python

function

张海宁

贵州大学

hnzhang1@gzu.edu.cn

March 17, 2019

Overview

introduce of function

函数（方法）具有以下特点：

- 是一段代码
- 能完成特定功能
- 可以在其他地方被调用
- 可以接收参数或不接收参数
- 可以有返回值或没有返回值

函数用于将一个复杂的问题分解为若干个简单的子问题。

函数的类型

Table: 函数的类型

类型	说明	使用方式
内建函数	print()、input() 等	直接调用
标准库函数	如 math 库里的函数等	先导入，再调用
自定义函数	用户自己按需编写的函数	先定义，再调用
第三方库函数	其他人编写的函数	先导入，再调用

内建函数

内建函数是指定义在 builtins 模块中的函数，这里面的函数都可以直接使用。

标准库函数

标准库函数是 python 语言中自带的一些模块中的函数，这里面的函数都可以先通过 import 语句导入模块，然后再通过 < 库名 >.< 函数名 > 的形式来使用。以下列出几个标准库，更多的标准库请参阅官方文档：

`https://docs.python.org/3/library/index.html`。

math	提供数学计算功能。'ceil', 'cos', 'degrees', 'e', 'exp', 'floor', 'gcd', 'log', 'log10', 'log2', 'pi', 'pow', 'radians', 'remainder', 'sqrt'
os	系统交互功能。'getcwd', 'open', 'read', 'write'
random	提供生成随机数功能。'randint', 'random', 'randrange', 'sample', 'seed'
io	提供输入输出相关的功能，如读写文件。

使用标准库函数

import

```
1 import random
2
3 r = random.randint(1, 38)
4 print(r)
```

from···import···

```
1 from random import randint
2
3 r = randint(1, 38)
4 print(r)
```

自定义函数

自定义函数是用户根据自己的实际需求而编写的函数。

第三方库函数

python 拥有大量的第三方库函数，第三方库函数是 python 的特色之一。这里面的函数都可以先通过 import 语句导入，然后再通过 < 库名 >.< 函数名 > 的形式来使用。

Table: 著名的第三方库

类型	说明
Django	开源 Web 开发框架
Tornado	一个轻量级的 Web 框架
Matplotlib	主要用来绘制高质量的二维数学图形。
SciPy	基于 Python 的 matlab 实现。
NumPy	基于 Python 的科学计算第三方库。
PyGtk	基于 Python 的 GUI 程序开发 GTK+ 库。
PyQt	用于 Python 的 QT 开发库。
BeautifulSoup	基于 Python 的 HTML/XML 解析器。

函数的定义

```
1  def functionName(parameter1, parameter2, ···):
2      functionBlock
3      return aValue
```

第 1 行称为函数签名，用于指定函数名称以及函数的每个形式参数变量名，注意要有结尾的英文冒号。

- def　定义函数的关键字
- functionName　函数名
- parameter1···　参数[1]
- functionBlock　函数体
- return　表明此函数是有返回值的[2]
- aValue　函数的返回值

[1]函数可以不需要参数
[2]函数可以不返回任何值

函数的调用

```
1  import math
2
3  def myGCD(a,b):
4      c, d = a, b
5      while a % b != 0:
6          a, b = b, a % b
7      print(c,' 和',d,' 的最大公约数是:',b)
8
9  a = int(input("a:"))
10 b = int(input("b:"))
11 myGCD(a,b)
12 print(math.gcd(a, b))
```

一系列的import

一系列的函数定义

任意数量的全局代码

程序的主体

默认 (可选) 参数

可以给一个参数指定默认值 (在函数签名中为参数赋值), 这样的参数也叫可选参数。可选参数可以有多个，但所有的可选参数都要定义在必选参数之后。

```
def is_prime(first_n=10):
    prime_list = [2]
    i = 3
    while True:
        for d in range(2, i):
            if i % d == 0:
                break
            elif (i % d != 0) and (d == (i-1)):
                prime_list.append(i)
        if len(prime_list) == first_n:
            break
        i += 1
    return prime_list

print(is_prime())
print(is_prime(5))
```

多态性

```
1  import math
2
3  def my_power(a, b):
4      return a**b
5
6  n1 = my_power(2,5)
7  n2 = my_power(math.pi, 1)
8  print(n1, n2)
9
10
11 32 3.141592653589793
```

从 *my_power()* 这个函数的返回值可以看出，传递 int 值的时候，返回的也是 int 值；如果传递的是浮点类型的值，返回值就是浮点类型的。

python 的这个特性体现了其灵活性，即多态性。多态性允许定义一个函数，供不同类型的对象使用。

但是要注意传递恰当类型的对象，如果向 *my_power()* 函数传递字符串类型的对象的话，程序会产生 *TypeError* 错误。

返回多个值

```
1  import math
2
3  def move(x, y, distance, angle = 0):
4      new_x = x + distance * math.cos(angle)
5      new_y = y + distance * math.sin(angle)
6      return new_x, new_y
7
8  x, y = move(0, 0, 1.414, math.pi/4)
9  print(x, y)
10 z = move(0, 0, 2, math.pi/6)
11 print(z)
12
13 0.9998489885977783 0.99984898885977778
14 (1.732050807568874, 0.9999999999999999)
```

第 8 行形式上返回了两个值，通过第 10 可以看出其实是一个元组。

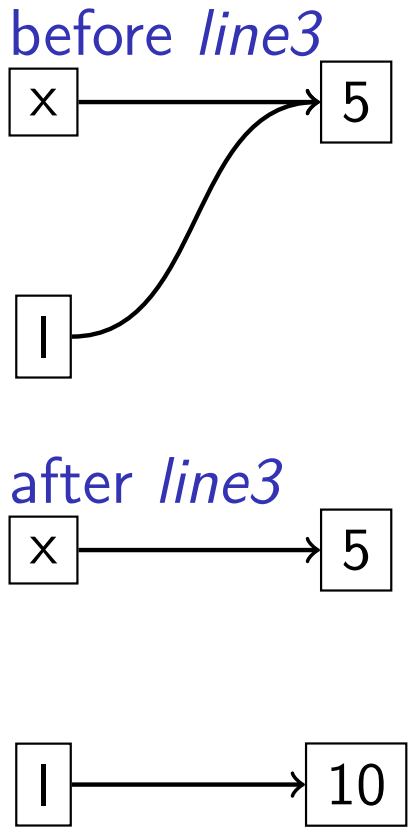
列表解析

当想要对一个序列中的每个元素都执行某个特定的操作时，通常可以使用 *for* 循环来实现。在 *python* 中，可以使用列表解析。

```
1  import math
2
3  l = [2, 3, 4]
4  m = [pow(x, 2) for x in l]
5  n = [math.sqrt(y) for y in m]
6  print(m)
7  print(n)
8
9  [4, 9, 16]
10 [2.0, 3.0, 4.0]
```

向函数传递不可变对象[3]

```
1  def double(l):
2      print('before:: l=',l,'id(l)=',id(l))
3      l = l * 2
4      print('after:: l=',l,'id(l)=',id(l))
5
6  x = 5
7  print('x=',x,'id(x)=:', id(x))
8  double(x)
9  print('x=',x)
10
11 x= 5 id(x)=: 4385389696
12 before:: l= 5 id(l)= 4385389696
13 after:: l= 10 id(l)= 4385389856
14 x= 5
```

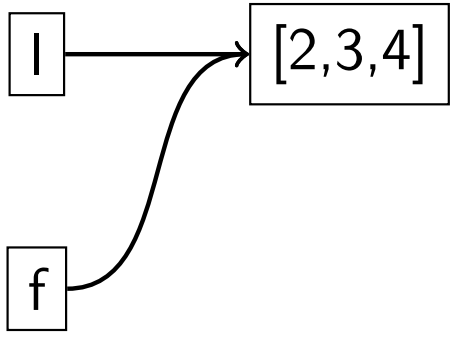


[3] https://www.python-course.eu/passing_arguments.php

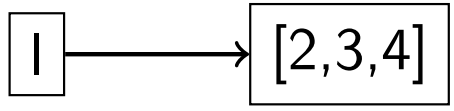
向函数传递可变对象 I

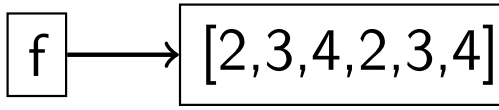
```
1  def double(f):
2      print('before:: f=',f,'id(f)=',id(f))
3      f = f * 2
4      print('after:: f=',f,'id(f)=',id(f))
5
6  l = [2, 3, 4]
7  print('l=',l,'id(l)=:', id(l))
8  double(l)
9  print('l:', l)
10
11 l= [2, 3, 4] id(l)=: 4386889608
12 before:: f= [2, 3, 4] id(f)= 4386889608
13 after:: f= [2, 3, 4, 2, 3, 4]
14 id(f)= 4388025352
15 l: [2, 3, 4]
```

before *line3*



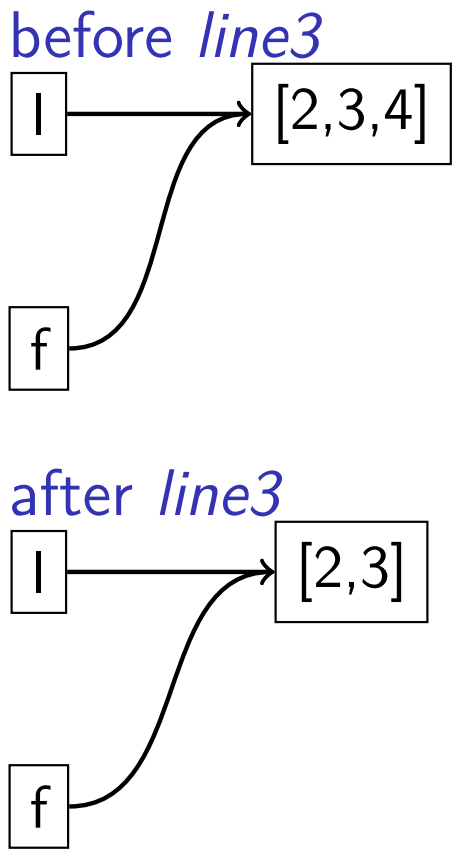
after *line3*





向函数传递可变对象 II[4]

```
1  def double_2(f):
2      print('before:: f=',f,'id(f)=',id(f))
3      del f[-1]
4      print('after:: f=',f,'id(f)=',id(f))
5
6  l = [2, 3, 4]
7  print('l=',l,'id(l)=',id(l))
8  double_2(l)
9  print('l:', l)
10
11 l= [2, 3, 4] id(l)= 4377313160
12 before:: f= [2, 3, 4] id(f)= 4377313160
13 after:: f= [2, 3] id(f)= 4377313160
14 l: [2, 3]
```

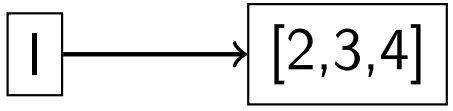


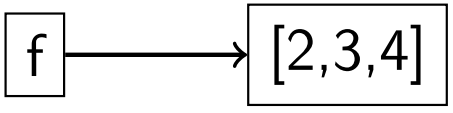
[4]Side effects: A function is said to have a side effect if, in addition to producing a value, it modifies the caller's environment in other ways.

如何避免函数改变主程序中变量的值

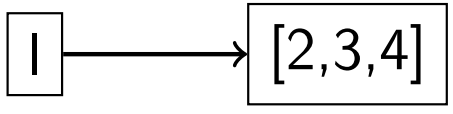
```
1  def double_2(f):
2      print('before:: f=',f,'id(f)=',id(f))
3      del f[-1]
4      print('after:: f=',f,'id(f)=',id(f))
5
6  l = [2, 3, 4]
7  print('l=',l,'id(l)=',id(l))
8  double_2(l[:])
9  print('l:', l)
10
11 l= [2, 3, 4] id(l)= 4537892744
12 before:: f= [2, 3, 4] id(f)= 4539028552
13 after:: f= [2, 3] id(f)= 4539028552
14 l: [2, 3, 4]
```

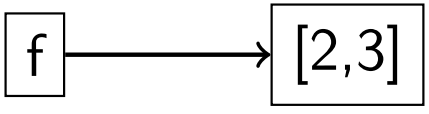
before *line3*





after *line3*





lamda

模块

模块，就是一个特殊的 *python* 文件，其特殊在：

- 包含了若干函数
- 这些函数可供其他 *python* 程序调用
- 没有全局代码

自定义模块

toolbox.py

```
1  import math
2  import random
3  import sys
4
5  def my_gcd(a, b):
6      return math.gcd(a, b)
7  def my_rand():
8      return random.randint(1, 9)
9  def main():
10     print('gcd(',sys.argv[1],',',sys.argv[2],')=',
11           my_gcd(int(sys.argv[1]), int(sys.argv[2])))
12     for i in range(3):
13         print(my_rand())
14 if __name__ == '__main__':
15     main()
```

引用自定义模块

test.py

```
1 import toolbox
2
3 print(toolbox.my_rand())
```

API

API, 应用程序编程接口，描述了模块所实现的具体功能。API 保证了代码的重用性，使得其他用户不用查看模块的具体定义就可以直接使用其实现的功能（函数）。用户自定义的模块也应给出相应的 API 以供参考。
toolbox 模块的 API 如下表所示：

Table: toolbox 模块的 API

函数调用	功能描述
my_gcd(a, b)	返回 a, b 两个数的最大公约数
my_rand()	返回一个 [1, 9] 之间的随机整数

Homework

1. 定义一个函数 *fact(n)*，用以求一个正整数 *n* 的阶乘。
2. 使用辗转相除法[5]求任意两个数的最大公约数。

[5]用较大数除以较小数，再用出现的余数（第一余数）去除除数，再用出现的余数（第二余数）去除第一余数，如此反复，直到最后余数是 0 为止。如果是求两个数的最大公约数，那么最后的除数就是这两个数的最大公约数。

Q&A