

# Python 字符串学习笔记 (持续更新)

作者: [angels](#)

原文链接: <https://ld246.com/article/1502269243492>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

# 字符串

- 写法 单引号、双引号均可

"hello world"

'hello world'

- 对引号转译

\'或者\"

- 拼接字符串

```
s1 = "hello"  
s2 = "world"  
s3 = s1+s2  
print(s1+s2)
```

- 字符串转换

str: 值转换为合理形式的字符串, (给人看的)

repr: 创建一个字符串, 以合法的python表达式的形式表示值,显示使用 (给python看的)

str()一般是将数值转成字符串。

repr()是将一个对象转成字符串显示, 注意只是显示用, 有些对象转成字符串没有直接的意思。如>list dict使用str()是无效的, 但使用repr可以, 这是为了看它们都有哪些值, 为了显示之用。

- 长字符串

- 1、用三个单引号或者双引号代替, 字符串可跨行

```
'''  
''''
```

- 2、行后加反斜杠

\

- 原始字符串

- 1、原始字符串以 `r` 开头,

- 2、原始字符串不能以 `\` 结尾, 仍要使用需对反斜杠进行转译`\\`

- Unicode字符串

- 1、字符串以 `u` 开头

## 字符串基本操作

- 所有的标准序列操作(索引、分片、乘法、判断成员资格、求长度、取最大值、最小值)对字符串都用。

- 注意: 字符串是不可变的

## 字符串格式化

- % 实现格式化, 想到了C / C++有没有

```
"hello %s ,my name is %s"%("name1","name2")
hello name1 ,my name is name2
```

- 保留小数点位数

```
v = 3.2321332;
print("%.3f"%v) f表示浮点数
23123.232
```

- Python还提供了模版字符串

```
s = Template(' hello $x');
print(s.substitute(x='boy'))
hello boy
```

```
s = Template(' hello ${x}s');
print(s.substitute(x='boy'))
hello boys
```

插入美元符号\$

```
s = Template(' hello $$ ${x}s');
print(s.substitute(x='boy'))
hello $ boys
```

使用字典存储匹配的值

```
s = Template(' hello $ ${x}s');
d={};
d['x'] = 'boy';
d['y'] = 'yy';
print(s.substitute(d))
hello $ boys
```

## 字符串常量

- string.digits:包含数字0-9的字符串
- string.ascii\_letters:包含所有字符 (大写或者小些的字符串)
- string.ascii\_lowercase:包含所有小写字母的字符串
- string.printable:包含所有可打印字符的字符串
- string.punctuation:包含所有标点的字符串
- string.ascii\_uppercase:包含所有大写字母的字符串

## 字符串方法

- 1、find:在一个较长的字符串中查找子字符串, 返回子字符串所在位置最左端索引, 未找到返回-1

```
str = "hello";
ix = str.find("el");
print(ix) : 1
```

- find提供了可选的起始点和终点参数

```
str = "hello hi hi hello"
ix = str.find('hello',1); 从索引为1开始找
print(ix) : 12 因为从索引1开始找, 第一个不符合, 所以找的是最后一个hello
str = "hello hi hi hello"
ix = str.find('hello',1,12); 从索引为1开始找
print(ix) : -1 从1找到12没找到
```

- 2、join: 在队列中添加元素,也就是将指定字符串添加到队列元素之间返回一个新的字符串

```
lst = ['1','2','3'];
str = '+';
print(str.join(lst)) : 1+2+3
print(str) : +
```

- 3、lower: 返回字符串的小写字符

```
str1 = "HELLO";
str2 = str1.lower();
print(str1) : HELLO
print(str2) : hello
```

- 4、replace: 返回某字符串所在匹配项被替换之后得到的字符串

```
str1 = "hello";
str2 = str1.replace('ll','xx');
print(str1) : hello
print(str2) : hexxo
```

- 5、split: 分割字符串, 与join反过来

```
str = "1+2+3";
lst = str.split('+');
print(lst) : ['1','2','3']
print(str) : 1+2+3
```

- 6、strip: 返回去除了两侧空格的字符串, 内部的不受影响
- 7、translate: 与replace类型, 其替换字符串中的单个字符, 是同时多个一起替换, 一般情况下性优于replace
- 使用translate时我们需要先获得一张转换表
- 静态函数str.maketrans()返回可用于str.translate()方法的转换表。
- 如果只有一个参数, 它必须是dict类型, 键key为长度为1的字符(unicode字符码或者字符), 值value为任意长度字符串或者None。键key对应的字符将被转换为值value对应的字符(串)。

- 如果有两个参数，他们长度必须相等，每一个x字符将被转换为对应的y字符。如果有第三个参数，对应的字符将被转换为None。

```
str1 = 'abcxyz123'  
trans = str1.maketrans('abc', 'ABC', '123456')  
str2 = str1.translate(trans)  
print(str2) : ABCxyz
```