



链滴

Python-CookBook: 42、对数值做格式化输出

作者: [zhaolixiang](#)

原文链接: <https://ld246.com/article/1567162045710>

来源网站: 链滴

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

问题

我们需要对数值做格式化输出，包括控制位数、对齐、包含千位分隔符以及其他一些细节。

解决方案

要对一个单独的数值做格式化输出，使用内建的format()函数即可。示例如下：

```
>>> x = 1234.56789

>>> # Two decimal places of accuracy
>>> format(x, '0.2f')
'1234.57'

>>> # Right justified in 10 chars, one-digit accuracy
>>> format(x, '>10.1f')
' 1234.6'

>>> # Left justified
>>> format(x, '<10.1f')
'1234.6 '

>>> # Centered
>>> format(x, '^10.1f')
' 1234.6 '

>>> # Inclusion of thousands separator
>>> format(x, ',')
'1,234.56789'
>>> format(x, '0,.1f')
'1,234.6'
>>>
```

如果想采用科学计数法，只要把f改为e或者E即可，根据希望采用的指数规格来指定。示例如下：

```
>>> format(x, 'e')
'1.234568e+03'
>>> format(x, '0.2E')
'1.23E+03'
>>>
```

以上两种情况中，指定宽度和精度的一般格式为'[<>^]?width[,]?(.digits)?'，这里width和digits为整，而?代表可选的部分。同样的格式也可用于字符串的format()方法中。示例如下：

```
>>> 'The value is {:0,.2f}'.format(x)
'The value is 1,234.57'
>>>
```

讨论

对数值做格式化输出通常都是很直接的。本节展示的技术既能用于浮点型数，也能适用于decimal模中的Decimal对象。

当需要限制数值的位数时，数值会根据`round()`函数的规则来进行取整。示例如下：

```
>>> x
1234.56789
>>> format(x, '0.1f')
'1234.6'
>>> format(-x, '0.1f')
'-1234.6'
>>>
```

对数值加上千位分隔符的格式化操作并不是特定于本地环境的。如果要将这个需求纳入考虑，应该察一下`local`模块中的函数。还可以利用字符串的`translate()`方法交换不同的分隔字符。示例如下：

```
>>> swap_separators = { ord('.'): ',', ord(','): '.' }
>>> format(x, ',').translate(swap_separators)
'1.234,56789'
>>>
```

在很多Python代码中，常用`%`操作符来对数值做格式化处理。示例如下：

```
>>> '%0.2f' % x
'1234.57'
>>> '%10.1f' % x
' 1234.6'
>>> '%-10.1f' % x
'1234.6 '
>>>
```

这种格式化操作仍然是可接受的，但是比起更加现代化的`format()`方法，这种方法就显得不是那么强了。比如说，当使用`%`操作符来格式化数值时，有些功能就没法得到支持了（例如添加千位分隔符）。