

python的推导式

指的是从一个数据序列，构建出另一个数据序列的结构体。
是python独有的特性。
又叫解析式。

列表推导式

表达式： 是由变量和运算符的组合。是以一行形式展现的。表达式的结果必然是一个数据(值)。

列表推导式是采用表达式的方式直接过滤数据并且返回新的列表的操作

基本格式：

[表达式 for i in 容器 if 布尔表达式]

表达式：未来列表里的元素。

for循环： 生成数据

if ：筛选数据

```
# l1 = [num for num in range(5)]  
# print(l1)  
# 1-10之间的所有奇数  
# l1 = [num for num in range(1,11) if num %2==1]  
  
# l1 = [num if num % 2==1 else None for num in  
range(1,11)]
```

```
# print(l1)

# l1 = [num if num % 2==1 else None for num in
range(1,11) if num % 2==0]
# print(l1)

# l1 = [num*num1 for num in range(2) for num1 in
range(3) if num % 2 ==1 or num1 % 2 ==0]
# print(l1)

# 列表推导式嵌套
l1 = [num1 for num1 in [num for num in range(2)]]
print(l1)
```

字典推导式

格式：

{key:value for key,value in 可迭代对象 if 布尔表达式}

集合推导式

格式：

{表达式 for i in 可迭代对象 if 布尔表达式}

```
# 字典推导式
# dict1 = {"你好":"雷猴","好厉害":"猴赛雷"}
# dict2 = {key:values for key,values in dict1.items()}
# print(dict2)

# dict1 = {"你好":"雷猴","好厉害":"猴赛雷"}
# set1 = {"熏悟空","局八戒"}
# dict2 = {k:v for k in dict1 for v in set1 }
# print(dict2)

# 集合推导式
set1 = {str1 for str1 in ["你好","你号~","集合"] if len(str1) !=3}
print(set1)
```

小结：

推导式：1.简洁优雅，代码量少，可读性高，灵活性高
缺点：无法进行太过复杂的逻辑处理
推导式的执行效率更高