

『大魔王』

博客园 首页 新闻 问答 职位 招聘 管理

python的可变与不可变数据类型

<python的可变与不可变数据类型>

python与C/C++不一样，它的变量使用有自己的特点，当初学python的时候，一定要记住“一切皆为对象，一切皆为对象的引用”这句话，其实这个特点类似于JAVA，所以在python里面大家也不用担心类似于C/C++中的指针的复杂问题，在python中数据分为**可变数据类型**，**不可变数据类型**。

- 可变数据类型：列表list和字典dict。
 - 不可变数据类型：整型int、浮点型float、字符串型string和元组tuple。
- 以int和list为例，分析“可变数据类型”和“不可变数据类型”之间的区别。

➤ 不可变数据类型分析

```
>>> x = 1
>>> id(x)
31106520
>>> y = 1
>>> id(y)
31106520
>>> x = 2
>>> id(x)
31106508
>>> y = 2
>>> id(y)
31106508
>>> z = y
>>> id(z)
31106508

>>> x += 2
>>> id(x)
31106484
```

上面这段程序都是对不可变数据类型中的int类型的操作，**id()**查看的是**当前变量的地址值**。我们先来看x = 1和y = 1两个操作的结果，从上面的输出可以看到**x和y在此时的地址值是一样的，也就是说x和y其实是引用了同一个对象，即1，也就是说内存中对于1只占用了一个地址，而不管有多少个引用指向了它，都只有一个地址值，只是有一个引用计数会记录指向这个地址的引用到底有几个而已**。当我们进行x = 2赋值时，发现x的地址值变了，虽然还是x这个引用，但是其地址值却变化了，后面的y = 2以及z = y，使得x、y和z都引用了同一个对象，即2，所以地址值都是一样的。当x和y都被赋值2后，1这个对象已经没有引用指向它了，所以1这个对象占用的内存，即31106520地址要被“垃圾回收”，即1

公告

昵称：「大魔王」
园龄：5个月
粉丝：0
关注：0
[+加关注](#)

< 2018年3月				
日	一	二	三	四
25	26	27	28	1
4	5	6	7	8
11	12	13	14	15
18	19	20	21	22
25	26	27	28	29
1	2	3	4	5

搜索

随笔分类

- Linux 设备驱动(79)
- linux 系统编程(5)
- ONVIF(6)
- Python(10)
- 音视频编解码(1)

随笔档案

这个对象在内存中已经不存在了。最后，x进行了加2的操作，所以创建了新的对象4，x引用了这个新的对象，而不再引用2这个对象。

之所以称为不可变数据类型，这里的不可变可以理解为x引用的地址处的值是不能被改变的，也就是31106520地址处的值在没被垃圾回收之前一直都是1，不能改变，如果要把x赋值为2，那么只能将x引用的地址从31106520变为31106508，相当于x = 2这个赋值又创建了一个对象，即2这个对象，然后x、y、z都引用了这个对象，所以int这个数据类型是不可变的，如果想对int类型的变量再次赋值，在内存中相当于又创建了一个新的对象，而不再是之前的对象。

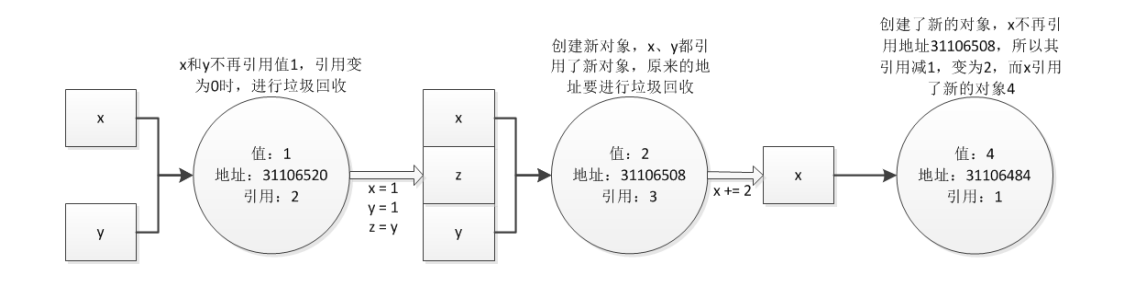


图1 python不可变数据类型分析

从上面的过程可以看出，不可变数据类型的优点就是内存中不管有多少个引用，相同的对象只占用了一块内存，但是它的缺点就是当需要对变量进行运算从而改变变量引用的对象的值时，由于是不可变的数据类型，所以必须创建新的对象，这样就会使得一次次的改变创建了一个个新的对象，不过不再使用的内存会被垃圾回收器回收。

➤ 可变量数据类型分析

```
>>> a = [1, 2, 3]
>>> id(a)
41568816
>>> a = [1, 2, 3]
>>> id(a)
41575088
>>> a.append(4)
>>> id(a)
41575088
>>> a += [2]
>>> id(a)
41575088
>>> a
[1, 2, 3, 4, 2]
```

从上面的程序中可以看出，进行两次a = [1, 2, 3]操作，两次a引用的地址值是不同的，也就是说其实创建了两个不同的对象，这一点明显不同于不可变数据类型，所以对于可变量数据类型来说，具有同样值的对象是不同的对象，即在内存中保存了多个同样值的对象，地址值不同。接着来看后面的操作，我们对列表进行添加操作，分别a.append(4)和a += [2]，发现这两个操作使得a引用的对象值变成了上面的最终结果，但是a引用的地址依旧是41575088，也就是说对a进行的操作不会改变a引用的地址值，只是在地址后面又扩充了新的地址，改变了地址里面存放的值，所以可变量数据类型的意义就是说对一个变量进行操作时，其值是可变的，值的变化并不会引起新建对象，即地址是不会变的，只是地址中的内容变化了或者地址得到了扩充。

2018年3月 (81)

2018年2月 (1)

2017年10月 (20)

阅读排行榜

- 1. 基于ONVIF协议的摄像头; 0)
- 2. 交叉编译OpenSSL(457)
- 3. Python 面向对象编程-- 象(426)
- 4. Linux内核镜像格式(365)
- 5. 使用gSOAP工具生成onvi 88)

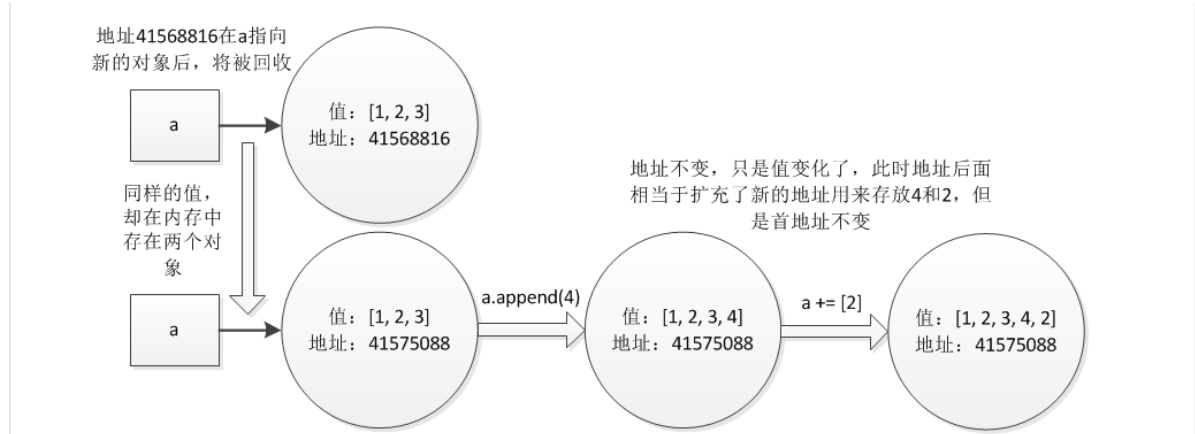


图2 python可变数据类型分析

从上述过程可以看到，可变数据类型是允许同一对象的内容，即值可以变化，但是地址是不会变化的。但是需要注意一点，对可变数据类型的操作不能是直接进行新的赋值操作，比如说a = [1, 2, 3, 4, 5, 6, 7]，这样的操作就不是改变值了，而是新建了一个新的对象，这里的可变只是对于类似于append、+=等这种操作。

> 总结

用一句话来概括上述过程就是：“python中的**不可变数据类型**，不允许变量的值发生变化，如果改变了变量的值，相当于新建了一个对象，而对于相同的值的对象，在内存中则只有一个对象，内部会有一个引用计数来记录有多少个变量引用这个对象；**可变数据类型**，允许变量的值发生变化，即如果对变量进行append、+=等这种操作后，只是改变了变量的值，而不会新建一个对象，变量引用的对象的地址也不会变化，不过对于相同的值的不同对象，在内存中则会存在不同的对象，即每个对象都有自己的地址，相当于内存中对于同值的对象保存了多份，这里不存在引用计数，是实实在在的对象。”

<wiz_tmp_tag id="wiz-table-range-border" contenteditable="false" style="display: none;">

来自为知笔记(Wiz)

分类： Python

好文要顶

关注我

收藏该文

「大魔王」

关注 - 0

粉丝 - 0

+加关注

0

0

« 上一篇: Python 2.7.x 和 3.x 版本的语法区别

» 下一篇: Python 面向对象编程——类定义与对象

posted @ 2017-10-04 17:41 「大魔王」 阅读(213) 评论(0) 编辑 收藏

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](#)。

最新IT新闻:

- 刷屏被封背后, “新世相”们的隐藏逻辑
 - 外卖小哥送餐途中撞伤路人 法院: 平台赔20万余元
 - 谷歌离职员工: 谁能从谷歌手里偷走安卓?
 - CryptoKitties脱离母公司, 获1200万美元独立融资
 - 阅文宣布与搜狗订立综合合作协议
- » [更多新闻...](#)

最新知识库文章:

- 写给自学者的入门指南
 - 和程序员谈恋爱
 - 学会学习
 - 优秀技术人的管理陷阱
 - 作为一个程序员, 数学对你到底有多重要
- » [更多知识库文章...](#)

Copyright ©2018 「大魔王」