代码这样写更优雅 (Python 版)

刘志军
公众号：Python之禅

关注他

487 人赞了该文章

Python 这门语言最大的优点之一就是语法简洁，好的代码就像伪代码一样，干净、整洁、一目了然。但有时候我们写代码，特别是 Python 初学者，往往还是按照其它语言的思维习惯来写，那样的写法不仅运行速度慢，代码读起来也费尽，给人一种拖泥带水的感觉，过段时间连自己也读不懂。

《计算机程序的构造和解释》的作者哈尔·阿伯尔森曾这样说： “Programs must be written for people to read, and only incidentally for machines to execute.”

要写出 Pythonic（优雅的、地道的、整洁的）代码，还要平时多观察那些大牛代码，Github 上有很多非常优秀的源代码值得阅读，比如：requests、flask、tornado，笔者列举一些常见的 Pythonic 写法，希望能给你带来一点启迪。

1、变量交换

大部分编程语言中交换两个变量的值时，不得不引入一个临时变量：

```
>>> a = 1
>>> b = 2
>>> tmp = a
>>> a = b
>>> b = tmp
```

pythonic

```
>>> a, b = b, a
```

2、循环遍历区间元素

```
for i in [0, 1, 2, 3, 4, 5]:
    print i
# 或者
```

pythonic

```
for i in xrange(6):
    print (i)
```

xrange 返回的是生成器对象，生成器比列表更加节省内存，不过需要注意的是 xrange 是 python2 中的写法，python3 只有 range 方法，特点和 xrange 是一样的。

3、带有索引位置的集合遍历

遍历集合时如果需要使用到集合的索引位置时，直接对集合迭代是没有索引信息的，普通的方式使用：

```
colors = ['red', 'green', 'blue', 'yellow']

for i in range(len(colors)):
    print (i, '--->', colors[i])
```

pythonic

```
for i, color in enumerate(colors):
    print (i, '--->', color)
```

4、字符串连接

字符串连接时，普通的方式可以用 + 操作

```
names = ['raymond', 'rachel', 'matthew', 'roger',
         'betty', 'melissa', 'judith', 'charlie']

s = names[0]
for name in names[1:]:
    s += ', ' + name
print (s)
```

pythonic

```
print (', '.join(names))
```

join 是一种更加高效的字符串连接方式，使用 + 操作时，每执行一次+操作就会导致在内存中生成一个新的字符串对象，遍历8次有8个字符串生成，造成无谓的内存浪费。而用 join 方法整个过程只会产生一个字符串对象。

5、打开/关闭文件

执行文件操作时，最后一定不能忘记的操作是关闭文件，即使报错了也要 close。普通的方式是在 finnally 块中显示的调用 close 方法。

```
f = open('data.txt')
try:
    data = f.read()
```

pythonic

```
with open('data.txt') as f:
    data = f.read()
```

使用 with 语句，系统会在执行完文件操作后自动关闭文件对象。

6、列表推导式

能够用一行代码简明扼要地解决问题时，绝不要用两行，比如

```
result = []
for i in range(10):
    s = i*2
    result.append(s)
```

pythonic

```
[i*2 for i in xrange(10)]
```

与之类似的还有生成器表达式、字典推导式，都是很 pythonic 的写法。

7、善用装饰器

装饰器可以把与业务逻辑无关的代码抽离出来，让代码保持干净清爽，而且装饰器还能被多个地方重复利用。比如一个爬虫网页的函数，如果该 URL 曾经被爬过就直接从缓存中获取，否则爬下来之后加入到缓存，防止后续重复爬取。


```
def web_lookup(url, saved={}):
    if url in saved:
        return saved[url]
    page = urllib.urlopen(url).read()
    saved[url] = page
    return page
```

pythonic

```
import urllib #py2
#import urllib.request as urllib # py3

def cache(func):
    saved = {}

    def wrapper(url):
        if url in saved:
            return saved[url]
        else:
            page = func(url)
            saved[url] = page
            return page

    return wrapper

@cache
```

用装饰器写代码表面上感觉代码量更多，但是它把缓存相关的逻辑抽离出来了，可以给更多的函数调用，这样总的代码量就会少很多，而且业务方法看起来简洁了。

8、合理使用列表

列表对象（list）是一个查询效率高于更新操作的数据结构，比如删除一个元素和插入一个元素时执行效率就非常低，因为还要对剩下的元素进行移动

```
names = ['raymond', 'rachel', 'matthew', 'roger',
         'betty', 'melissa', 'judith', 'charlie']
names.pop(0)
names.insert(0, 'mark')
```

pythonic

```
from collections import deque
names = deque(['raymond', 'rachel', 'matthew', 'roger',
               'betty', 'melissa', 'judith', 'charlie'])
names.popleft()
names.appendleft('mark')
```

deque 是一个双向队列的数据结构，删除元素和插入元素会很快

9、序列解包


```
p = 'vttalk', 'female', 30, 'python@qq.com'

name = p[0]
gender = p[1]
age = p[2]
email = p[3]
```

pythonic

```
name, gender, age, email = p
```

10、遍历字典的 key 和 value

方法一速度没那么快，因为每次迭代的时候还要重新进行hash查找 key 对应的 value。

方法二遇到字典非常大的时候，会导致内存的消耗增加一倍以上

```
# 方法一
for k in d:
    print k, '--->', d[k]

# 方法二
for k, v in d.items():
    print (k, '--->', v)
```

pythonic

iteritems 返回迭代器对象，可节省更多的内存，不过在 python3 中没有该方法了，只有 items 方法，等值于 iteritems。

当然还有很多 pythonic 写法，在此不再一一列举，说不定有第二期，欢迎留言。觉得不错就赞一个吧 (^o^)/

> 公众号『Python之禅』（id:VTtalk），分享 Python 等技术干货和有温度的内容，欢迎关注
> 博客地址：代码这样写更优雅(Python版) - FooFish

编辑于 2017-03-10

Python

知乎 首发于 Python之禅 关注

38 条评论 切换为时间排序

写下你的评论...

吃面崩掉牙 1 年前
希望能介绍深入介绍一下装饰器。每次看的时候都觉得懂了，但是写起来脑子里就是一团浆糊
2

刘志军 (作者) 回复 吃面崩掉牙 1 年前
看这篇，zhuanlan.zhihu.com/p/24...， 就不信你还搞不懂
赞

马龙 回复 吃面崩掉牙 1 年前
同感.....
赞

展开其他 2 条回复

陈二白 1 年前
返回一个列表的时候，使用yield将函数变成一个生成函数，不知道这样说能不能理解。
2

程序猿 1 年前
python 只不过是后台帮你做了哪些工！
1

只增笑耳Jason 1 年前
那个 i2 是什么意思?
1

Karen L 回复 只增笑耳Jason 1 年前
同问
赞

刘志军 (作者) 回复 只增笑耳Jason 1 年前
多谢指正，看得非常仔细
赞

郭俊卿 1 年前
妈呀，最近正在刷leetcode。有了这些有技巧，我的强迫症又要发作了！
赞

Rancho 1 年前
大部分还可以，就是列表的格式不好，命名也不太好。
赞

刘志军 (作者) 回复 Rancho 1 年前
是的，没有想到太真实的场景，就用了很随意的名字。这很不优雅。啊哈哈
1

GregoryGT 1 年前
用3的OCD表示看到print
赞

哈哈哈，我加上

1

卖鱼Sri 1 年前

答主好人，好人一生平安

赞

芝麻 1 年前

简书上有篇文章一模一样

赞

Harp Liu 1 年前

我之前一直写的都是for i in range(len()): 😂

赞

zltrajectory 1 年前

正在考虑学phyon，想问一个问题，这种需要是否能用于硬件？控制器。谢谢！

赞

orangleliu 回复 zltrajectory 1 年前

可以啊 树莓派呀。py的硬件驱动还是不错的，有专门的嵌入式版本了。

1

weiyinfu 1 年前

为啥不用Python3？写教程最好用Python3

赞

蜗牛在唱歌 1 年前

学习了。

赞

银色闪光 1 年前

所以2和10就是说——P3大法好

赞

刘志军 (作者) 回复 银色闪光 1 年前

是的，就python2中字符串能把人搞崩溃，改天写写

赞

王志辰 1 年前

攒+收藏

赞

混沌的鳄鱼 1 年前

4、字符串连接 这个说法不准确， 实际上在连接少量字符串时，比如2到3个，直接用+号要比join效率高。

赞

刘志军 (作者) 回复 混沌的鳄鱼 1 年前

为什么?

赞

混沌的鳄鱼 回复 刘志军 (作者) 1 年前

不为什么，用timeit做过100万次的对比测试的结果。

1

天元 1 年前

mark

赞

天日赵 1 年前

python版？之后还要出其它语言版本吗

赞

艾客 1 年前

写的很好，期待第二期！装饰器确实比较难(捂脸)

赞

AkagiInc 1 年前

這個。兩個專欄裡面看到同一個文章。什麼情況

赞

刘志军 (作者) 回复 AkagiInc 1 年前

另外一个专栏转载了

赞

1 2 下一页