

Python语言基础与应用

高级特性 / 推导式

陈斌 北京大学 gischen@pku.edu.cn



推导式

- › **什么是推导式**
- › **推导式基本语法**
- › **生成器推导式**

什么是推导式

- › **推导式是从一个或者多个迭代器快速简洁地创建数据结构的一种方法**
- › **将循环和条件判断结合，从而避免语法冗长的代码**
- › **可以用来生成列表、字典和集合**

推导式基本语法

› 列表推导式

[<表达式> for <变量> in <可迭代对象> if <逻辑条件>]

› 字典推导式

{<键值表达式>:<元素表达式> for <变量> in <可迭代对象> if <逻辑条件>}

› 集合推导式

{<元素表达式> for <变量> in <可迭代对象> if <逻辑条件>}

推导式基本语法

```
>>> [x*x for x in range(10)]
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]
>>>
>>> {'K%d'%(x,):x**3 for x in range(10)}
{'K2': 8, 'K8': 512, 'K5': 125, 'K6': 216, 'K3': 27, 'K9': 729, 'K0': 0,
'K7': 343, 'K1': 1, 'K4': 64}
>>>
>>> {x*x for x in range(10)}
{0, 1, 64, 4, 36, 9, 16, 49, 81, 25}
>>>
>>> {x+y for x in range(10) for y in range(x)}
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17}

>>> [x+y for x in range(10) for y in range(x)]
[1, 2, 3, 3, 4, 5, 4, 5, 6, 7, 5, 6, 7, 8, 9, 6, 7, 8, 9, 10, 11, 7, 8, 9,
10, 11, 12, 13, 8, 9, 10, 11, 12, 13, 14, 15, 9, 10, 11, 12, 13, 14, 15,
16, 17]
>>>
>>> [x*x for x in range(10) if x % 2 == 0]
[0, 4, 16, 36, 64]
>>>
>>> [x.upper() for x in [1, 'abc', 'xyz', True] if isinstance(x, str)]
['ABC', 'XYZ']
```

生成器推导式

› 与推导式一样语法：

(**<元素表达式> for <变量> in <可迭代对象> if <逻辑条件>**)

› 返回一个生成器对象，也是可迭代对象

› 但生成器并不立即产生全部元素，仅在要用到元素的时候才生成，可以极大节省内存

生成器推导式

```
>>> agen = (x*x for x in range(10))  
>>> agen  
<generator object <genexpr> at 0x1078f5620>  
>>> for n in agen:  
    print (n)
```

```
0  
1  
4  
9  
16  
25  
36
```

- › 除了通过生成器推导式创建生成器，还可以使用下一节将介绍的生成器函数