[返回主页](https://www.cnblogs.com/gengyi/)

[耕毅](https://www.cnblogs.com/gengyi/)

* [博客园](https://www.cnblogs.com/)
* [首页](https://www.cnblogs.com/gengyi/)
* [新随笔](https://i.cnblogs.com/EditPosts.aspx?opt=1)
* [联系](https://msg.cnblogs.com/send/%E8%80%95%E6%AF%85)
* [管理](https://i.cnblogs.com/)

[python学习笔记013——推导式](https://www.cnblogs.com/gengyi/p/8094946.html)

**1 推导式简介**

推导式comprehensions（又称解析式），是Python的一种独有特性。

推导式是可以从一个数据序列构建另一个新的数据序列的结构体。

推导式有三种形式：

1）列表推导式 **(list)**

2）字典推导式 **(dict)**

3）集合推导式 **(set)**

**2 列表推导式**

是用可迭代对象生成列表的一种方法

列表推导式是表达式

**2.1 基本的列表推导式**

**语法**：

[表达式 for 变量 in 可迭代对象]

[表达式(具体操作) for 变量 in 可迭代对象 if 真值表达式]

**示例1：**[1,2,3,4,5,6.......,10]

[x for x in range(1, 11)]

或

L = []

for x in range(1, 10001):

L.append(x)

**示例2**：[1, 4, 9, 16, 25, ...]

Square=[x\*\*2 for x in range(1, 11)] #计算列表中所有元素的平方然后返回新的列表

**示例3**：[1 3 5 7 .... 99]

[x for x in range(1, 100) if x % 2 == 1]#获取1-100之间的奇数

**示例4**：生成1000以内的所有素数的列表

[复制代码](javascript:void(0);)

def is\_prime(x):

"""

判断x是否为素数(或质数)

如果为素数，返回True

"""

if x <= 1:

return False

for i in range(2, x):

if x % i == 0:

return False

return True

L = [x for x in range(1, 10000) if is\_prime(x)]

print(L)

[复制代码](javascript:void(0);)

**示例5：**

s1 = "ABC"  
s2 = "123"  
如何生成以下列表  
L = ['A1', 'A2', 'A3', 'B1', 'B2'...]

L = []

for x in s1:

for y in s2:

L.append(x + y)

如何生成以下列表  
L = ['A1', 'B2', 'C3']

[复制代码](javascript:void(0);)

s1 = 'ABC'

s2 = '123'

a = []

for i in range(len(s1)):

a.append(s1[i]+s2[i])

print(a)

[复制代码](javascript:void(0);)

**2.2 嵌套的列表推导式**

语法：

[表达式

for 变量1 in 可迭代对象1 if 条件表达式1

for 变量2 in 可迭代对象2 if 条件表达式2

]

**示例1：**

s1 = "ABC"  
s2 = "123"  
如何生成以下列表  
L = ['A1', 'A2', 'A3', 'B1', 'B2'...]

s1 = "ABC"

s2 = "123"

L = [x + y for x in s1 for y in s2]

**3 字典推导式**

**语法：**

{ 键表达式 : 值表达式 for 变量 in 可迭代对象 if 真值表达式 [for 变量 in 可迭代....]}

字典中的用法与列表中的用法类似，将 [ ] 换成  { } 即可

**示例1：**

生成表达式：{1:"1的平方", 4:"2的平方", 9:'3的平方', ..100:'10的平方'}

d = {x\*\*2 : str(x)+'的平方' for x in range(1, 11)}

**示例2：快速更换key和value**

mcase = {'a': 1, 'b': 2}

mcase\_frequency = {v: k for k, v in mcase.items()}

print(mcase\_frequency)

# output {10: 'a', 34: 'b'}

**示例3：**将字典中大小写a b 键所对应的值合并

[复制代码](javascript:void(0);)

mcase = {'a': 10, 'b': 34, 'A': 7, 'Z': 3}

mcase\_frequency = {

k.lower(): mcase.get(k.lower(), 0) + mcase.get(k.upper(), 0)

for k in mcase.keys()

if k.lower() in ['a','b']

}

print(mcase\_frequency)

# Output: {'a': 17, 'b': 34}

[复制代码](javascript:void(0);)

**备注： D.get(key, default)   返回键 key 所对应的值,如果没有此键，则返回 default（默认 None）**

参考 [python的各种推导式（列表推导式、字典推导式、集合推导式）](http://www.cnblogs.com/tkqasn/p/5977653.html)

**4 集合推导式**

 语法：

集合推导式语法：

{ 表达式 for 变量 in 可迭代对象 if 真值表达式 [for 变量 in 可迭代....]}

集合中的用法与列表中的用法类似，将 [ ] 换成  { } 即可

**示例1：**

squared = {x\*\*2 for x in [1, 1, 2]}

print(squared)

# Output: {1, 4}

**注意：集合中具有特有的性质，其元素不能重复**

**示例2：**

**生成集合：{1, 8, 27, 4\*\*4... n\*\*n}**

s = {x \*\* x for x in range(1, 10)}

**另外还有更为广泛的应用：**[python的推导式(实用)](http://blog.csdn.net/langb2014/article/details/53156032)  [python-列表推倒式-轻量级循环](http://blog.csdn.net/u012795264/article/details/50811662)