

深入理解Python中字典的键的使用

字典的键

字典中的值没有任何限制，可以是任意Python对象，即从标准对象到用户自定义对象皆可，但是字典中的键是有类型限制的。

(1) 不允许一个键对应多个值

必须明确一条原则：每个键只能对应一个项。也就是说：一键对应多个值是不允许的（像列表、元组和其他字典这样的容器对象是可以的）。当有键发生冲突（即字典键重复赋值），取最后（最近）的赋值。Python并不会因字典中的键存在冲突而产生一个错误，它不会检查键的冲突是因为如果真这样做的话，在每个键-值对赋值的时候都会做检查，这将会占用一定量的内存。

```
>>> dict1 = {'foo':789, 'foo': 'xyz'}
>>> dict1
{'foo': 'xyz'}
>>> dict1['foo'] = 123
>>> dict1
{'foo': 123}
```

(2) 键必须是可哈希的

大多数Python对象可以作为键，但它们必须是可哈希的对象。像列表和字典这样的可变类型，由于它们不是可哈希的，所以不能作为键。

所有不可变的类型都是可哈希的，因此它们都可以做为字典的键。要说明的是：值相等的数字表示相同的键，即整型数字1和浮点数1.0的哈希值是相同的，它们是相同的键。

同时，也有一些可变对象（很少）是可哈希的，它们可以做字典的键，但很少见。举一个例子，一个实现了__hash__()特殊方法的类。因为__hash__()方法返回一个整数，所以仍然是用不可变的值（做字典的键）。

为什么键必须是可哈希的？解释器调用哈希函数，根据字典中键的值来计算存储你的数据的位置。如果键是可变对象，它的值可改变。如果键发生变化，哈希函数会映射到不同的地址来存储数据。如果这样的情况发生，哈希函数就不可能可靠地存储或获取相关的数据。选择可哈希的键的原因就是因为它们的值不能改变。

数字和字符串可以被用做字典的键，元组是不可变的但也可能不是一成不变的，因此用元组做有效的键必须要加限制：若元组中只包括像数字和字符串这样的不可变参数，才可以作为字典中有效的键。

示例：

```
# vi userpw.py
```

```
#!/usr/bin/env python
```

```
db = {}
```

```
def newuser():
```

```
    prompt= 'please regist your name: '
```

```
    while True:
```

```
        name = raw_input(prompt)
```

```
        if db.has_key(name):
```

```
            prompt = 'name taken,try another: '
```

```
            continue
```

```
        else:
```

```
            break
```

```
        pwd = raw_input('passswd: ')
```

```
        db[name] = pwd
```

```
        print 'Newuser [%s] has added successfully!' %name
```

```
def olduser():
```

```
    name = raw_input('login: ')
```

```
    pwd = raw_input('passwd: ')
```

```

passwd = db.get(name)
if passwd == pwd:
    print 'welcome back',name
else:
    print 'login incorrect!'
def showmenu():
    prompt = ""
    (N)ew User Login
    (E)xisting User Login
    (Q)uit
    Enter choice: ""
    while True:
        try:
            choice = raw_input(prompt).strip()[0].lower()
            print "\nYou picked: [%s]" % choice
            if choice not in 'neq':
                print 'invalid option,please try again'
            if choice == 'n':
                newuser()
            if choice == 'e':
                olduser()
            if choice == 'q':
                break
        except(EOFError,KeyboardInterrupt):
            print 'invalid option,please try again'
    if __name__ == '__main__':
        showmenu()

```

您可能感兴趣的文章:Python字典循环添加一键多值的用法实例Python字典中的键映射多个值的方法(列表或者集合)Python实现的字典值比较功能示例在Python 字典中一键对应多个值的实例