

set的使用

集合 (set) 是一个**无序的不重复元素**序列，可以使用大括号 { } 或者 **set()** 函数创建集合。

注意：创建一个空集合必须用 **set()** 而不是 { }，因为 { } 是用来创建一个空字典。

创建格式：

```
parame = {value01,value02,...}  
或者  
set(value)
```

添加元素

语法格式如下：

```
s.add(x)
```

将元素 x 添加到集合 s 中，如果元素已存在，则不进行任何操作。

```
>>>thisset = set(("Google", "Runoob", "Taobao"))  
>>> thisset.add("Facebook")  
>>> print(thisset)  
{'Taobao', 'Facebook', 'Google', 'Runoob'}
```

还有一个方法，也可以添加元素，且参数可以是列表，元组，字典等，语法格式如下：

```
s.update( x )
```

x 可以有多个，用逗号分开。

```
>>>thisset = set(("Google", "Runoob", "Taobao"))  
>>> thisset.update({1,3})  
>>> print(thisset)  
{1, 3, 'Google', 'Taobao', 'Runoob'}  
>>> thisset.update([1,4],[5,6])  
>>> print(thisset)  
{1, 3, 4, 5, 6, 'Google', 'Taobao', 'Runoob'}
```

移除元素

语法格式如下：

```
s.remove( x )
```

将元素 x 从集合 s 中移除，如果元素不存在，则会发生错误。

```
>>>thisset = set(("Google", "Runoob", "Taobao"))
>>> thisset.remove("Taobao")
>>> print(thisset)
{'Google', 'Runoob'}
>>> thisset.remove("Facebook")    # 不存在会发生错误
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'Facebook'
>>>
```

此外还有一个方法也是移除集合中的元素，且如果元素不存在，不会发生错误。格式如下所示：

```
s.discard( x )
```

```
>>>thisset = set(("Google", "Runoob", "Taobao"))
>>> thisset.discard("Facebook")    # 不存在不会发生错误
>>> print(thisset)
{'Taobao', 'Google', 'Runoob'}
```

我们也可以设置随机删除集合中的一个元素，语法格式如下：

```
s.pop()
```

```
thisset = set(("Google", "Runoob", "Taobao", "Facebook"))
x = thisset.pop()

print(x)
print(thisset)
```

set常见方法列表

方法	描述
add()	为集合添加元素
clear()	移除集合中的所有元素
copy()	拷贝一个集合
pop()	随机移除元素
remove()	移除指定元素
union	返回两个集合的并集
update()	给集合添加元素
difference()	返回多个集合的差集
difference_update()	移除集合中的元素，该元素在指定的集合也存在。
discard()	删除集合中指定的元素
intersection()	返回集合的交集
intersection_update()	删除集合中的元素，该元素在指定的集合中不存在。
isdisjoint()	判断两个集合是否包含相同的元素，如果没有返回 True，否则返回 False。
issubset()	判断指定集合是否该方法参数集合的子集。
issuperset()	判断该方法的参数集合是否为指定集合的子集
symmetric_difference()	返回两个集合中不重复的元素集合。
symmetric_difference_update()	移除当前集合中在另外一个指定集合相同的元素，并将另外一个指定集合中不同的元素插入到当前集合中。

练习

有一个无序且元素数据重复的列表nums, `nums=[5,8,7,6,4,1,3,5,1,8,4]` ,要求对这个列表里的元素去重，并进行降序排序。

```
# 方法一：调用列表的sort方法
nums2 = list(set(nums))
nums2.sort(reverse=True)
print(nums2)

# 方法二：使用sorted内置函数
print(sorted(list(set(nums)),reverse=True))
```

千锋Python人工智能学院