

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09 动态类型

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

谢谢TeaEra，猫咪cat

动态类型(dynamic typing)是Python另一个重要的核心概念。我们之前说过，Python的变量(variable)不需要声明，而在赋值时，变量可以重新赋值为任意值。这些都与动态类型的概念相关。

动态类型

在我们接触的对象中，有一类特殊的对象，是用于存储数据的。常见的该类对象包括各种数字，字符串，表，词典。在C语言中，我们称这样一些数据结构为变量。而在Python中，这些是对象。

对象是储存在内存中的实体。但我们并不能直接接触到该对象。我们在程序中写的对象名，**只是指向这一对象的引用(reference)**。

引用和对象分离，是动态类型的核心。引用可以随时指向一个新的对象：

```
a = 3
a = 'at'
```

第一个语句中，3是储存在内存中的一个整数对象。**通过赋值，引用a指向对象3**。

第二个语句中，内存中建立对象'at'，是一个字符串(string)。引用a指向了'at'。此时，对象3不再有引用指向它。Python会自动将没有引用指向的对象销毁(destroy)，释放相应内存。

(对于小的整数和短字符串，Python会缓存这些对象，而不是频繁的建立和销毁。)

```
a = 5
b = a
a = a + 2
```

再看这个例子。通过前两个句子，我们让a,b指向同一个整数对象5 (b = a的含义是让引用b指向引用a所指的**那一个**对象)。但第三个句子实际上对引用a重新赋值，让a指向一个新的对象7。此时a,b分别指向不同的对象。我们看到，**即使是多个引用指向同一个对象，如果一个引用值发生变化，那么实际上是让这个引用指向一个新的引用，并不影响其他的引用的指向**。从效果上看，就是各个引用各自独立，互不影响。

其它数据对象也是如此：

```
L1 = [1, 2, 3]
L2 = L1
L1 = 1
```

但注意以下情况

```
L1 = [1, 2, 3]
L2 = L1
L1[0] = 10
print L2
```

在该情况下，我们不再对L1这一引用赋值，而是对L1所指向的表的元素赋值。结果是，L2也同时发生变化。

原因何在呢？因为L1, L2的指向没有发生变化，依然指向那个表。表实际上是包含了多个引用的对象（每个引用是一个元素，比如L1[0], L1[1]..., 每个引用指向一个对象，比如1, 2, 3), 。而L1[0] = 10这一赋值操作，并不是改变L1的指向，而是对L1[0], 也就是表对象的一部份(一个元素)，进行操作，所以所有指向该对象的引用都受到影响。

(与之形成对比的是，我们之前的赋值操作都没有对对象自身发生作用，只是改变引用指向。)

列表可以通过引用其元素，**改变对象自身 (in-place change)**。这种对象类型，称为**可变数据对象 (mutable object)**，词典也是这样的数据类型。

而像之前的数字和字符串，**不能改变对象本身，只能改变引用的指向**，称为**不可变数据对象 (immutable object)**。

我们之前学的元组 (tuple)，尽管可以调用引用元素，但不可以赋值，因此不能改变对象自身，所以也算是 immutable object。

从动态类型看函数的参数传递

函数的参数传递，本质上传递的是引用。比如说：



```
def f(x):  
    x = 100  
    print x
```

```
a = 1  
f(a)  
print a
```



参数x是一个新的引用，指向a所指的**对象**。如果参数是不可变 (immutable) 的对象，a和x引用之间相互独立。对参数x的操作不会影响引用a。这样的传递类似于C语言中的值传递。

如果传递的是可变 (mutable) 的对象，那么改变函数参数，有可能改变原对象。所有指向原对象的引用都会受影响，编程的时候要对此问题留心。比如说：



```
def f(x):  
    x[0] = 100  
    print x
```

```
a = [1, 2, 3]
f(a)
print a
```



动态类型是Python的核心机制之一。可以在应用中慢慢熟悉。

总结

引用和对象的分离，对象是内存中储存数据的实体，引用指向对象。

可变对象，不可变对象

函数值传递

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](#)

15

0

(请您对文章做出评价)

« 上一篇: [Python进阶08 异常处理](#)

» 下一篇: [Python快速教程总结](#)

posted @ 2012-07-10 19:43 Vamei 阅读(29471) 评论(22) 编辑 收藏

评论列表

#1楼 2012-09-18 16:34 TeaEra

楼主的教程好棒!!! 我最近一直在看

不过第二段“2. 从动态类型看函数的参数传递”中, 关于参数是mutable还是immutable会发生值传递还是地址传递是不是写反了呢? 希望探讨一下

支持(0) 反对(0)

#2楼[楼主] 2012-09-18 16:57 Vamei

@ TeaEra

你说的对，是我写错了。谢谢你帮我纠正！

支持(0) 反对(0)

#3楼 2012-09-18 17:00 TeaEra

@ Vamei

我该谢谢你才对，你这个教程让我对python有了更深的认识~希望你能写更多的关于python的文章，我其实对python web方面的也很有趣~

支持(0) 反对(0)

#4楼[楼主] 2012-09-18 17:08 Vamei

@ TeaEra

我对template研究不深入，只是试用过google app engine。最近在写python标准库和python绘图方面的文章。等有时间了研究下template再说python web方面的吧。

支持(0) 反对(0)

#5楼 2012-11-01 13:37 猫咪cat

```
def f(x):
```

```
    x = 100
```

```
    print x
```

```
a = 1
```

```
f(x) #是指f(a)吧？
```

```
print a
```

支持(0) 反对(0)

#6楼[楼主] 2012-11-01 14:53 Vamei

@ 猫咪cat

yes, I just corrected it.

支持(0) 反对(0)

#7楼 2013-11-20 12:51 zoo-code

可以使用id函数判断一下 是不是同一个对象

支持(0) 反对(0)

#8楼[楼主] 2013-11-20 13:24 Vamei

@ zoo-code

是的。这个我在Python深入06里面提了。

支持(0) 反对(0)

#9楼 2013-11-21 10:11 sunrong

"即使是多个变量指向同一个对象，如果一个变量值发生变化，那么实际上是让这个变量指向一个新的变量，并不影响其他的变量的指向。"

是不是要改为".....让这个变量指向一个新的对象....."不知道我这样理解对吗？

支持(0) 反对(0)

#10楼 2013-11-23 21:04 庞昊

@ sunrong

应该是指向一个新的对象。

另外："我们的变量，实际上只是指向这一对象的参考(reference)"，这里reference应该翻译成这一对象的"引用"，就像C++的引用一样。

支持(0) 反对(0)

#11楼 2014-02-04 22:02 kingboo501

想请问下，比如列表L1的对象是否 是L1[0]？因为L1[0] 是指向元素1的引用，引用可变，即列表的对象可变，所以列表就是可变数据类型？？

支持(0) 反对(0)

#12楼[楼主] 2014-02-05 12:09 Vamei

@ kingboo501

列表的对象就是指整个列表，包含了指向各个元素的引用。

一个元素，即一个引用，只是列表的组成部分。

由于组成部分可变，所以列表包含的数据发生了变化，但列表的id不变。所以说，列表是可变数据类型。

支持(0) 反对(0)

#13楼 2014-04-23 22:21 Seandor

python的对象就像一个tag。

支持(0) 反对(0)

#14楼 2014-07-01 12:23 wzhscrip

感觉这个地方表述有点绕，不容易看明白：

原文的表述“即使是多个引用指向同一个对象，如果一个引用值发生变化，那么实际上是让这个引用指向一个新的引用，并不影响其他的引用的指向”，

建议的表述“即使是多个引用指向同一个对象**a**，如果一个引用值发生变化，那么实际上是让这个引用指向一个新的对象**b**，并不影响其他的引用的值以及对象**a**”

初学者，有什么理解不对的地方，希望楼主指点:D

支持(0) 反对(0)

#15楼 2014-09-29 18:15 逆火xhz

哥，这个模块加上对象内存分配状况图就更好了

支持(0) 反对(0)

#16楼 2015-03-18 17:44 liuwaner118

这里的可变对象和不可变对象参照与java中的String 来理解比较好

支持(0) 反对(0)

#17楼 2015-04-13 20:26 笨鸟凌峰

'即使是多个引用指向同一个对象，如果一个引用值发生变化，那么实际上是让这个引用指向一个新的引用，并不影响其他的引用的指向'

这里应该是'让这个引用指向一个新的对象'吧？

支持(0) 反对(0)

#18楼 2015-04-26 00:11 卡夫卡不卡

@ 笨鸟凌峰

一个意思吧。

支持(0) 反对(0)

#19楼 2015-08-01 22:24 woodyle

个人认为把list和dict理解成C里面的指针类型就好多了

支持(1) 反对(0)

#20楼 2016-02-21 20:55 Iamlunatic

对于零基础的，真的是很好理解

支持(0) 反对(0)

#21楼 2016-04-04 16:57 hitwh_Gypsy

多谢，学习ing

[支持\(0\)](#) [反对\(0\)](#)

#22楼 2016-04-07 19:09 wqh2016

学习完毕 今天就学到这啦

[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

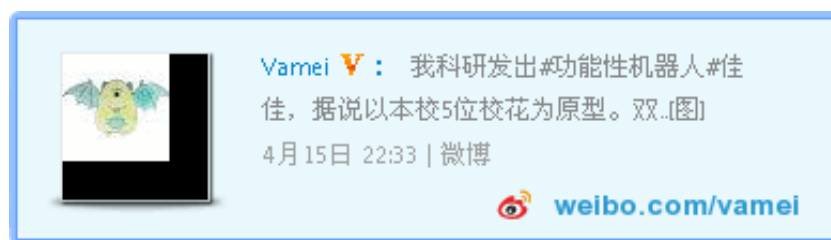



JPUSH 极光推送 消息推送领导品牌全面升级 JIGUANG 极光
详情点击

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



Vamei ：我科研发出#功能性机器人#佳佳，据说以本校5位校花为原型。双...[图]
4月15日 22:33 | 微博
weibo.com/vamei

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

05370304