

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06 循环对象

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

这一讲的主要目的是为了大家在读Python程序的时候对循环对象有一个基本概念。

循环对象的并不是随着Python的诞生就存在的，但它的发展迅速，特别是Python 3x的时代，循环对象正在成为循环的标准形式。

什么是循环对象

循环对象是这样一个人物，它包含有一个`next()`方法(`__next__()`方法，在python 3x中)，这个方法的目的是进行到下一个结果，而在结束一系列结果之后，举出`StopIteration`错误。

当一个循环结构（比如for）调用循环对象时，它就会每次循环的时候调用`next()`方法，直到`StopIteration`出现，for循环接收到，就知道循环已经结束，停止调用`next()`。

假设我们有一个test.txt的文件：

```
1234
abcd
efg
```

我们运行一下python命令行：

```
>>>f = open('test.txt')

>>>f.next()

>>>f.next()
```

...

不断输入`f.next()`，直到最后出现`StopIteration`

`open()` 返回的实际上是一个循环对象，包含有`next()`方法。而该`next()`方法每次返回的就是新的一行的内容，到达文件结尾时举出`StopIteration`。这样，我们相当于手工进行了循环。

自动进行的话，就是：

```
for line in open('test.txt'):
    print line
```

在这里，`for`结构自动调用`next()`方法，将该方法的返回值赋予给`line`。循环知道出现`StopIteration`的时候结束。

相对于序列，用循环对象的好处在于：不用在循环还没有开始的时候，就生成好要使用的元素。所使用的元素可以在循环过程中逐次生成。这样，节省了空间，提高了效率，编程更灵活。

迭代器

从技术上来说，循环对象和`for`循环调用之间还有一个中间层，就是要将循环对象转换成迭代器(`iterator`)。这一转换是通过使用`iter()`函数实现的。但从逻辑层面上，常常可以忽略这一层，所以循环对象和迭代器常常相互指代对方。

生成器

生成器(`generator`)的主要目的是构成一个用户自定义的循环对象。

生成器的编写方法和函数定义类似，只是在`return`的地方改为`yield`。生成器中可以有多多个`yield`。当生成器遇到一个`yield`时，会暂停运行生成器，返回`yield`后面的值。当再次调用生成器的时候，会从刚才暂停的地方继续运行，直到下一个`yield`。生成器自身又构成一个循环器，每次循环使用一个`yield`返回的值。

下面是一个生成器：

```
def gen():  
    a = 100  
    yield a  
    a = a*8  
    yield a  
    yield 1000
```

该生成器共有三个yield， 如果用作循环器时， 会进行三次循环。

```
for i in gen():  
    print i
```

再考虑如下一个生成器：

```
def gen():  
    for i in range(4):  
        yield i
```

它又可以写成生成器表达式 (Generator Expression)：

```
G = (x for x in range(4))
```

生成器表达式是生成器的一种简便的编写方式。读者可进一步查阅。

表推导

表推导 (list comprehension) 是快速生成表的方法。它的语法简单，很有实用价值。

假设我们生成表L：

```
L = []  
for x in range(10):  
    L.append(x**2)
```

以上产生了表L，但实际上有快捷的写法，也就是**表推导**的方式：

```
L = [x**2 for x in range(10)]
```

这与生成器表达式类似，只不过用的是**中括号**。

（表推导的机制实际上是利用循环对象，有兴趣可以查阅。）

练习 下面的表推导会生成什么？

```
x1 = [1,3,5]
y1 = [9,12,13]
L = [ x**2 for (x,y) in zip(x1,y1) if y > 10]
```

总结

循环对象

生成器

表推导

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](#)

16

0

(请您对文章做出评价)

« 上一篇: [Python进阶05 循环设计](#)

» 下一篇: [Python进阶07 函数对象](#)

posted @ 2012-07-09 15:05 Vamei 阅读(32678) 评论(43) 编辑 收藏

评论列表

#1楼 2012-12-16 17:38 leopardsaga

循环对象 与 循环器，
可以再论述下不

支持(0) 反对(0)

#2楼 2012-12-28 22:03 next163

我一直管这货叫迭代器。

支持(1) 反对(0)

#3楼 2013-03-04 22:27 pang18a

谁起的这名啊 表理解 真别扭。。。

支持(0) 反对(0)

#4楼[楼主] 2013-03-04 22:37 Vamei

@ pang18a

我根据英文翻译的，
你给个建议呗。

支持(0) 反对(0)

#5楼 2013-03-22 20:40 名而已

为什么我在F:\python\sorce路径下写入text.txt
然后f=open('F:\python\sorce\text.txt')

会提示错误：

Traceback (most recent call last):

File "<pyshell#1>", line 1, in <module>

f = open('F:\python\sorce\test.txt','r')

IOError: [Errno 22] Invalid argument: 'F:\\python\\sorce\test.txt'

这是怎么回事啊

支持(0) 反对(0)

#6楼[楼主] 2013-03-22 23:19 Vamei

@ 名而已

为什么变成双\\呢，这个很奇怪，没见过。
你把文件放到工作目录，看看有问题没有。

支持(0) 反对(0)

#7楼 2013-03-26 21:30 名而已

@ Vamei

我把文件放在了工作目录，结果提示

```
>>> f = open('test.txt')
>>> f.next()
Traceback (most recent call last):
File "<pyshell#1>", line 1, in <module>
f.next()
AttributeError: '_io.TextIOWrapper' object has no attribute 'next'
```

怎么回事呀

支持(0) 反对(0)

#8楼[楼主] 2013-03-26 22:19 Vamei

@ 名而已

你的test.txt里内容是什么？我尝试了一下，用下面的内容

a
b
c

是可以的。

支持(0) 反对(0)

#9楼 2013-03-26 22:34 名而已

@ Vamei

之前的内容是教程里的，后面改成a b c也是这样的提示哦

会不会是python3有什么变化吗？

我是在win7下面运行的

支持(0) 反对(0)

#10楼[楼主] 2013-03-26 22:50 Vamei

@ 名而已

你尝试一下，用

```
f = open("test.txt", "r")
```

```
f.read()
```

看是否可以。

支持(0) 反对(0)

#11楼[楼主] 2013-03-26 22:56 Vamei

@ 名而已

哦，还有一个，我发现我已经注明了，用__next__()方法，而不是next()方法

支持(0) 反对(0)

#12楼 2013-03-27 00:08 名而已

@ Vamei

嗯嗯，试过后可了啦

而且我发现文件放在别的目录下，文件名里好像不能有test这几个字呐，我改成其它名字就是可以open的，好奇怪哦

谢谢啦

支持(0) 反对(0)

#13楼 2013-05-17 00:05 红烧狮子头

练习是LIST[9,25]吧

支持(0) 反对(0)

#14楼[楼主] 2013-05-17 22:29 Vamei

@ 红烧狮子头

O(n_n)O~，运行一下就直到了。

支持(0) 反对(0)

#15楼 2013-05-17 22:53 红烧狮子头

昨天看了下list comprehesion有的地方翻译成列表推导式~博主可以看看这种可行否哈哈~

支持(0) 反对(0)

#16楼[楼主] 2013-05-18 07:29 Vamei

@ 红烧狮子头

有一点奇怪。我再找找看。

支持(0) 反对(0)

#17楼[楼主] 2013-05-18 07:29 Vamei

@ 红烧狮子头

谢谢你了。

支持(0) 反对(0)

#18楼 2013-10-06 17:24 weixiao201

标记下

支持(0) 反对(0)

#19楼 2013-10-15 17:16 混沌奇迹

@ 名而已

这个是因为路径名出现了\t，这个转义字符。

你改成f = open(r'F:\python\sorce\test.txt')

支持(0) 反对(0)

#20楼 2013-10-15 17:23 混沌奇迹

@ 名而已

可以参考

<http://www.douban.com/note/246259307/#>

支持(1) 反对(0)

#21楼 2013-11-20 19:24 sunrong

```
1 1234
2
3 abcd
4
5 efg
6
7 100
8 800
9 1000
10 [9, 25]
```

支持(0) 反对(0)

#22楼 2013-12-29 14:31 _DN

换种叫法吧——迭代器。不管是C++ STL 中，还是设计模式（Design Pattern）中，都直接叫做迭代器。当然，这样叫是为了遵循大众（Population）的约定。

总之，楼主，我很敬佩你，我要好好学学你写文章的风格，这是某一类人的福音。。。

支持(1) 反对(0)

#23楼[楼主] 2013-12-29 22:24 Vamei

@ _DN

嗯。迭代器确实更通用。但这篇文章里有两个东西：

iterable object和iterator。哪个是你说的迭代器呢？

支持(0) 反对(0)

#24楼 2014-02-18 13:52 只是_曾经

```
xl = [1,3,5]
```

```
yl = [9,12,13]
```

```
L = [ x**2 for (x,y) in zip(xl,yl) if y > 10]
```

我把练习理解错了。。

我以为是用表推导式表示xl,yl，然后写出L的表元素。。。

看了半天没想通yl怎么弄

支持(0) 反对(0)

#25楼 2014-04-22 22:17 Seandor

那东西原来叫生成器表达式啊，一直在用，却不知道是什么。学到了，感谢楼主！

支持(0) 反对(0)

#26楼 2014-06-11 04:26 nietzschetmh

@ 混沌奇迹

此楼是正解。

支持(0) 反对(0)

#27楼 2014-06-15 15:15 呼噜鱼90

Win7下用Python2.7.3版本运行

```
1 | G = (x for x in range(4))
2 | print G
```

会报

```
1 | <generator object <genexpr> at 0x000000002BBF510>
```

这种错误。但是

```
1 | G = [x for x in range(4)]
2 | print G
```

就能正常显示

```
1 | >>> ===== RESTART =====
2 | >>>
3 | [0, 1, 2, 3]
```

请问这是什么原因？

支持(0) 反对(0)

#28楼 2014-06-16 19:13 darkread

list用[]而不是()

支持(0) 反对(0)

#29楼[楼主] 2014-06-16 19:49 Vamei

@ 呼噜鱼90

见楼上

支持(0) 反对(0)

#30楼 2014-09-03 20:35 水白酱

@ Vamei

但是这样定义的迭代器应该怎么用呢？

```
1 | G = (x for x in range(4))
2 | print G
```

支持(0) 反对(0)

#31楼 2014-09-12 02:40 热心网友韭菜盒子

讲的太明白了，生成器那里

支持(0) 反对(0)

#32楼 2014-10-10 11:18 NUST小文

G = (x for x in range(4))应该写成

G = [x for x in range(4)]

这样才对的把

支持(0) 反对(0)

#33楼 2014-10-16 10:46 winstic

@ 呼噜鱼90:

G = (x for x in range(4))返回的是迭代器(generator)对象，再遍历一次即可

```
1  >>> G = (x for x in range(4))
2  >>> type(G)
3  <type 'generator'>
4  >>> for i in G:
5      print i
6
7
8  0
9  1
10 2
11 3
```

支持(1) 反对(0)

#34楼 2015-03-26 16:10 SLOW_LEARNER

建议标题直接用标准的英文:

比如生成器标题直接写 Generator

表推导标题直接写 List Comprehension

因为有些其他的教程List Comprehension又叫生成式，列表生成式之类的。

Generator 和 List Comprehension 又较为相似，狐括号和方括号的区别。

所以各种教程的翻译不同，而不少初学者又没有看英文的习惯，可能一些术语的英文就直接忽略了，这样就很容易产生混淆，而自己一直都不知道。

支持(0) 反对(0)

#35楼 2015-05-30 22:10 迷途ing

```
1  x1 = [1,3,5]
2  y1 = [9,12,13]
```

```
3 | Ly=[x**2 for(x,y)in zip(x1,y1) if y>10]
```

分解：首先zip(x1,y) 生成[(1,9),(3,12),(5,15)] 然后对应该(x,y)

x=1,y=9 第一组

x=3,y=12第二组

x=5,y=15第三组

if y>10 适合的只有第二 第三组，在然后在求x**2

最终结果[9 , 2 5]

支持(0) 反对(0)

#36楼 2015-07-01 17:32 少南，无恒

貌似generator expression只能迭代一次，同iterator和generator不同

```
1 | ##1. implement itetator
2 | class TrArray:
3 |
4 |     def __init__(self,n):
5 |         self._currentIndex = 0
6 |         self._length = n
7 |
8 |     def __iter__(self):
9 |         return self
10 |
11 |     def next(self): #python 3: __next__
12 |         if self._currentIndex < self._length :
13 |             index = self._currentIndex
14 |             self._currentIndex += 1
15 |             return index
16 |         else:
17 |             raise StopIteration()
18 |
19 | for i in TrArray(10):
20 |     print i,
21 |
22 | print ""
23 |
24 | for i in TrArray(10):
25 |     print i,
26 |
27 | print ""
28 | ##2. generator
29 | def Gen(n):
30 |     i = 0
```

```
31     while i < n:
32         yield i
33         i += 1
34
35 for i in Gen(11):
36     print i,
37
38
39 ##3. generator expression
40 G = (i*2 for i in range(10))
41 print "\r\nG:",G
42 print "\r\n", sum(G), "List:"
43
44 ##only use once
45 for i in G:
46     print i,
47
48
49 G1 = [i*2 for i in range(10)]
50 print "\r\nG1:",G1
51 print "\r\n", sum(G1), "List:"
52 ##use iterator object of list
53 for i in G1:
54     print i,
55
56 ##just test multiple yield
57 print "\r\n"
```

支持(1) 反对(0)

#37楼 2015-09-07 11:47 yana789

所使用的元素在循环过程中逐次生成是什么意思啊？谢谢解答

支持(0) 反对(0)

#38楼 2015-10-05 15:02 苏生不惑

@ 水白酱

list(x for x in range(4))

支持(0) 反对(0)

#39楼 2015-11-10 09:03 nevermoreee

为什么[print(line.strip()) for line in open('test.txt')]输出的结果是

X

X,X

X,X,X

[None,None,None]

而L=[print(line.strip()) for line in open('test.txt')]输出的结果则不包含
[None,None,None]，再输入L输出只有[None,None,None]?

支持(0) 反对(0)

#40楼 2016-01-27 21:55 loogn

表推导 好像和 lisp里有点像，但是忘记那里叫什么了

支持(0) 反对(0)

#41楼 2016-04-07 15:59 wqh2016

```
1 >>> x1=[1,3,5]
2 >>> y1=[9,12,13]
3 >>> L = [x**2 for (x,y) in zip(x1,y1) if y >10]
4 >>> print L
5 [9, 25]
6 >>>
```

支持(0) 反对(0)

#42楼 2016-05-16 19:35 ilpay99

@ Vamei

表推导和列表解析有什么不同么 - -

支持(0) 反对(0)

#43楼 2016-06-29 10:10 ashic

好好地列表解析变成了表推导

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。

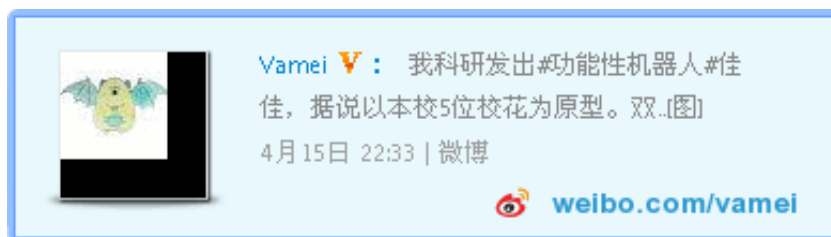
【推荐】50万行VC++源码：大型组态工控、电力仿真CAD与GIS源码库

【推荐】融云即时通讯云—豆果美食、Faceu等亿级APP都在用



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/assets/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

05370299