

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进阶05 循环设计

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

在“循环”一节，我们已经讨论了Python基本的循环语法。这一节，我们将接触更加灵活的循环方式。

range()

在Python中，for循环后的in跟随一个序列的话，循环每次使用的序列元素，而不是序列的下标。

之前我们已经使用过range()来控制for循环。现在，我们继续开发range的功能，以实现下标对循环的控制：

```
S = 'abcdefghijk'
for i in range(0, len(S), 2):
    print S[i]
```

在该例子中，我们利用len()函数和range()函数，用i作为S序列的下标来控制循环。在range函数中，分别定义上限，下限和每次循环的步长。这就和C语言中的for循环相类似了。

enumerate()

利用enumerate()函数，可以在每次循环中同时得到下标和元素：

```
S = 'abcdefghijk'
for (index, char) in enumerate(S):
```

```
print index
print char
```

实际上, `enumerate()` 在每次循环中, 返回的是一个包含两个元素的定值表 (tuple), 两个元素分别赋予 `index` 和 `char`

zip()


如果你多个等长的序列, 然后想要每次循环时从各个序列分别取出一个元素, 可以利用 `zip()` 方便地实现:

```
ta = [1,2,3]
tb = [9,8,7]
tc = ['a','b','c']
for (a,b,c) in zip(ta,tb,tc):
    print(a,b,c)
```

每次循环时, 从各个序列分别从左到右取出一个元素, 合并成一个 tuple, 然后 tuple 的元素赋予给 `a,b,c`

`zip()` 函数的功能, 就是从多个列表中, 依次各取出一个元素。每次取出的 (来自不同列表的) 元素合成一个元组, 合并成的元组放入 `zip()` 返回的列表中。`zip()` 函数起到了聚合列表的功能。


我们可以分解聚合后的列表, 如下:



```
ta = [1,2,3]
tb = [9,8,7]

# cluster
zipped = zip(ta,tb)
print(zipped)

# decompose
na, nb = zip(*zipped)
print(na, nb)
```



总结

```
range()

enumerate()

zip()
```

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

80

(请您对文章做出评价)

< 上一篇: Python进阶04 函数的参数对应

> 下一篇: Python进阶06 循环对象

posted @ 2012-07-09 11:52 Vamei 阅读(33438) 评论(25) 编辑 收藏

评论列表

- #1楼 2012-07-09 14:51 Mr h...

简单明了。

支持(0) 反对(0)
- #2楼 2012-07-10 22:30 icuu

呼呼，3 呢

支持(0) 反对(0)
- #3楼[楼主] 2012-07-10 22:57 Vamei

@ culater

3?

支持(0) 反对(0)

#4楼 2012-07-11 16:53 icuu

@ Vamei

引用

@culater

3?

'''

2. 利用enumerate()

4. 利用zip()

'''

缺个3，呵呵

支持(0) 反对(0)

#5楼[楼主] 2012-09-19 13:43 Vamei

@ culater

标错了

支持(0) 反对(0)

#6楼 2013-05-16 01:28 红烧狮子头

很受用希望楼主再接再厉

支持(0) 反对(0)

#7楼 2013-05-21 21:22 kanero

S = 'abcdefghijkl'

--- 这个地方符号错了。应该是半角的

支持(0) 反对(0)

#8楼[楼主] 2013-05-21 23:26 Vamei

@ kanero

改过来了。谢谢你。

支持(0) 反对(0)

#9楼 2013-06-18 16:12 Eve.月

楼主啊...一直觉得python觉少一本好的中文教材，我觉得你的讲述方式很好，家一些

补充和深化，要不你整理成一本书吧...

支持(1) 反对(0)

#10楼 2013-07-11 20:32 杨高明

最后一个程序错了

```
# decompose
```

```
na, nb = zip(*zipped)
```

```
print(na, nb)
```

改成

```
na, nb, nc = zip(*zipped)
```

```
print (na,nb,nc)
```

少一个参数

支持(0) 反对(2)

#11楼[楼主] 2013-07-11 22:01 Vamei

@ 杨高明

是前面的地方错了，不应该有tc。这里应该没错。

支持(1) 反对(0)

#12楼 2013-07-14 09:30 杨高明

@ Vamei

嗯，确实是对的~我自己手动输入的，zipped多了一个tc的值，即

```
zipped = zip(ta,tb,tc)
```

谢谢楼主提醒

支持(0) 反对(0)

#13楼 2013-10-06 16:42 weixiao201

```
>>> for (x,y,z) in zip(a,b,c):
```

```
print [x,y,z]
```

```
[1, 4, 7]
```

```
[2, 5, 8]
```

```
[3, 6, 9]
```

```
>>> for [x,y,z] in zip(a,b,c):  
print (x,y,z)
```

```
(1, 4, 7)
```

```
(2, 5, 8)
```

```
(3, 6, 9)
```

```
>>> for [x,y,z] in zip(a,b,c):  
print {x,y,z}
```

```
set([1, 4, 7])
```

```
set([8, 2, 5])
```

```
set([9, 3, 6])
```

楼主最后一个为什么是这样的结果呢？

支持(0) 反对(0)

#14楼 2013-11-20 11:21 zoo-code

@ weixiao201

那个是定义一个集合 `set` 也可以使用

```
1 | set([1, 2, 3, 1])
```

定义 会返回一个无重复的`set`对象

支持(0) 反对(0)

#15楼 2013-12-28 11:33 itfanr

应该说明一下`range` 函数不取最后一个下标

支持(0) 反对(0)

#16楼 2014-01-08 22:00 凡程子

很清晰，赞一个。

支持(0) 反对(0)

#17楼 2014-12-10 17:00 小静 (Cathy)

zip真没用过，学习了

支持(0) 反对(0)

#18楼 2015-08-05 17:06 小小西红柿

```
1 t1=[1,2,3]
2 t2=['a','b','c']
3 zipped=zip(t1,t2)
4 (b1,b2)=zip(*zipped)
5 print(b1,b2)
6 >>>(1, 2, 3) ('a', 'b', 'c')
```

这段代码到此为止都是正确的，但是如果中间加了一段打印就会报错了：

```
1 t1=[1,2,3]
2 t2=['a','b','c']
3 zipped=zip(t1,t2)
4 print(*zipped)
5 (b1,b2)=zip(*zipped)
6 print(b1,b2)
7 >>>(1, 'a') (2, 'b') (3, 'c')
8 Traceback (most recent call last):
9   File "D:\Python\workspace\150731\src\150731\150731.py", line 39, in <module>
10     (b1,b2)=zip(*zipped)
11 ValueError: need more than 0 values to unpack
```

想请教一下这是为什么呀？只是多了一行打印。。。

支持(0) 反对(0)

#19楼 2015-08-18 17:19 Nefeltari

@小小西红柿

你写错了

```
1 t1 = [1,2,3]
2 t2 = ['a','b','c']
3 zipped = zip(t1,t2)
4 print(zipped) # 这里不能加*
5 (b1,b2)=zip(*zipped)
6 print(b1,b2)
```

在函数调用中使用*list/tuple的方式表示将list/tuple分开，作为位置参数传递给对应函数（前提是对应函数支持不定个数的位置参数）

支持(0) 反对(0)

#20楼 2015-08-25 09:43 靡诗者

正好需要用到zip()这个功能，真是太感谢楼主了！

支持(0) 反对(0)

#21楼 2015-09-21 12:40 yana789

```
1 ta=[1,2,3,3,8]
2 tb=[2,4]
3 tc=[3,3,2,4,0]
4 for (a,b,c) in zip(ta,tb,tc):
5     print(a,b,c)
6 zipped=zip(ta,tb,tc)
7 print(zipped)
8 na,nb,nc=zip(*zipped)
9 print(na,nb,nc)
```

打印出来的结果是：<zip object at 0x030D50A8>

(1, 2) (2, 4) (3, 3)

想问楼主，一个聚合包裹的结果给出的是内存地址吗？

支持(1) 反对(0)

#22楼 2015-11-30 13:41 irunner

```
1 #!/usr/bin/env python
2 #-*- coding:utf-8 -*-
3 #range使用
4
5 s = 'abcdefghjhi'
6 for i in range(0,len(s),2):
7     print s[i]
8 #enumertae使用
9 for (index,char) in enumerate(s):
10     print str(index)+':%s'%char
```



```
11  #zip()使用
12  a1 = [1,2,3]
13  b1 = [4,5,6]
14  c1 = [7,8,9]
15  for (a,b,c) in zip(a1,b1,c1):
16      print (a,b,c)
17
18  ta = [1,2,3]
19  tb = [4,5,6]
20  #cluster
21  zipped = zip(ta,tb)
22  print(zipped)
23
24  #decompose
25  na , nb = zip(*zipped)
26  print (na,nb)
```

支持(0) 反对(0)

#23楼 2016-03-07 18:07 frankie123

@ yana789

引用

同问！

打印出来的结果是：<zip object at 0x030D50A8>

(1, 2) (2, 4) (3, 3)

想问楼主，一个聚合包裹的结果给出的是内存地址吗？

支持(0) 反对(0)

#24楼 2016-03-29 15:25 hitwh_Gypsy

```
[root@MrLiu loop]# cat zipDemo_.py
#!/usr/bin/env python
ta = [1,2,3]
tb = [9,8,7]
print 'ta:',ta
print 'tb:',tb

zipped = zip(ta,tb)
print 'Compose:',(zipped)
na, nb = zip(*zipped)
print 'Decompose:',(na, nb)
print 'Decompose:',na,nb
# the result type is tuple.
[root@MrLiu loop]# py zipDemo_.py
ta: [1, 2, 3]
tb: [9, 8, 7]
Compose: [(1, 9), (2, 8), (3, 7)]
Decompose: ((1, 2, 3), (9, 8, 7))
Decompose: (1, 2, 3) (9, 8, 7)
[root@MrLiu loop]#
```

[支持\(0\)](#) [反对\(0\)](#)

#25楼 2016-04-07 15:05 wqh2016

简单明了，重点突出，楼主啥时候出书 告诉一下啊

[支持\(0\)](#) [反对\(0\)](#)

[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



GrapeCity

ActiveReports

企业级报表服务平台

单独部署、集成应用、报表制作、数据整合
权限管理、移动办公、二次集成开发

[立即了解](#)

JPush 极光推送

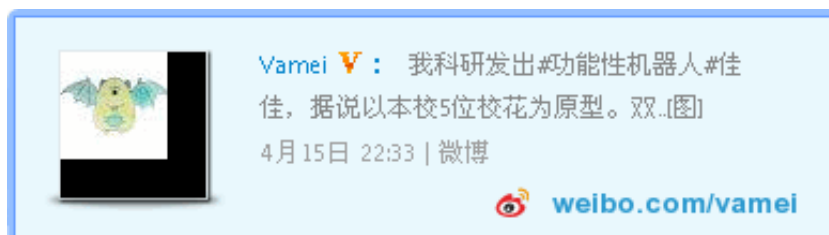
消息推送领导品牌全面升级

JIGUANG 极光

详情点击

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

05370298