

第 3 章 使用字符串

1.字符串格式化

一种和 C 语言差不多

```
format = "hello,%s %s" values=('world','!') print format % values
```

另一种模板字符串

```
from string import Template
s=Template("$x,world!")
s.substitute(x='hello')
```

基本的转换说明符

- 1. %字符 标记转换说明符的开始
- 2. 转换标志 - 左对齐 +在转换值之前加正负号 “ ”表示整数之前保留空格 0 表示位数不够用 0 填充
- 3. 最小字段宽度 至少应该具有的宽度
- 4. (.)后跟精度值 实数表示小数点后的位数 字符串则为最大字段宽度
- 5. 转换类型

表3-1 字符串格式化转换类型

转换类型	含 义
d, i	带符号的十进制整数
o	不带符号的八进制
u	不带符号的十进制
x	不带符号的十六进制（小写）
X	不带符号的十六进制（大写）
e	科学计数法表示的浮点数（小写）
E	科学计数法表示的浮点数（大写）
f, F	十进制浮点数
g	如果指数大于-4或者小于精度值则和e相同，其他情况与f相同
G	如果指数大于-4或者小于精度值则和E相同，其他情况则与F相同
c	单字符（接受整数或者单字符字符串）
r	字符串（使用repr转换任意Python对象）
s	字符串（使用str转换任意Python对象）

字符串方法:

- 1. find 查找子串，返回最左端索引，没找到返回-1
subject.find("11",1,30) 提供起点和终点
- 2. join 在队列中添加元素 队列元素必须是字符串
seq=['1','2','3'] '+' .join(seq)
- 3. lower 返回字符串的小写字母版
- 4. replace 返回字符串的所以匹配替换之后的字符串
- 5. split 将字符串分割为序列 默认所以空格为分隔符
- 6. strip 返回去除两侧空格的字符串
- 7. translate 替换单个字符 但能同时进行多个替换

表B-6 字符串方法

方 法	描 述
<code>string.capitalize()</code>	返回首字母大写的字符串的副本
<code>string.center(width[, fillchar])</code>	返回一个长度为 <code>max(len(string), width)</code> 且其中String的副本居中的字符串，两侧使用 <code>fillchar</code> （默认为空字符）填充
方 法	描 述
<code>string.count(sub[, start[, end]])</code>	计算子字符串 <code>sub</code> 的出现次数，可将搜索范围限制为 <code>string[start:end]</code>
<code>string.decode(encoding[, errors])</code>	返回使用给定编码方式的字符串的解码版本，由 <code>errors</code> 指定错误处理方式（'strict'、'ignore'或者'replace'）
<code>string.encode(encoding[, errors])</code>	返回使用给定编码方式的字符串的编码版本，由 <code>errors</code> 指定错误处理方式（'strict'、'ignore'或者'replace'）
<code>string.endswith(suffix[, start[, end]])</code>	检查 <code>string</code> 是否以 <code>suffix</code> 结尾，可使用给定的索引 <code>start</code> 和 <code>end</code> 来选择匹配的范围
<code>string.expandtabs([tabsize])</code>	返回字符串的副本，其中 <code>tab</code> 字符会使用空格进行扩展，可选择使用给定的 <code>tabsize</code> （默认为8）
<code>string.find(sub[, start[, end]])</code>	返回子字符串 <code>sub</code> 的第一个索引，如果不存在这样的索引的话返回-1。可选定义搜索的范围为 <code>string[start:end]</code>
<code>string.index(sub[, start[, end]])</code>	返回子字符串 <code>sub</code> 的第一个索引，或者在找不到索引的时候引发 <code>ValueError</code> 异常，可定义搜索的范围为 <code>string[start:end]</code>
<code>string.isalnum()</code>	检查字符串是否由字母或数字字符组成
<code>string.isalpha()</code>	检查字符串是否由字母字符组成
<code>string.isdigit()</code>	检查字符串是否由数字组成
<code>string.islower()</code>	检查字符串中所有基于实例的字符（字母）是否都为小写
<code>string.isspace()</code>	检查字符串是否由空格组成
<code>string.istitle()</code>	检查字符串中不基于实例的字母后面的基于实例的字符都是大写的，且其他的基于实例的字符都是小写的
<code>string.isupper()</code>	检查是否所有字符串中的基于实例的字符都是大写的
<code>string.join(sequence)</code>	返回其中 <code>sequence</code> 的字符串元素已用String连接的字符串
<code>string.ljust(width[, fillchar])</code>	返回一个长度为 <code>max(len(string), width)</code> 且其中String的副本左对齐的字符串，右侧使用 <code>fillchar</code> （默认为空字符）填充
<code>string.lower()</code>	返回一个字符串的副本，其中所有基于实例的字符都是小写的
<code>string.lstrip([chars])</code>	返回一个字符串副本，其中所有的 <code>chars</code> （默认为空白字符，比如空格、 <code>tab</code> 和换行符）都被从字符串开始处去除
<code>string.partition(sep)</code>	在字符串中搜索 <code>sep</code> 并返回（ <code>head</code> , <code>sep</code> , <code>tail</code> ）
<code>string.replace(old, new[, max])</code>	返回字符串的副本，其中 <code>old</code> 的匹配项都被替换为 <code>new</code> ，可选择最多替换 <code>max</code> 个
<code>string.rfind(sub[, start[, end]])</code>	返回子字符串 <code>sub</code> 被找到的位置的最后一个索引，如果不存在这样的索引则返回-1。可定义搜索的范围为 <code>string[start:end]</code>
<code>string.rindex(sub[, start[, end]])</code>	返回子字符串 <code>sub</code> 被找到位置的最后一个索引，如果不存在这样的索引则引发一个 <code>ValueError</code> 异常。可定义搜索范围为 <code>string[start:end]</code>
<code>string.rjust(width[, fillchar])</code>	返回一个长度为 <code>max(len(string), width)</code> 且其中 <code>string</code> 的副本右对齐的字符串，左侧使用 <code>fillchar</code> （默认为空字符）填充
<code>string.rpartition(sep)</code>	同 <code>Partition</code> ，但从右侧开始查找
<code>string.rstrip([chars])</code>	返回一个字符串副本，其中所有的 <code>chars</code> （默认为空白字符，比如空格、 <code>tab</code> 和换行符）都被从字符串结束处去除
<code>string.rsplit([sep[, maxsplit]])</code>	同 <code>split</code> ，但是在使用 <code>maxsplit</code> 时是从右向左进行计数
<code>string.split([sep[, maxsplit]])</code>	返回字符串中所有单词的列表，使用 <code>sep</code> 作为分隔符（如果未特别指出的话以空格切分单词），可使用 <code>maxsplit</code> 指定最大切分数

方 法	描 述
<code>string.splitlines([keepends])</code>	返回string中所有行的列表,可选择是否包括换行符(如果提供keepend参数则包括)
<code>string.startswith(prefix[,start[,end]])</code>	检查string是否以prefix开始,可使用给的索引start和end来定义匹配的范围
<code>string.strip([chars])</code>	返回字符串的副本,其中所有chars(默认为空格)都从字符串的开头和结尾去除(默认为所有空白字符,如空格、tab和换行符)
<code>string.swapcase()</code>	返回字符串的副本,其中所有基于实例的字符都交换大小写
<code>string.title()</code>	返回字符串的副本,其中单词都以大写字母开头
<code>string.translate(table[,deletechars])</code>	返回字符串的副本,其中所有字符都使用table(由string模块中的maketrans函数构造)进行了转换,可选择删除出现在deletechars中的所有字符
<code>string.upper()</code>	返回字符串的副本,其中所有基于实例的字符都是大写的
<code>string.zfill(width)</code>	在string的左侧以0填充width个字符