

1. lambda函数

1.1 基本定义

匿名函数lambda：没有名字的函数，是一种简单的、在同一行中定义函数的方法

- lambda函数可以接收任意多个参数，并且返回单个表达式的值
- lambda表达式只允许包含一个表达式
- 表达式的结果就是返回值
- lambda函数实际生成一个lambda对象
- 基本语法： `lambda arg1,arg2,...:<表达式>`
 - `arg1,arg2`为函数的参数，即输入
 - 表达式为函数体

1.2 例子

```
# 有限个输入输出
lambda x,y:x*y

# 没有输入, 输出是None
lambda:None

# 输入是任意个数的参数, 输出是他们的和
lambda *args:sum(args)

# 和if-else语句联动: 如果if为真, 则返回if前面的内容, 如果为假, 返回else后面的内容
lambda x:x+0.1 if x<1 else x-0.1

# 同样可以使用if not-else语句
```

2. filter()函数

2.1 基本定义

过滤函数filter(): 过滤掉不符合条件的元素，返回有符合条件元素组成的新列表

- 接受两个参数，第一个为函数，第二个为序列
- 序列中的每个元素传给函数进行判断，将结果为True的item放到新列表里
- 基础语法： `filter(function, iterable)`
 - `function`--判断函数
 - `iterable`--可迭代对象

- filter返回的是一个迭代器，如果需要把它转化为列表，可以使用list()

filter()的作用是简化循环写法：对列表中的值依次执行一次函数运算

2.2 例子

```
# 过滤列表中所有的奇数
is_odd = lambda x:x%2==1
list = [1,2,3,4,5,6,7,8,9]
newlist = filter(is_odd,list)

# 如果把lambda加入到filter中:
newlist = filter(lambda x:x%2==1,range(1,10))
```

3. apply()函数

这是pandas中的一个函数，可以作用于整个Series或者整个DataFrame

功能同样是自动遍历整个Series或者DataFrame，对每一个元素运行指定的函数

3.1 Series.apply()

```
# 如果a列的数据大于0，则c列值为True，否则为false
df["c"] = df["a"].apply(lambda x:True if x > 0 else false)

# apply()同样可以执行python内置的函数，例如：返回字符个数
df["d"] = df["b"].apply(len)
```

3.2 DataFrame.apply()

```
# 对数据帧中的所有值加一
df.apply(lambda x:x+1)

# 对所有值都执行平方计算，需要导入numpy库
import numpy as np
df.apply(np.square)

# 如果只要apply作用于指定的行或者列，可以使用name属性进行限定
# 仅对A列
df.apply(lambda x:np.square(x) if x.name=="A" else x)
```

