

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01 词典

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

基础教程介绍了基本概念, 特别是对象和类。

进阶教程对基础教程的进一步拓展, 说明Python的细节。希望在进阶教程之后, 你对Python有一个更全面的认识。

之前我们说了, 列表是Python里的一个类。一个特定的表, 比如说`nl = [1, 3, 8]`, 就是这个类的一个对象。我们可以调用这个对象的一些方法, 比如`nl.append(15)`。

我们要介绍一个新的类, **词典 (dictionary)**。与列表相似, 词典也可以**储存多个元素**。这种储存多个元素的对象称为**容器 (container)**。

基本概念

常见的创建词典的方法:

```
>>>dic = {'tom':11, 'sam':57,'lily':100}
>>>print type(dic)
```

词典和表类似的地方, 是包含有**多个元素**, 每个元素以逗号分隔。但词典的元素包含有两部分, **键和值**, 常见的是以字符串来表示键, 也可以使用数字或者真值来表示键 (不可变的对象可以作为键)。值可以是任意对象。键和值两者一一对应。

比如上面的例子中, `'tom'` 对应11, `'sam'`对应57, `'lily'`对应100

与表不同的是，词典的元素没有顺序。你~~不能通过下标引用元素~~。词典是~~通过键来引用~~。

```
>>>print dic['tom']
```

```
>>>dic['tom'] = 30
```

```
>>>print dic
```

构建一个新的空的词典：

```
>>>dic = {}
```

```
>>>print dic
```

在词典中增添一个新元素的方法：

```
>>>dic['lilei'] = 99
```

```
>>>print dic
```

这里，我们引用一个新的键，并赋予它对应的值。

词典元素的循环调用

```
dic = {'lilei': 90, 'lily': 100, 'sam': 57, 'tom': 90}
for key in dic:
    print dic[key]
```

在循环中，`dic`的每个键，被提取出来，赋予给`key`变量。

通过`print`的结果，我们可以再次确认，`dic`中的元素是没有顺序的。

词典的常用方法

```
>>>print dic.keys()           # 返回dic所有的键
>>>print dic.values()         # 返回dic所有的值
>>>print dic.items()          # 返回dic所有的元素（键值对）
>>>dic.clear()                # 清空dic，dict变为{}
```

另外有一个很常用的用法：

```
>>>del dic['tom']              # 删除 dic 的'tom'元素
```

`del`是Python中保留的关键字，用于删除对象。

与表类似，你可以用`len()`查询词典中的元素总数。

```
>>>print(len(dic))
```

总结

词典的每个元素是键值对。元素没有顺序。

```
dic = {'tom':11, 'sam':57,'lily':100}
```

```
dic['tom'] = 99
```

```
for key in dic: ...
```

```
del, len()
```

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基础10 反过头来看看](#)

» 下一篇: [Python进阶02 文本文件的输入输出](#)

posted @ 2012-06-06 10:40 Vamei 阅读(44304) 评论(22) 编辑 收藏

评论列表

#1楼 2012-06-10 09:33 长琴

键的数据类型是不可变的数据类型，不仅是字符串，数字也可以

[支持\(5\)](#) [反对\(0\)](#)

#2楼[楼主] 2012-06-19 08:35 Vamei

@ 长琴

是这样啊，谢谢你的提醒。我可以根据你说的修改我的博客么？还是说你更希望能别人能直接看到你的评论？

[支持\(1\)](#) [反对\(0\)](#)

#3楼 2013-02-08 16:39 峻祁连

@ Vamei

我越俎代庖了，当然你可以(最好这么做)根据评论修改原博客从而更确切，然后把原评论留在这里。

[支持\(0\)](#) [反对\(0\)](#)

#4楼[楼主] 2013-02-08 18:22 Vamei

@ 峻祁连

我已经是这么做了。

[支持\(0\)](#) [反对\(0\)](#)

#5楼 2014-04-22 15:28 Seandor

mark。key 必须是immutable。

[支持\(1\)](#) [反对\(0\)](#)

#6楼 2014-04-25 10:41 geekww

请问@Vamei

for key in dic: 这个用法。

```
1  dic={'name':33,'age':31,'xingbie':2}
2  for a in dic:
3      print a
```

结果是

age

name

xingbie

为什么只有Key而没有value?

谢谢!

支持(1) 反对(0)

#7楼 2014-05-05 14:41 Mountain_yang

@geekww

在循环中，dict的每个键，被提取出来，赋予给key变量。

支持(0) 反对(0)

#8楼 2014-05-16 16:32 死侍

膜拜大神 !!!

支持(0) 反对(0)

#9楼 2014-06-09 14:26 soube

0基础，英语最高33分，智商又比较捉急，很多概念自学起来很模糊，有没有哪位前辈空闲时间比较多，带我下？

支持(0) 反对(0)

#10楼 2014-08-16 00:34 11_xiao_7

楼主果然讲的很细致，有通俗易懂，赞一个！

支持(0) 反对(0)

#11楼 2014-09-03 09:24 乐芝木糖醇

@ geekww

结尾应该是 `print dic[a]`,才能返回值, `a`就是键

支持(1) 反对(0)

#12楼 2015-03-30 20:02 pythonerJ

@ Vamei

@长琴

关于key的类型问题,我尝试通过 如下方式一次赋值不可行:

```
>>> dic{1:2,3:4}
File "<stdin>", line 1
dic{1:2,3:4}
^
```

SyntaxError: invalid syntax

但是单独赋值key为数字,却又可以了。。

```
>>> dic[2]=100
>>> print dic[2]
100
```

那以后使用中通过dict来查询的时候,会不会歧义?

支持(0) 反对(0)

#13楼 2015-04-05 17:31 pythonier

键的排列顺序应该是默认的吧。

比如,如果都含有字母,则是以26个字母顺序排列先后。字母数字混合,则以第一个字母作为判别先后标准。只有数字就由小到大,纯数字在纯字母前等。等。等。

支持(0) 反对(0)

#14楼 2015-04-09 14:13 crayygy

你好

```
>>>del dic['tom'] # 删除 dic 的'tom'元素
```

这里似乎应该是删除dic中key 'Tom'所对应的key和value吧?

支持(0) 反对(0)

#15楼 2015-04-30 22:07 muki

mark

支持(0) 反对(0)

#16楼 2015-05-04 17:34 _hsin

@ pythonier

也不是这样啊，我加了一个键名首字母为 **a** 的键值对，但是遍历输出的时候还是在 **lilei** 这个键后面

支持(0) 反对(0)

#17楼 2015-05-30 20:52 迷途ing

此外 `dic[key]` 可以重复但是value会覆盖前面的

```
1  dic = {'lilei':90,'tom':50,'lily':80,'sam':70,'tom':60,'tom':100}
2  print len(dic)
3  print dic.keys()
4  print dic.values()
5  for key in dic:
6      print dic[key]
```

支持(0) 反对(0)

#18楼 2015-09-16 12:07 旒筱

在canopy里面试了一下，感觉输出是顺序的

```
dic = {'lilei': 90, 'lily': 100, 'sam': 57, 'tom': 90}
```

```
for key in dic:
```

```
print dic[key]
```

```
90
```

```
100
```

```
57
```

```
90
```

试了两次，都是顺序输出

支持(0) 反对(0)

#19楼 2015-09-30 23:04 蜗牛搬家x

貌似就是java你的 集合 Collection

支持(0) 反对(0)

#20楼 2015-11-21 17:37 Iamlunatic

@ 旒筱

对啊，和你同样的疑问。

支持(0) 反对(0)

#21楼 2016-04-06 14:15 wqh2016

少就是多 慢就是快

[支持\(0\)](#) [反对\(0\)](#)

#22楼 2016-05-04 11:54 目光与或

@ pythonerJ

你这里有个明显的错误就是dic{1:2,3:4}

应该写成这样dic = {1:2,3:4}

[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



SpreadJS 表格控件

Excel界面数据处理
可视化、可定制

[了解详情](#)

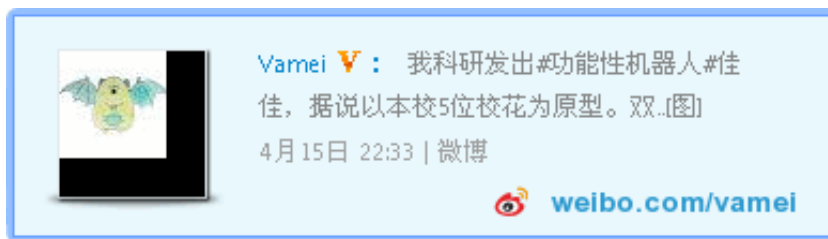


JPush 极光推送 消息推送领导品牌全面升级 JIGUANG | 极光

[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

05370293