

Python

1 生成列表

要生成list [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 我们可以用range(1, 11):

```
>>> range(1, 11)
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

但如果要生成[1x1, 2x2, 3x3, ..., 10x10]怎么做? 方法一是循环:

```
>>> L = []
>>> for x in range(1, 11):
...     L.append(x * x)
...
>>> L
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

但是循环太繁琐, 而列表生成式则可以用一行语句代替循环生成上面的list:

```
>>> [x * x for x in range(1, 11)]
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

这种写法就是Python特有的列表生成式。利用列表生成式, 可以以非常简洁的代码生成 list。

range(1, 100, 2) 可以生成list [1, 3, 5, 7, 9,...]

2 复杂表达式

假设有如下的dict:

```
d = { 'Adam': 95, 'Lisa': 85, 'Bart': 59 }
```

可以通过一个复杂的列表生成式把它变成一个 HTML 表格:

```
tds = ['<tr><td>%s</td><td>%s</td></tr>' % (name, score) for name, score in d.iteritems()]
print '<table>'
print '<tr><th>Name</th><th>Score</th></tr>'
print '\n'.join(tds)
print '</table>'
```

字符串可以通过%进行格式化，用指定的参数替代 %s。
#\字符串的join()方法可以把一个 list 拼接成一个字符串。

把打印出来的结果保存为一个html文件，就可以在浏览器中看到效果了：

```
<table border="1">
<tr><th>Name</th><th>Score</th><tr>
<tr><td>Lisa</td><td>85</td></tr>
<tr><td>Adam</td><td>95</td></tr>
<tr><td>Bart</td><td>59</td></tr>
</table>
```

#红色可以用 <td style="color:red"> 实现。

3 条件过滤

列表生成式的 for 循环后面还可以加上 if 判断。例如：

```
>>> [x * x for x in range(1, 11)]
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

如果我们只想要偶数的平方，不改动 range()的情况下，可以加上 if 来筛选：

```
>>> [x * x for x in range(1, 11) if x % 2 == 0]
[4, 16, 36, 64, 100]
```

1. isinstance(x, str) 可以判断变量 x 是否是字符串；
2. 字符串的 upper() 方法可以返回大写的字母。

```
if type(x)==str
```

4 多层表达式

for循环可以嵌套，因此，在列表生成式中，也可以用多层 for 循环来生成列表。

对于字符串 'ABC' 和 '123'，可以使用两层循环，生成全排列：

```
>>> [m + n for m in 'ABC' for n in '123']
['A1', 'A2', 'A3', 'B1', 'B2', 'B3', 'C1', 'C2', 'C3']
```

翻译成循环代码就像下面这样:

```
L = []  
for m in 'ABC':  
    for n in '123':  
        L.append(m + n)
```