第1编 Python语法基础

中国人民大学
RENMIN UNIVERSITY OF CHINA

第3讲 函数

余力

buaayuli@ruc.edu.cn

中国人民大学
RENMIN UNIVERSITY OF CHINA

01. 函数基本概念

函数 例子

```
def save(data_pd,file):
    data_pd.to_csv(file[:-4]+DATE_NOW+file[-4:], index=False, encoding='gbk')


def recommend_fund(top_num):
    fd = ft_pd_dict.get("all")
    fd = fd.reindex(columns=["代码", "名称", '综合_排名'])
    fd = fd.sort_values(by='综合_排名')
    fd['综合_排名'] = fd['综合_排名'].map(lambda x:int(x))
    print("推荐基金： ")
    print(fd.head(top_num))


def get_fund_type(code):
    tp=[]
    for fund_type in FUND_TYPE:
        for x in ft_pd_dict[fund_type]["代码"].values:
            x=str(x)
            x="0"*(6-len(x))+x
            if x==code:
                tp.append(fund_type)
    return tp
```

书写格式和定义调用

```
def <name> (arg1, arg2, ... argN):
    <statements>
    return <value>                    #可有可无

def times(x, y):                      #建立函数并做决定
     return x * y                     #回传x*y的值

times(2, 4)                           #小括号里的是自变量
times("No", 4)                        #自变量没有型态之分
```

```
def menu():
    print('''
    系统提供以下功能
    1：添加
    2：删除
    3：修改
    4：搜索
    5：退出
    6: 显示全部联系人信息''')
```

```
people = person()
people.read()
while True:
  try:
    menu()
    choice = int(input('请输入相应数字操作'))
    if choice==1:
        people.add()
    elif choice ==2:
        people.dele()
    elif choice ==3:
        people.modify()
    elif choice ==4:
        people.search()
    elif choice ==5:
        people.write()
        break
    elif choice==6:
        people.show()
    else:
        print('输入不合法，请输入合法数字')
  except ValueError:
     print('请输入数字选项')
```

全局变量与局部变量

```
X = 99

def func(Y):

    Z = X + Y

    return Z

func(1)
```

```
X = 99

def fun():

    X=X+1

    return X

print(fun())
```

```
X = 99

def fun():

    Z=X+1

    return Z

fun()
```

return多个值

```
def multiple(x, y)
    x = 2                          #只改变区域名称
    y = [3, 4]
    return x, y                    #传回新值(tuple)

X = 1
L = [1, 2]
X, L = multiple(X, L)              #结果回传给自己
X, L
```

函数参数传递

```
def func(x, y, a=0, b=0):
        print (x, y, a, b)
func(1, 2, 3, 4)
func(1, 2)
func(1, b=1, y=0)
func(x=1, y=0)
func(a=1, y=2, x=3)
def request():
print( “good” ,end= “\n” ,sep= “” )
```

中国人民大学
RENMIN UNIVERSITY OF CHINA

02. 几个重要函数

匿名函数：lambda

- 书写格式
  - name = lambda arg1, … argN : Expr. using args
- lamda是表达式, 不是语句
  - 可以内嵌在主体程序内
  - 回传一个函数给赋值的函数名称
- lamda是一行的表达式, 不是区段的语句
  - 类似return的结构,
  - 比函数简单
- 小函数的必备良药

lambda-一些范例

```
def func(x, y, z):  return x+y+z
func(2, 3, 4)
func = lamda(x, y, z): x+y+z
func(2, 3, 4)
```

```
def func(x, y, z):
    return x+y+z
print (func(2,3,4))       #→ 9
func = lambda x,y,z: x+y+z
print ( func(2,3,4) )     #→ 9
```

```
L = [lambda x: x**2, lambda y: y**3, lambda z: z**4]
for f in L:
    print (f(2))
print (L[0](3))       #→ 9
```

map()

map(func, *iterables)-->map object

- 功能：对多个序列的每个元素都执行相同的操作，并返回一个map对象。

map()

```
counts = [1, 2, 3, 4]
new = [ ]
for x in counts:
        new.append(x+10)
new
```

```
def inc(x):
        return x + 10
counts = [1, 2, 3, 4]
list(map(inc, counts))
list(map((lambda x: x+10), counts))
```

```
code_name["名称"] = code_name["名称"].map(lambda x: x.replace("₩t", ""))
```

filter()

- 根据返回结果是否为真，对某个序列进行过滤

```
def func(x):
    if  x>0 :   return x
list( filter(func,range(-9,10)) )
```

[1, 2, 3, 4, 5, 6, 7, 8, 9]

```
def func(x):
    if  x>0 :   return x
list  (map(func,range(-9,10)))
```

[None, None, None, None, None, None, None, None, None, 1, 2, 3, 4, 5, 6, 7, 8, 9]

```
def func(x):
    return x+2
list( filter(func,range(-9,10)) )
```

[-9, -8, -7, -6, -5, -4, -3, -1, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

reduce()

- reduce(func, list[,initail])-> value
- reduce()对list每个元素反复调用函数f，并返回最终结果值

```
from functools import reduce

def add(x, y):
    return x + y
print(reduce(add, [1, 3, 5, 7, 9]))
```

f(1, 3)，结果为4；

f(4, 5)，结果为9；

f(9, 7)，结果为16；

f(16, 9)，结果为25

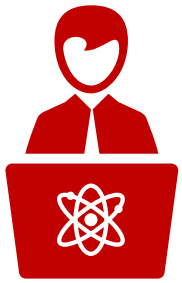
reduce()

```
def add(x,y):
    return x+y
print (reduce(add, range(0,10)))      #→ 45
print (reduce(add, range(0,10),10))   #→ 55
print (reduce(add, range(0,0),10))    #→ 10
```

```
my_list = [3, 1, 5, 4, 10]

# 元素全加1，结果：[4, 2, 6, 5, 11]
list(map(lambda i:i+1, my_list))

# 过滤小于10的元素，结果：[3, 1, 5, 4]
list(filter(lambda i:i<10, my_list))

# 元素累加，结果：33
from functools import reduce
reduce(lambda i,j:i+j, my_list, 10)


# 字典按值排序，结果：[('b', 1), ('a', 3), ('d', 4), ('c', 5)]
my_dict = {'a':3, 'b':1, 'c':5, 'd':4}
sorted(my_dict.items(), key=lambda item: item[1])
```

```
# "基金名称 "   "基金代码 "   "基金币种 "   "收费方式 "   "总份额 "   "可用份数 "
# "成本净值   "   "最新净值 "   "公布日期 "   "当前市值 "   "浮动盈亏 "   "浮动收益率(%) "


df = pd.read_excel(file, sheet_name="bank", encoding='gbk')
df.columns = df.columns.map(lambda x: x.replace("\t", ""))
df.columns = df.columns.map(lambda x: x.replace("基金代码", "代码"))
df.columns = df.columns.map(lambda x: x.replace("基金名称", "名称"))
df.columns = df.columns.map(lambda x: x.replace("当前市值", "资产JT"))
df.columns = df.columns.map(lambda x: x.replace("浮动盈亏", "盈亏JT"))
df.columns = df.columns.map(lambda x: x.replace("浮动收益率(%)", "浮动收益JT"))
```

中国人民大学
RENMIN UNIVERSITY OF CHINA



谢谢大家！