链滴

Python 实战（五）| 字符串

作者：JavaFish

原文链接：https://ld246.com/article/1626434436658

来源网站：链滴

01 前言

哈喽，我是狗哥，一名程序猿。做过 Android、撸过 Java、目前在自学 Python 。注册 「一个优秀废人」这个公号已有些日子，真正有心将它运营起来是这两天萌生的想法。注册这个号的初衷是分享的 Python 学习笔记。一个知识，你自己懂，不算是真的懂，你能把他人说懂，才是真正掌握。分享是一个最好的复习过程。

02 Python 字符串

字符串可以说是程序员最常用的数据类型了，而在 Python 中我们可以通过使用 ('或")来创建字符串。

创建字符串很简单，只要为变量分配一个值即可。例如：

```
var1 = 'Hello World!'
var2 = "Runoob"
```

####Python 访问字符串中的值

不同于 Java 和 C 等语言(有 char 类型)，Python 不支持单字符类型，单字符在 Python 中也是作为个字符串使用。

Python 想要访问子字符串，可以使用方括号(字符串下标从 0 开始)来截取字符串，如下实例：

```
var1 = 'Hello World!'
var2 = "Nasus"

print ("var1[0]: ", var1[0])
print ("var2[1:4]: ", var2[1:4])
```

以上实例执行结果：

```
var1[0]:  N
var2[1:5]:  asus
```

####Python 字符串更新

Python 字符串更新常用方法是：截取字符串的一部分并与其他字段拼接，如下实例：

```
var1 = 'Hello World!'

print ("已更新字符串 : ", var1[:6] + 'Nasus!')
```

以上实例执行结果：

```
已更新字符串 :  Hello Nasus!
```

03 Python 转义字符

当我们需要在字符中使用特殊字符时，python 用反斜杠()
义字符。如下表：

转义字符	描述
\(在行尾时)	续行符

\	反斜杠符号
'	单引号
"	双引号
\a	响铃
\b	退格(Backspace)
\e	转义
\000	空
\n	换行
\v	纵向制表符
\t	横向制表符
\r	回车
\f	换页
\oyy	八进制数，yy代表的字符，例如：\o12代换行
\xyy	十六进制数，yy代表的字符，例如：\x0a代换行
\other	其它的字符以普通格式输出

####Python 字符串运算符

下表实例变量 a 值为字符串 "Hello"，b 变量值为 "Python"：

操作符	描述	实例
+	字符串连接	a + b 输出结果：elloPython
*	重复输出字符串	a*2 输出结果 HelloHello
[]	通过索引获取字符串中字符	a[1 输出结果 e
[:]	截取字符串中的一部分	a[1:4] 输出结果 ell
in	成员运算符 - 如果字符串中包含给定的字符返回 True	H' in a 输出结果 1
not in	成员运算符 - 如果字符串中不包含给定的字符返回 True	M' not in a 输出结果 1
r/R	原始字符串 - 原始字符串：所有的字符串都是直接按照字面的思来使用，没有转义特殊或不能打印的字符。 原始字符串除在字符串的第一个引号前加上字母 r（可大小写）以外，与普通字符串有着几乎完全相同的语法。	print(r'\n') print(R'\n')
%	格式字符串	请看下一节内容。

以下代码用于验证上述表格的字符串运算符作用：

```
a = "Hello"
```

```
b = "Python"

print("a + b 输出结果：", a + b)
print("a * 2 输出结果：", a * 2)
print("a[1] 输出结果：", a[1])
print("a[1:4] 输出结果：", a[1:4])

if( "H" in a) :
   print("H 在变量 a 中")
else :
   print("H 不在变量 a 中")

if( "M" not in a) :
   print("M 不在变量 a 中")
else :
   print("M 在变量 a 中")

print (r'\n')
print (R'\n')
```

以上实例输出结果为：

```
a + b 输出结果： HelloPython
a * 2 输出结果： HelloHello
a[1] 输出结果： e
a[1:4] 输出结果： ell
H 在变量 a 中
M 不在变量 a 中
\n
\n
```

04 Python 字符串格式化

Python 支持格式化字符串的输出 。尽管这样可能会用到非常复杂的表达式，但最基本的用法是将一值插入到一个有字符串格式符 %s 的字符串中

在 Python 中，字符串格式化使用与 C 中 sprintf 函数一样的语法：

```
print ("我叫 %s 今年 %d 岁!" % ('狗哥', 24))
```

以上实例输出结果：

```
我叫 狗哥 今年 24 岁!
```

python字符串格式化符号:

符号	描述
%c	格式化字符及其ASCII码
%s	格式化字符串
%d	格式化整数
%u	格式化无符号整型
%o	格式化无符号八进制数

%x	格式化无符号十六进制数
%X	格式化无符号十六进制数（大写）
%f	格式化浮点数字，可指定小数点后的精度
%e	用科学计数法格式化浮点数
%E	作用同%e，用科学计数法格式化浮点数
%g	%f和%e的简写
%G	%f 和 %E 的简写
%p	用十六进制数格式化变量的地址

格式化操作符辅助指令:

符号	**功能**
*	定义宽度或者小数点精度
-	用做左对齐
+	在正数前面显示加号(+)
<sp>	在正数前面显示空格
#	在八进制数前面显示零('0')，在十六进制前面显示'0x'或者'0X'(取决于用的是'x'还是'X')
0	显示的数字前面填充'0'而不是默认的空格
%	'%%'输出一个单一的'%'
(var)	映射变量(字典参数)
m.n.	m 是显示的最小总宽度,n 是小数点后的位数(如果可用的话)

Python2.6 开始，新增了一种格式化字符串的函数 str.format()，它增强了字符串格式化的功能。

#05 Python 三引号

python 三引号允许一个字符串跨多行，字符串中可以包含换行符、制表符以及其他特殊字符。实例下

```
para_str = """这是一个多行字符串的实例
多行字符串可以使用制表符
TAB ( \t )。
也可以使用换行符 [ \n ]。
"""
print (para_str)
```

以上实例执行结果为:

```
这是一个多行字符串的实例
多行字符串可以使用制表符
TAB (    )。
也可以使用换行符 [
 ]。
```

三引号让程序员从引号和特殊字符串的泥潭里面解脱出来，自始至终保持一小块字符串的格式是所谓

WYSIWYG（所见即所得）格式的。

一个典型的用例是，当你需要一块 HTML 或者 SQL 时，这时用字符串组合，特殊字符串转义将会非的繁琐。如果使用 Python 三引号这事情将会变得轻而易举。

```
errHTML = '''
<HTML><HEAD><TITLE>
Friends CGI Demo</TITLE></HEAD>
<BODY><H3>ERROR</H3>
<B>%s</B><P>
<FORM><INPUT TYPE=button VALUE=Back
ONCLICK="window.history.back()"></FORM>
</BODY></HTML>
'''
cursor.execute('''
CREATE TABLE users (
login VARCHAR(8),
uid INTEGER,
prid INTEGER)
''')
```

06 Unicode 字符串

在 Python2 中，普通字符串是以 8 位 ASCII 码进行存储的，而 Unicode 字符串则存储为16 位 unicde 字符串，这样能够表示更多的字符集。使用的语法是在字符串前面加上前缀 u。而在 Python3 中所有的字符串都是Unicode字符串。

07 Python 的字符串内建函数

Python 中有很多内建函数，我们在开发中会经常用到，以下就是 Python 的字符串常用内建函数：

序号	方法及描述
1	capitalize()将字符串的第一个字符转换为大写
2	center(width, fillchar)返回一个指定的宽度 width 居中的字符串，fillchar 为填充的字符，默认为空格。
3	count(str, beg= 0,end=len(string))返回 str 在string 里面出现的次数，如果 beg 或者 end 指定则返回指定范围内 str 出现的次数
4	bytes.decode(encoding="tf-8", errors="strict") ython3 中没有 decode 方法，但我们可以使用 bytes 对象的 decode() 方法来解码给定的 bytes 对，这个 bytes 对象可以由 str.encode() 来编码返回。
5	encode(encoding='TF-8',errors='strict') encoding 指定的编码格式编码字符串，如果出错默认报一个ValueError 的异常，除非 errors 指定是'ignore'或者'replace'
6	endswith(suffix, beg=0, end=len(string))检字符串是否以 obj 结束，如果beg 或者 end 指定则检查指定的范围内是否以 obj 结束，如果是，返回 True,否则返回 False.
7	expandtabs(tabsize=8)把字符串 string 中的 t

b 符号转为空格，tab 符号默认的空格数是 8 。

8 find(str, beg=0 end=len(string))检测 str 是包含在字符串中，如果指定范围 beg 和 end ，则检查是否包含在指定范围内，如果包含返回开始的引值，否则返回-1

9 index(str, beg=0, end=len(string))跟find()方一样，只不过如果str不在字符串中会报一个异常.

10 isalnum()如果字符串至少有一个字符并且所字符都是字母或数字则返 回 True,否则返回 False

11 isalpha()如果字符串至少有一个字符并且所有符都是字母则返回 True, 否则返回 False

12 isdigit()如果字符串只包含数字则返回 True 则返回 False..

13 islower()如果字符串中包含至少一个区分大小的字符，并且所有这些(区分大小写的)字符都是小写，则返回 True，否则返回 False

14 isnumeric()如果字符串中只包含数字字符，返回 True，否则返回 False

15 isspace()如果字符串中只包含空白，则返回 Tre，否则返回 False.

16 istitle()如果字符串是标题化的(见 title())则返回 True，否则返回 False

17 isupper()如果字符串中包含至少一个区分大写的字符，并且所有这些(区分大小写的)字符都是大写，则返回 True，否则返回 False

18 join(seq)以指定字符串作为分隔符，将 seq 所有的元素(的字符串表示)合并为一个新的字符串

19 len(string)返回字符串长度

20 ljust(width[, fillchar])返回一个原字符串左对齐并使用 fillchar 填充至长度 width 的新字符串，fillchar 默认为空格

21 lower()转换字符串中所有大写字符为小写

22 lstrip()截掉字符串左边的空格或指定字符。

23 maketrans()创建字符映射的转换表，对于接两个参数的最简单的调用方式，第一个参数是字符串，表示需要转换的字符，第二个参数也是字符串示转换的目标。

24 max(str)返回字符串 str 中最大的字母。

25 min(str)返回字符串 str 中最小的字母。

26 replace(old, new [, max])把 将字符串中的 str 替换成 str2,如果 max 指定，则替换不超过 max 次。

27 rfind(str, beg=0,end=len(string)) 类似于 find()函数，不过是从右边开始查找.

28 rindex(str, beg=0, end=len(string)) 类似于 ndex()，不过是从右边开始.

29 rjust(width,[, fillchar]) 返回一个原字符串右齐,并使用fillchar(默认空格）填充至长度 width 的新字符串

30 rstrip() 删除字符串字符串末尾的空格.

31 split(str="" num=string.count(str)) num=string.count(str)) 以 str 为分隔符截取字符串，如果 num 有指定值则仅截取 num 个子字符串

32 splitlines([keepends]) 按照行('\r', '\r\n', \n' 分隔，返回一个包含各行作为元素的列表，如果参数 keepends 为 False，不包含换行符，如果为 Tru ，则保留换行符。

33 startswith(str, beg=0,end=len(string)) 检字符串是否是以 obj 开头，是则返回 True，否则返回 False。如果beg 和 end 指定值，则在指定范内检查。

34 strip([chars]) 在字符串上执行 lstrip()和 rstrip()

35 swapcase() 将字符串中大写转换为小写，小转换为大写

36 title() 返回"标题化"的字符串,就是说所有单词是以大写开始，其余字母均为小写(见 istitle())

37 translate(table, deletechars="") 根据 str 给出的表(包含 256 个字符)转换 string 的字符, 要过滤掉的字符放到 deletechars 数中

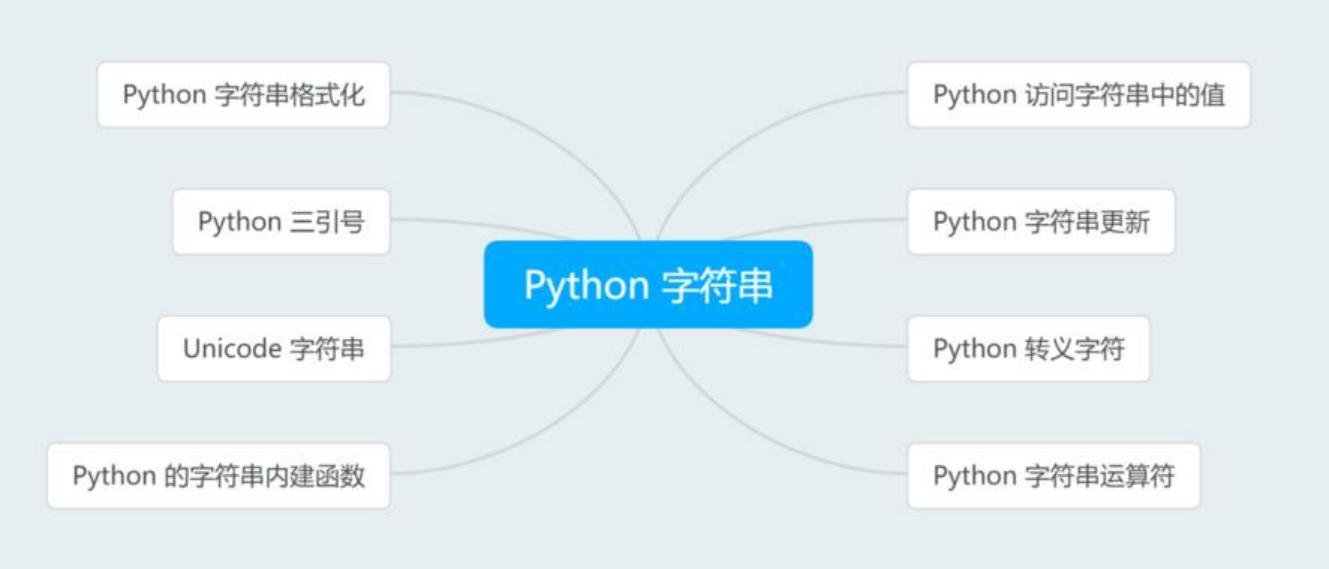
38 upper() 转换字符串中的小写字母为大写

39 zfill (width) 返回长度为 width 的字符串，原符串右对齐，前面填充0

40 isdecimal() 检查字符串是否只包含十进制字 ，如果是返回 true，否则返回 false。

08 后语

写完这篇文章看了下表，已是北京时间 2018年 6 月 24 日的凌晨 3:15 分。累瘫，但还是按照惯例送思维导图帮助你们理解并记忆，但最重要的还是多动手自己实践，虽然看上去很简单，但是你自己敲遍可能会出现各种各样的问题。学习，千万不能有所见即所得的想法，实践才是检验真理的唯一标准。



好啦，本节内容就到这里，下节内容：「自学 Python 之列表(list)」