

Python复习2 闭包和高阶函数

1. 闭包

在 Python 中，闭包是指包含了对外部（封闭的）作用域变量引用的函数。具体来说，当一个函数在其内部定义了另一个函数，并且内部函数引用了外部函数的变量时，就创建了一个闭包。

闭包三要素：1. 函数的嵌套；2. 内层函数使用外层函数的变量；3. 外层函数返回内层函数的引用。

下面是一个简单的闭包示例：

```
1 def outer_function(x): # 外部函数
2     def inner_function(y): # 内部函数
3         return x + y
4     return inner_function
5
6 closure = outer_function(10)
7 result = closure(5)
8
9 print(result) # 输出 15
10
```

2. 推导式

详细解释：

```
[name.upper() for name in list if len(name)>3]
```

3

1

2

首先执行第1步，从list中遍历获取每一个列表项；

然后将拿取到的列表项进行if条件判断；

如果满足if条件，那么执行第3步，否则不做任何事情。

CSDN @风赢十三水

3. 匿名函数

匿名函数也被称为lamda函数。

```
1 add = lambda x, y: x + y
2 result = add(3, 5)
3 print(result) # 输出 8
```

4. 高阶函数

高阶函数是指接受其他函数作为参数或者返回函数作为结果的函数。

- sorted()

```
1 words = ["banana", "apple", "kiwi", "orange"]
2 sorted_words = sorted(words, key=lambda x: len(x))
3 print(sorted_words) # 输出 ['kiwi', 'apple', 'banana', 'orange']
```

```
1 list_tuples = [(1, 'byd'), (3, 'xiaopeng'), (2, 'tesla'), (4, 'weilai')]
2
3 listed_tuples = sorted(list_tuples, key=lambda x: x[1])
4
5 print(listed_tuples)
6 输出: [(1, 'byd'), (2, 'tesla'), (4, 'weilai'), (3, 'xiaopeng')]
```

- map()

- map(function_to_apply, list_of_inputs)

- function_to_apply: 代表函数
 - list_of_inputs: 代表输入序列

- 注意: Python3中, map函数返回的是迭代器

```
1 items = [1, 2, 3, 4, 5]
2 def f(x):
3     return x**2
4 squared = list(map(f, items))
```

```
1 items = [1, 2, 3, 4, 5]
2
3 squared = list(map(lambda x: x**2, items))
```

- `reduce()`

- **`reduce(function, iterable[, initializer])`**

- **function: 代表函数**

- **iterable: 序列**

- **initializer: 初始值 (可选)**

- **与map不同，reduce需要用from functools import reduce导入，才可使用**

```
1 # 10的阶乘
2 from functools import reduce
3 def f(x,y):
4     return x*y
5 items = range(1,11)
6 result=reduce(f,items)
```

- `filter()`

`filter` 函数用于筛选可迭代对象中满足指定条件的元素，返回一个新的可迭代对象。

```
1 # 过滤出列表中的所有奇数
2 def is_odd(n):
3     return n % 2 == 1
4 f_list = filter(is_odd,[1,2,3,4,5,6,7,8,9,10])
```