

什么是高阶函数？

高阶函数：一个函数可以作为参数传给另外一个函数，或者一个函数的返回值为另外一个函数（若返回值为该函数本身，则为递归），满足其一则为高阶函数。

参数为函数

Python

```
1 #参数为函数
2 def bar():
3     print("in the bar..")
4 def foo(func):
5     func()
6     print("in the foo..")
7
8 foo(bar)
```

返回值为函数

Python

```
1 #返回值为函数
2 def bar():
3     print("in the bar..")
4 def foo(func):
5     print("in the foo..")
6     return bar
7 res=foo(bar)
8 res()
```

以上两个示例中，函数foo()为高阶函数，示例一中函数bar作为foo的参数传入，示例二中函数bar作为foo的返回值。

注：函数名（例如bar、foo）-->其为该函数的内存地址；函数名+括号（例如bar()、foo()）-->调用该函数。

高阶函数-map、filter、reduce

这三个函数均为高阶函数，其也为Python内置的函数。接下来我们看一下这三个函数的用法以及其内部原理是怎样的：

map函数

map函数接收的是两个参数，一个函数，一个序列，其功能是将序列中的值处理再依次返回至列表内。其返回值为一个迭代器对象--》例如：<map object at 0x00000214EEF40BA8>。其用法如图：

接下来我们看一下map函数的机制是怎么样的：

Python

```
1 num=[1,2,3,4,5]
2 def square(x):
3     return x**2
4 #map函数模拟
5 def map_test(func,iter):
6     num_1=[]
7     for i in iter:
8         ret=func(i)
9         # print(ret)
10        num_1.append(ret)
11    return num_1.__iter__() #将列表转为迭代器对象
12
13 #map_test函数
14 print(list(map_test(square,num)))
15 #map函数
16 print(list(map(square,num)))
17
18 #当然map函数的参数1也可以是匿名函数、参数2也可以是字符串
19 print(list(map_test(lambda x:x.upper(),"amanda")))
20 print(list(map(lambda x:x.upper(),"amanda")))
```

filter函数

filter函数也是接收一个函数和一个序列的高阶函数，其主要功能是过滤。其返回值也是迭代器对象，例如：<filter object at 0x000002042D25EA90>，其图示如下：

接下来我们看一下filter函数的用法以及其机制是怎么样的：

Python

```
1 names=["Alex","amanda","xiaowu"]
2 #filter函数机制
3 def filter_test(func,iter):
4     names_1=[]
5     for i in iter:
6         if func(i): #传入的func函数其结果必须为bool值，才有意义
7             names_1.append(i)
8     return names_1
9 #filter_test函数
10 print(filter_test(lambda x:x.islower(),names))
11 #filter函数
12 print(list(filter(lambda x:x.islower(),names)))
```

reduce函数

reduce函数也是一个参数为函数，一个为可迭代对象的高阶函数，其返回值为一个值而不是迭代器对象，故其常用与叠加、叠乘等，图示例如下：

实例如下：

Python

```
1 #reduce函数不是内置函数，而是在模块functools中的函数，故需要导入
2 from functools import reduce
3
4 nums=[1,2,3,4,5,6]
5 #reduce函数的机制
6 def reduce_test(func,array,ini=None): #ini作为基数
7     if ini == None:
8         ret =array.pop(0)
9     else:
10         ret=ini
11     for i in array:
```

```
12         ret=func(ret,i)
13     return ret
14 #reduce_test函数，叠乘
15 print(reduce_test(lambda x,y:x*y,nums,100))
16 #reduce函数，叠乘
17 print(reduce(lambda x,y:x*y,nums,100))
```