万万不能的哦。虽说前两次哈哈小都没留课后思考题,从下次课开始就要留了哦,敬情期待吧~

哈哈小偷偷透露一句,下节课我要讲的是分支与循环哦。

结束语:我们不想给你带来多少多少的知识,我们只想尽绵薄之力给你带来可能的智慧的启迪。

____ END ____

菜猿编程宝典 重新定义编程入门教育





长按,识别二维码,加关注

声明:本文为原创文章,文章仅代表作者本人观点。转载请注明作者信息及文本链接,谢谢您的阅读和支持,期待您的喜欢和评论。