**Python 推导式**

Python 推导式是一种独特的数据处理方式，可以从一个数据序列构建另一个新的数据序列的结构体。

Python 支持各种数据结构的推导式：

* 列表(list)推导式
* 字典(dict)推导式
* 集合(set)推导式
* 元组(tuple)推导式

**列表推导式**

列表推导式格式为：

[表达式 for 变量 in 列表]

[out\_exp\_res for out\_exp in input\_list]

或者

[表达式 for 变量 in 列表 if 条件]

[out\_exp\_res for out\_exp in input\_list if condition]

* out\_exp\_res：列表生成元素表达式，可以是有返回值的函数。
* for out\_exp in input\_list：迭代 input\_list 将 out\_exp 传入到 out\_exp\_res 表达式中。
* if condition：条件语句，可以过滤列表中不符合条件的值。

过滤掉长度小于或等于3的字符串列表，并将剩下的转换成大写字母：

**实例**

>>> names = ['Bob','Tom','alice','Jerry','Wendy','Smith']  
>>> new\_names = [name.upper()**for** name **in** names **if** len(name)>3]  
>>> **print**(new\_names)  
['ALICE', 'JERRY', 'WENDY', 'SMITH']

计算 30 以内可以被 3 整除的整数：

**实例**

>>> multiples = [i **for** i **in** range(30) **if** i % 3 == 0]  
>>> **print**(multiples)  
[0, 3, 6, 9, 12, 15, 18, 21, 24, 27]

**字典推导式**

字典推导基本格式：

{ key\_expr: value\_expr for value in collection }

或

{ key\_expr: value\_expr for value in collection if condition }

使用字符串及其长度创建字典：

**实例**

listdemo = ['Google','Runoob', 'Taobao']  
# 将列表中各字符串值为键，各字符串的长度为值，组成键值对  
>>> newdict = {key:len(key) **for** key **in** listdemo}  
>>> newdict  
{'Google': 6, 'Runoob': 6, 'Taobao': 6}

提供三个数字，以三个数字为键，三个数字的平方为值来创建字典：

**实例**

>>> dic = {x: x\*\*2 **for** x **in** (2, 4, 6)}  
>>> dic  
{2: 4, 4: 16, 6: 36}  
>>> type(dic)  
<**class** 'dict'>

**集合推导式**

集合推导式基本格式：

{ expression for item in Sequence }

或

{ expression for item in Sequence if conditional }

计算数字 1,2,3 的平方数：

**实例**

>>> setnew = {i\*\*2 **for** i **in** (1,2,3)}  
>>> setnew  
{1, 4, 9}

判断不是 abc 的字母并输出：

**实例**

>>> a = {x **for** x **in** 'abracadabra' **if** x **not** **in** 'abc'}  
>>> a  
{'d', 'r'}  
>>> type(a)  
<**class** 'set'>

**元组推导式（生成器表达式）**

元组推导式可以利用 range 区间、元组、列表、字典和集合等数据类型，快速生成一个满足指定需求的元组。

元组推导式基本格式：

(expression for item in Sequence )

或

(expression for item in Sequence if conditional )

元组推导式和列表推导式的用法也完全相同，只是元组推导式是用 **()** 圆括号将各部分括起来，而列表推导式用的是中括号 **[]**，另外元组推导式返回的结果是一个生成器对象。

例如，我们可以使用下面的代码生成一个包含数字 1~9 的元组：

**实例**

>>> a = (x **for** x **in** range(1,10))  
>>> a  
<generator object <genexpr> at 0x7faf6ee20a50>  # 返回的是生成器对象  
  
>>> tuple(a)       # 使用 tuple() 函数，可以直接将生成器对象转换成元组  
(1, 2, 3, 4, 5, 6, 7, 8, 9)