

Vamei

编程, 数学, 设计

博客园 首页 订阅 管理

随笔-209 文章-1 评论-3802

Python基础07 函数

作者: Vamei 出处: <http://www.cnblogs.com/vamei> 欢迎转载, 也请保留这段声明。谢谢!

函数最重要的目的是方便我们重复使用相同的一段程序。

将一些操作隶属于一个函数, 以后你想实现相同的操作的时候, 只用调用函数名就可以, 而不需要重复敲所有的语句。

函数的定义

首先, 我们要定义一个函数, 以说明这个函数的功能。

```
def square_sum(a,b):  
    c = a**2 + b**2  
    return c
```

这个函数的功能是求两个数的平方和。

首先, def, 这个关键字通知python: 我在定义一个函数。square_sum是函数名。

括号中的a, b是函数的参数, 是对函数的输入。参数可以有多个, 也可以完全没有(但括号要保留)。

我们已经在循环和选择中见过冒号和缩进来表示的隶属关系。

```
c = a**2 + b**2           # 这一句是函数内部进行的运算  
  
return c                  # 返回c的值, 也就是输出的功能。Python的函数  
                           允许不返回值, 也就是不用return。
```

return可以返回多个值, 以逗号分隔。相当于返回一个tuple(定值表)。

```
return a,b,c          # 相当于 return (a,b,c)
```

在Python中，当程序执行到return的时候，程序将停止执行函数内余下的语句。return并不是必须的，当没有return，或者return后面没有返回值时，函数将自动返回None。None是Python中的一个特别的数据类型，用来表示什么都没有，相当于C中的NULL。None多用于关键字参数传递的默认值。

函数调用和参数传递

定义过函数后，就可以在后面程序中使用这一函数


```
print square_sum(3,4)
```

Python通过位置，知道3对应的是函数定义中的第一个参数a，4对应第二个参数b，然后把参数传递给函数square_sum。

（Python有丰富的参数传递方式，还有关键字传递、表传递、字典传递等，基础教程将只涉及位置传递）

函数经过运算，返回值25，这个25被print打印出来。

我们再看下面两个例子



```
a = 1

def change_integer(a):
    a = a + 1
    return a

print change_integer(a)
print a

#==(Python中 "#" 后面跟的内容是注释，不执行 )

b = [1,2,3]
```

```
def change_list(b):  
    b[0] = b[0] + 1  
    return b  
  
print change_list(b)  
print b
```



第一个例子，我们将一个整数变量传递给函数，函数对它进行操作，但原整数变量a不发生变化。

第二个例子，我们将一个表传递给函数，函数进行操作，原来的表b发生变化。

对于基本数据类型的变量，变量传递给函数后，函数会在内存中复制一个新的变量，从而不影响原来的变量。（我们称此为值传递）

但是对于表来说，表传递给函数的是一个指针，指针指向序列在内存中的位置，在函数中对表的操作将在原有内存中进行，从而影响原有变量。（我们称此为指针传递）

总结

```
def function_name(a,b,c):  
    statement  
    return something # return不是必须的
```

函数的目的： 提高程序的重复可用性。

return None

通过位置，传递参数。

基本数据类型的参数：值传递

表作为参数：指针传递

练习：

写一个判断闰年的函数，参数为年、月、日。若是闰年，返回True

标签: [Python](#)

好文要顶

关注我

收藏该文



Vamei

关注 - 26

粉丝 - 4985

荣誉: [推荐博客](#)

[+加关注](#)

19

0

(请你对文章做出评价)

« 上一篇: [Python基础06 循环](#)

» 下一篇: [Python基础08 面向对象的基本概念](#)

posted @ 2012-06-01 10:41 Vamei 阅读(56471) 评论(85) 编辑 收藏

< Prev

1

2

评论列表

#51楼 2014-08-02 15:34 lovelian

```
1 def runnian(year):
2     if((year%4==0 and year%100!=0) or year%400==0):
3         return True
4     return False
```

支持(0) 反对(1)

#52楼 2014-08-21 10:33 zjl-1110

```
1 #!/usr/bin/python
2 #-*-coding:utf-8-*-
3 def run_y(i):
4     if i%400==0 and i%4==0 or i%100!=0:
5         print("yes")
6     else:
7         print("no")
8
9 run_y(2014)
```

支持(0) 反对(2)

#53楼 2014-08-21 10:35 zjl-1110

额，算闰年，月和日的参数有必要吗？？

支持(1) 反对(1)

#54楼[楼主] 2014-08-21 11:03 Vamei

@ zjl-1110

貌似没有啊，我给修改一下。

支持(0) 反对(0)

#55楼 2014-11-19 21:35 珩儿

@ Vamei

可以有呀~~如果月和日刚好是2月28日就可以直接返回是闰年啦~

支持(0) 反对(2)

#56楼 2014-11-19 21:36 珩儿

```
1 >>> def runnian(year,month,day):
2     if (month==2 and day==28):
3         return '是闰年'
4     elif ((year%100==0 and year%400==0) or (year%100!=0 and year%4==0)):
5         return '是闰年'
6     else:
7         return '不是闰年呀~'
```

支持(1) 反对(1)

#57楼 2014-12-10 16:00 小静 (Cathy)

```
1 def func(year):
2     return year%400==0 or (year%4==0 and year%100!=0)
```

学习了。感觉python的语法真是简洁，而且函数还能直接返回多个参数

支持(4) 反对(0)

#58楼 2014-12-15 13:24 ly.bing

```
1 def isLeapYear(b):
2     year = int(b[0])
3     print year, type(year)
4     if year%4==0 and year%100<>0 or year%400==0:
5         return True
6     else:
7         return False
8
```

```
9
10 print isLeapYear([2014,12,15]) #False-不是闰年
11 print isLeapYear([2000,12,15]) #True -是闰年
```

支持(0) 反对(0)

#59楼 2015-02-08 12:28 地主属兔

```
1 def isLeap(year):
2     if year%100==0:
3         return year%400==0
4     else:
5         return year%4==0
```

博主你的文章太好了，适合快速学习！支持你~

支持(0) 反对(0)

#60楼 2015-02-13 22:20 ArtinHuang

@ 珩儿

每年都有2月28日.....应该改为2月29日，闰年才有2月29日

支持(0) 反对(0)

#61楼 2015-02-13 22:26 ArtinHuang

```
1 # 普通年能被4整除且不能被100整除的为闰年。（如2004年就是闰年,1901年不是闰年）
2 # 世纪年能被400整除的是闰年。
3
4 def leafyear(year,month,day):
5     if (month==2 and day==29):
6         return True
7     elif ((year%400==0) or (year%100!=0 and year%4==0)):
8         return True
9     else:
10        return False
```

支持(0) 反对(0)

#62楼 2015-03-02 17:15 EZ_lzh

问一个问题，如果想要函数能够改变原整数变量该怎么办？

支持(0) 反对(0)

#63楼 2015-03-11 14:16 pshaoshuai

```
def rn(a):
import string
s='%d' %a
t=s[0:4]
x=string.atoi(t)
if((x%4==0 and x%100!=0)or x%400==0):
print (a,'所在年份是闰年')
else:
print (a,'所在年份不是闰年')
为什么运行后输入rn(20141028)
返回错误:
Traceback (most recent call last):
File "<pyshell#55>", line 1, in <module>
rn(20140101)
File "C:\Users\ThinkPad\Desktop\rn.py", line 5, in rn
x=string.atoi(t)
AttributeError: 'module' object has no attribute 'atoi'
```

支持(0) 反对(0)

#64楼 2015-03-23 16:21 pshaoshuai

```
1  def rn(a):
2      if type(a)is not int:
3          print('输入的年月日错误')
4      elif a>99999999:
5          print('输入的年月日错误')
6      else:
7          s='%d' %a
8          t=s[0:4]
9          x=int(t)
10         if((x%4==0 and x%100!=0)or x%400==0):
11             print (a,'所在年份是闰年')
12         else:
13             print (a,'所在年份不是闰年')
14  原来可以直接用int ()
```

支持(0) 反对(0)

#65楼 2015-03-23 16:41 pshaoshuai

@ EZ_lzh

改变原整数变量是什么意思？改变变量类型还是变量值？

支持(0) 反对(0)

#66楼 2015-03-30 14:10 pythonerJ

感谢博主，新手写的可能有点丑

```
#!/usr/bin/python
def leapyear(year):
    if year % 400 == 0:
        return True
    elif year % 4 == 0 and year % 100 != 0:
        return True
    else:
        return False
```

支持(0) 反对(0)

#67楼 2015-05-08 00:21 dotdog

不一定只是为了“重复使用”

支持(0) 反对(0)

#68楼 2015-05-09 14:41 NG魂

```
1  def isRNA(year):
2      if(year%100==0):
3          return year%400==0
4      else:
5          return year%4==0
6
7  def isRNB(year):
8      return year%100!=0 and year%4==0 or year%400==0
```

同样执行1000*10000次

方法a比方法b快1秒（约4.7和5.7）

支持(0) 反对(0)

#69楼 2015-07-01 16:39 brzone

@ lovelian

其实还可以更短：

直接return (year%4==0 and year%100!=0) or year%400==0

支持(0) 反对(0)

#70楼 2015-07-08 16:34 big tree

```
def isleap(year):
    if(a%100==0):
        if (a%400==0):
            print('run nian')
        else:
            print('No run nian')
    else:
        if(a%4==0):
            print('run nian')
        else:
            print('No run nian')
```

支持(0) 反对(0)

#71楼 2015-07-08 16:35 big tree

```
1  def isleap(year):
2      if(a%100==0):
3          if (a%400==0):
4              print('run nian')
5          else:
6              print('No run nian')
7      else:
8          if(a%4==0):
9              print('run nian')
10         else:
11             print('No run nian')
```

支持(0) 反对(0)

#72楼 2015-08-05 12:21 Nefeltari

```
1  def judgeleapyear(y):
2      if ((int)(y[0:4])%4==0 and (int)(y[0:4])%100!=0) or (int)(y[0:4])%400==0:
3          return y+' is leapyear,True'
4      else:
5          return y+' is leapyear,False'
6
```

```
7 | print judgeleapyear('20150805')
```

支持(0) 反对(0)

#73楼 2015-08-14 16:56 JR2016

博主的文章写的简单易懂,有条理,希望能通过学习掌握python吧

支持(0) 反对(0)

#74楼 2015-08-17 16:01 felord

写一个判断闰年的函数,参数为年、月、日。若是闰年,返回True

这个题目看似简单,我看了评论,想了一下午,都没有一个完备的答案,

1、year 按规定不能等于0,因为公元元年就是公元1年,并没有公元0年,所以就要分year大于0或小于0 两种情况

2、当year大于0,非整百年能被4整除的为闰年,整百年能被400整除为闰年,对于数值很大的年份,这年如果能被3200整除,并且能被172800整除则是闰年。

3、当year小于0,非整百年除以4余1的为闰年,整百年除以400余1为闰年,对于数值很大的年份,这年如果除以3200余1,并且除以172800余1则是闰年

4、另外假设你的参数 month 和 day,通过year的判断是非闰年的话,那么当参数month为2 时,day的取值范围只能是1-28,我不知道 如何修改函数参数的值。并且month为其他值时,day取值范围要在1-30 或 1-31之间变动,不知道这个变动如何体现,比如 4月 不能有31。

请楼主告知详细的做法。

支持(0) 反对(0)

#75楼 2015-10-22 12:58 劉丱

@ Vamei

```
def run_nian(year):
```

```
r = (year % 400 == 0 or (year % 4 == 0 and year % 100 != 0))
```

```
if (r):
```

```
print(year)
```

```
for r in range(2100):
```

```
run_nian(r)
```

支持(0) 反对(0)

#76楼 2015-10-22 16:43 劉丱

```
def run_nian(year):
    r = (year % 400 == 0 or (year % 4 == 0 and year % 100 != 0))
    if (r):
        print(True)
    else:
        print(False)

s = input('year: ')
year = int(s)
run_nian(year)
```

支持(0) 反对(0)

#77楼 2016-01-17 14:17 我的风儿

```
def run(a):
    if (0!=a%100 and 0==a%4 )or 0==a%400:
        print('yes')
    else:
        print('no')
```

支持(0) 反对(0)

#78楼 2016-01-17 14:18 我的风儿

```
1  def run(a):
2      if (0!=a%100 and 0==a%4 )or 0==a%400:
3          print('yes')
4      else:
5          print('no')
```

支持(0) 反对(0)

#79楼 2016-01-31 12:54 Evel'sBlog

```
1  def check_leap_year(y,m,d):
2      if type(y)is int:
3          print "ignore",m,d
4          if (y % 400 == 0) or (y % 4 == 0 and y % 100 != 0):
5              return "leap"
6          else:
7              return "non-leap"
```

```
8         else:
9             return "invalid"
10
11 ytype = check_leap_year(2015,3,4)
12 print "it's a",ytype,"year"
```

对于数值很大的年份?如何定义这个很大? @felord

支持(0) 反对(0)

#80楼 2016-03-26 11:23 hitwh_Gypsy

```
1  #!/usr/bin/env python
2  def isLeapYear(year):
3      if year % 100 == 0:
4          if year % 400 == 0:
5              print True
6          else:
7              print False
8      elif year % 4 == 0:
9          print True
10     else:
11         print False
12
13 print isLeapYear(2016)
```

- 1、第一个判断条件是能否被100整除
- 2、第二个判断条件是能否被4整除

支持(0) 反对(0)

#81楼 2016-03-26 11:24 hitwh_Gypsy

```
1  #!/usr/bin/env python
2  def isLeapYear(year):
3      if year % 100 == 0:
4          if year % 400 == 0:
5              print True
6          else:
7              print False
8      elif year % 4 == 0:
9          print True
```

```
10     else:
11         print False
12
13 print isLeapYear(2016)
```

- 1、第一个判断条件是能否被100整除
- 2、第二个判断条件是能否被4整除

支持(0) 反对(0)

#82楼 2016-03-26 11:27 hitwh_Gypsy

```
1  #!/usr/bin/env python
2  def isLeapYear(year):
3      if year % 100 == 0:
4          if year % 400 == 0:
5              print True
6          else:
7              print False
8      elif year % 4 == 0:
9          print True
10     else:
11         print False
12
13 print isLeapYear(2016)
```

- 1、第一个判断条件是能否被100整除
- 2、第二个判断条件是能否被4整除

支持(0) 反对(0)

#83楼 2016-04-21 17:25 jackchen77

@ zjl-1110

当然，这道题是比较结合实际情况的

支持(0) 反对(0)

#84楼 2016-04-21 17:37 jackchen77

```
def ryear(year):
    if not isinstance(year,int):
        raise TypeError('传递参数格式不正确')
    elif year%400==0:
        return True
    elif ((year%4==0) & (year%100 !=0)):
        return True
    else:
        return False

print(ryear(2100))
```

支持(0) 反对(0)

#85楼 2016-04-21 17:38 jackchen77

```
1  def ryear(year):
2      if not isinstance(year,int):
3          raise TypeError('传递参数格式不正确')
4      elif year%400==0:
5          return True
6      elif ((year%4==0) & (year%100 !=0)):
7          return True
8      else:
9          return False
10
11  print(ryear(2100))
```

支持(0) 反对(0)

< Prev

1

2

[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

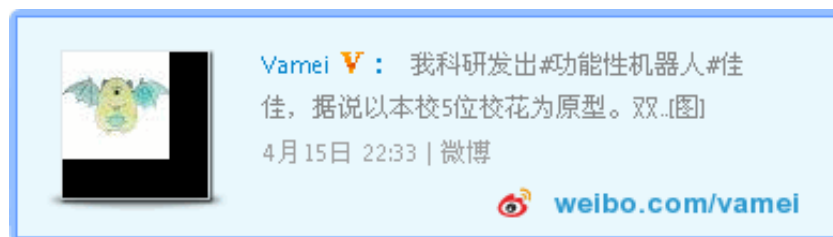
【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库

【推荐】融云即时通讯云—豆果美食、Faceu等亿级APP都在用



公告

你好，这里是Vamei，一名编程爱好者。我在博客里写了**Python/Linux/网络协议/算法/Java/数据科学**系列文章，从这里开始阅读。非常期待和你的交流。



我的微博

下列教程已经做成电子出版物，内容经过修订，也方便离线阅读：
协议森林

欢迎阅读我写的其他书籍：

现代小城的考古学家

天气与历史的相爱相杀

随手拍光影

昵称：Vamei

园龄：4年1个月

荣誉：推荐博客

粉丝：4985

关注：26

+加关注

[常用链接](#)

[我的随笔](#)

[我的评论](#)

[我的参与](#)

[最新评论](#)

[我的标签](#)

[我的标签](#)

[Python\(61\)](#)

[Java\(42\)](#)

[大数据\(22\)](#)

[Linux\(17\)](#)

[网络\(16\)](#)

[算法\(15\)](#)

[文青\(14\)](#)

[技普\(9\)](#)

[系列索引\(6\)](#)

[开发工具\(4\)](#)

[更多](#)

[系列文章](#)

[Java快速教程](#)

[Linux的概念与体系](#)

[Python快速教程](#)

[数据科学](#)

[协议森林](#)

[纸上谈兵：算法与数据结构](#)

[积分与排名](#)

[积分 - 659668](#)

[排名 - 122](#)

[最新评论](#)

[1. Re:Java基础11 对象引用](#)

[受教！](#)

--MissLost

[2. Re:Python快速教程](#)

看评论区一片喝彩！看来我得在此扎营了！

--测试小蚂蚁

3. Re:Python进阶06 循环对象

好好地列表解析变成了表推导

--ashic

4. Re:"不给力啊，老湿！": RSA加密与破解

感谢楼主精彩分享

--worldball

5. Re:概率论04 随机变量

你写的这一系列太棒了，刚加入博客园就在你这里学到了，我要转载过去学习一下

--yixius

6. Re:Python基础03 序列

挺好的教程、、、

--王小拽的号

7. Re:Python进阶07 函数对象

```
def func(x,y): print x**ydef test(f,a,b): print 'test' print f(a,b)test (func,3,2)
```

输出的内容:tes.....

--M-edea

8. Re:Python进阶02 文本文件的输入输出

@coderXT换行符: \n...

--行者之印

9. Re:数据科学

博主啊，这里是一枚即将大二的计算机新人，大一学了python，java，还有一些算法，数据结构，图论了，感觉我对数学又一些反感，但是听说离散数学对计算机专业的很重要，不知道怎么去学比较好呢，我想像您写.....

--Acokil

10. Re:为什么要写技术博

楼主是用自己自定义的模板吗？在博客园里找不到这种风格的blog模板？

--行者之印

11. Re:来玩Play框架01 简介

挖煤哥,我补充了一下Windows下的搭建play框架,希望有点帮助,谢谢!

--Sungeek

12. Re:来玩Play框架07 静态文件

```
@helper.form(action = routes.Application.upload, 'enctype ->
```

```
"multipart/form-data") {--action = rout.....
```

--quxiaozha

13. Re:来玩Play框架07 静态文件

该记录将/assets/下的URL，对应到项目的/public文件夹内的文件。比如在项目的/public/images/test.jpg，就可以通过/assests/images/test.jpg这一.....

--quxiaozha

14. Re:来玩Play框架06 用户验证

支持挖煤哥~~~

--quxiaozha

15. Re:“不给力啊，老湿！”：RSA加密与破解

@maanshancss请你仔细阅读了这个文章再来评价。...

--Vamei

推荐排行榜

1. “不给力啊，老湿！”：RSA加密与破解(218)
2. Python快速教程(140)
3. 野蛮生长又五年(91)
4. Java快速教程(88)
5. 协议森林01 邮差与邮局 (网络协议概观)(79)
6. 为什么要写技术博(71)
7. 编程异闻录(54)
8. 博客一年：心理之旅(49)
9. 协议森林08 不放弃 (TCP协议与流通信)(45)
10. Python快速教程 尾声(43)
11. 协议森林(42)
12. Java基础01 从HelloWorld到面向对象(42)
13. Python基础08 面向对象的基本概念(40)
14. 一天能学会的计算机技术(34)
15. 博客第二年，杂谈(33)

Copyright ©2016 Vamei

05370247