

# format格式字符串

## 语法

它通过{}和:来代替%。

## “映射”示例

### 通过位置

```
In [1]: '{0},{1}'.format('kzc',18)
Out[1]: 'kzc,18'
In [2]: '{},{ {}'.format('kzc',18)
Out[2]: 'kzc,18'
In [3]: '{1},{0},{1}'.format('kzc',18)
Out[3]: '18,kzc,18'
```

字符串的format函数可以接受不限个参数，位置可以不按顺序，可以不用或者用多次，不过2.6不能为空{}，2.7才可以。

通过关键字参数

```
In [5]: '{name},{age}'.format(age=18,name='kzc')
Out[5]: 'kzc,18'
```

## 通过对象属性

```
class Person:
    def __init__(self,name,age):
        self.name,self.age = name,age
    def __str__(self):
        return 'This guy is {self.name},is {self.age} old'.format(self=self)
In [2]: str(Person('kzc',18))
Out[2]: 'This guy is kzc,is 18 old'
```

## 通过下标

```
In [7]: p=['kzc',18]
In [8]: '{0[0]},{0[1]}'.format(p)
Out[8]: 'kzc,18'
```

有了这些便捷的“映射”方式，我们就有了偷懒利器。基本的Python知识告诉我们，list和tuple可以通过“打散”成普通参数给函数，而dict可以打散成关键字参数给函数（通过和\*）。所以可以轻松的传个list/tuple/dict给format函数。非常灵活。

## 格式限定符

它有着丰富的“格式限定符”（语法是{}中带:号），比如：

## 填充与对齐

填充常跟对齐一起使用

^、<、>分别是居中、左对齐、右对齐，后面带宽度  
:号后面带填充的字符，只能是一个字符，不指定的话默认是用空格填充  
比如

```
In [15]: '{:>8}'.format('189')
Out[15]: '      189'
In [16]: '{:0>8}'.format('189')
Out[16]: '00000189'
In [17]: '{:a>8}'.format('189')
Out[17]: 'aaaaa189'
```

## 精度与类型f

精度常跟类型f一起使用

```
In [44]: '{:.2f}'.format(321.33345)
Out[44]: '321.33'
```

其中.2表示长度为2的精度，f表示float类型。

## 其他类型

主要就是进制了，b、d、o、x分别是二进制、十进制、八进制、十六进制。

```
In [54]: '{:b}'.format(17)
Out[54]: '10001'
In [55]: '{:d}'.format(17)
Out[55]: '17'
In [56]: '{:o}'.format(17)
Out[56]: '21'
In [57]: '{:x}'.format(17)
Out[57]: '11'
```

**用逗号还能用来做金额的千位分隔符。**

```
In [47]: '{:,}'.format(1234567890)
Out[47]: '1,234,567,890'
```