Python核心编程学习笔记之映射类型（上）

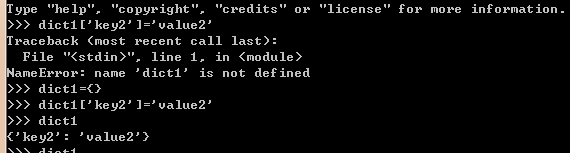
1. Python中字典的定位：
   1. 字典是python中唯一的映射类型，通常被认为是可变的哈希表。
   2. 字典对象是可变的，能存储任意多个python对象。
   3. 字典是Python中最强大的数据类型之一
2. 字典(dict)和序列类型容器类（列表和元组）的区别：
   1. 存储和访问数据的方式不同，序列类型只用数字类型的键，映射可以使用其他对象类型作为键，比如数字和字符串，也可以混用数字和字符串作为键；
   2. 字典中的数据是无需排列的，序列中的数据时有序的。
3. 如何得到dict的有序集合：

字典(dict)中的有序集合只能是键的集合或者值得集合。方法keys()或value()返回一个列表，是可排序的。还可以使用items()得到包含键值对的元组的列表排序

1. 字典（dict）的语法：

语法格式： 键：值 一个键值对成为一个元素

* 1. >>>dict1={} 🡪创建一个空值的字典
  2. >>>dict1={‘key’:’value’} 🡪设置字典的键值对
  3. >>>dict1={‘key1’:’value1’} 🡪 这样dict1又重新映射到{‘key1’:’value1’}。{‘key’:’value’}就作废了。
  4. >>>dict1[‘key2’]=’value2’ 🡪 如果字典中不存在该键就添加新的键值对，存在该键就会替换掉原来的值。如果dict1还未赋过值，则这句话会报错



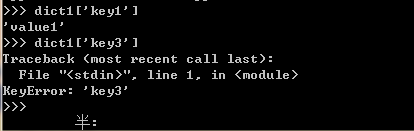
* 1. >>>del dict1[‘key2’] 🡪删除字典对象中键为“key2”的条目
  2. >>>dict1.clear() 🡪清空dict1中的所有内容
  3. >>> del dict1 🡪删除dict1
  4. >>>dict1.pop(‘key1’) 🡪删除并返回键为‘key1’的值

笔记提醒：

避免使用内建的对象名字作为变量的标示符。虽然编译器允许，但是你能确定你不出错吗？

1. Python怎么查找值呢？

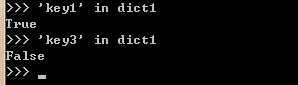
对了通过键查找，就像这样dict1[‘key1’],当然你得有dict1这个对象并且还得有键”key1”才行哦！！否则有错哦！！



好像还有件事，我怎么确定字典中是否有”key1”或者”key3”键呢？

差点忘了，看这

>>>’key1’ in dict1 🡪存在”key1”键，返回True，否则就返回False

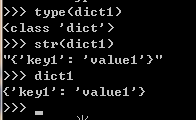


1. 字典的内建函数和工厂函数

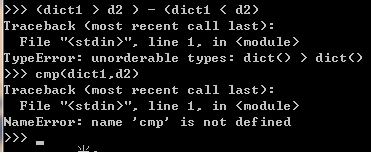
标准类型函数【type()、str()】

type()和str()比较简单，type(对象) 🡪 返回对象类型

str(对象) 🡪返回字符串！看图知道有啥区别了吧



PS：cmp()在3.x中不支持，网传的（a>b）- (a<b) 出错，暂时跳过，等待中



字典对象间比较顺序为：字典长度🡪字典键🡪字典值，全都相同时，返回0值。

【dict()、len()】

dict() 工厂函数被用来创建字典。如果不提供参数，会生成空字典。当容器类型对象做为一个参数传 递给方法 dict() 时很有意思。如果参数是可以迭代的，即，一个序列，或是一个迭代器，或是一个 支持迭代的对象，那每个可迭代的元素必须成对出现。在每个值对中，第一个元素是字典的键、第二个元素是字典中的值。见 Python 文档里关于 dict()的例子:

>>> dict(zip(('x', 'y'), (1, 2)))  
{'y': 2, 'x': 1}  
>>> dict([['x', 1], ['y', 2]])  
{'y': 2, 'x': 1}  
>>> dict([('xy'[i-1], i) for i in range(1,3)])  
{'y': 2, 'x': 1}

len() 可以用在序列、映像和集合上，对字典调用len()会返回所有元素的数据。

Python核心编程学习笔记之映射类型（中）

1. 字典中键的特点：
2. 不允许一个键对应多个值
3. 键必须是可哈希的，大部分Python对象都可以作为键，但必须是可以哈希的对象，像列表和字典这样的可变类型，由于是不可哈希的所以不能作为键。

值相等的数字的哈希值是相同的，表示相同的键。

1. 为什么键必须是可哈希的呢？

解释器调用哈希函数，根据字典中的键的幌子来计算存储你的数据的位置。

1. 集合类型：可变集合(set) 和 不可变集合(frozenset)

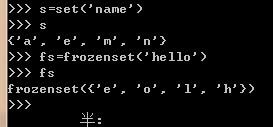
可变集合可以添加和删除元素，不可哈希，因此不能作为字典的键或其他集合的元素，

不可变集合不允许这样做，是可哈希的，能被用作字典的键或者集合的元素

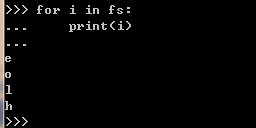
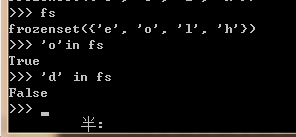
1. 集合的创建

>>>s=set(‘name’) 🡪 创建集合

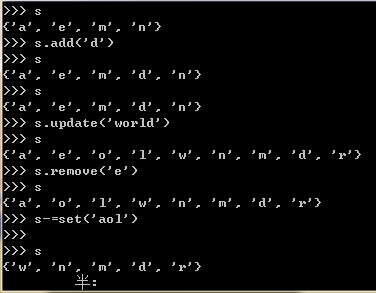
>>>fs=frozenset(‘hello’) 🡪 创建不可变集合



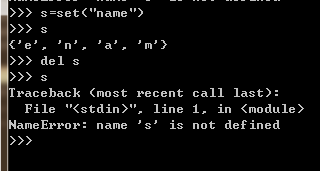
1. 访问集合中的值：



1. 更新集合

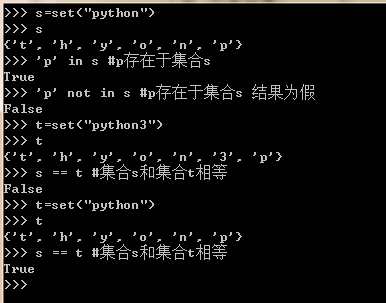


1. 删除集合



|  |  |
| --- | --- |
| Python符号 | 说明 |
| in | 是…的成员 |
| not in | 不是…的成员 |
| == | 等于 |
| != | 不等于 |
| < | 是..的真子集 |
| <= | 是…的子集 |
| > | 是…的真超集 |
| >= | 是…的超集 |
| & | 交集 |
| | | 合集 |
| - | 差补或相对补集 |
| ^ | 对称差分 |

集合中以上各符号的使用



如果p元素存在于集合中，用in返回结果为True，not in返回结果为False。判断两个集合是否相等时用“==”,判断标准是两个集合中的元素个数和元素都相同