**1、集合set**

(1)概念

#集合是可变的容器，集合内的数据对象都是唯一的

#集合是无序的存储结构，集合中的数据没有先后顺序

#元素必须是不可变的对象，相当于只有键没有值的字典（键则是集合的数据）

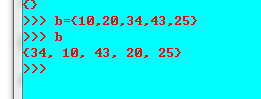
（2）表示方法

#空集合

set()

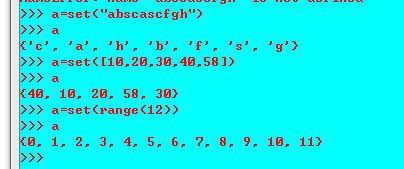
#非空集合

集合里的元素是无序的，且不能重复



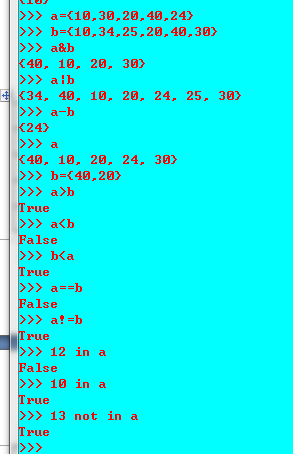
（3）集合的构造函数

set(iterable)可用迭代对象创建一个新的集合对象



（4）集合的运算

|  |  |
| --- | --- |
| 运算符 | 描述 |
| & | 生成两个集合的交集 |
| | | 生成两个集合的并集 |
| - | 生成两个集合的补集a-b（生成属于a但是不属于b的集合） |
| > | 判断一个集合是另一个集合的超集（布尔类型） |
| < | 判断一个集合是另一个集合的子集 |
| == != | 判断两个集合是否相同或者不同 |
| in not in | 判断一个值在不在集合中 |
| ^ | 对称补集（存在于a或b中的元素，又不同时存在于a和b中）0 |



练习:

一个公司中：

经理有：张三 李四 王二

技术员：张三 王二 小刘 小张

求：

1、既是经理也是技术员是？

2、是经理，但不是技术员是？

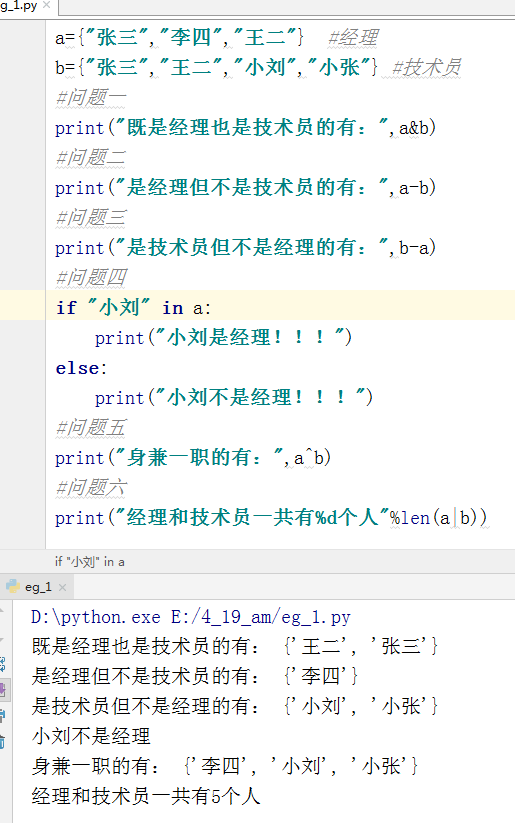
3、是技术员，但不是经理的是？

4、小刘是经理么？

5、身兼一职的有哪些？

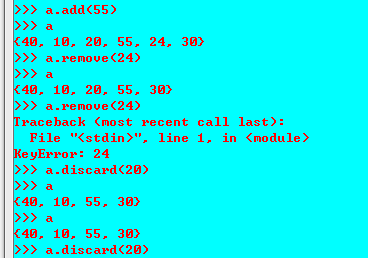
6、经理和技术员一共多少个人？

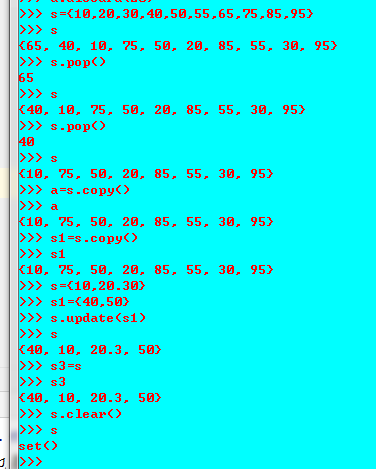
结果如下：



（5）集合的常用方法

|  |  |
| --- | --- |
| 方法 | 描述 |
| s.add(e) | 在集合中添加一个元素e，如果元素已存在就不添加 |
| s.remove(e) | 在集合中删除一个元素e，如果元素不已存在就不删除，将会发生一个keyerror的错误 |
| s.discard(e) | 删除指定对象，一次只能删除一个，如果对象不已存在时，不报错 |
| s.pop() | 从集合S中随机删除一个元素，如果此集合为空，则发出keyerror错误 |
| s.copy() | 将集合进行一次浅拷贝 |
| s.update(s1) | 用于S与S1的得到并更新集合 |
| s.clear() | 清空集合 |

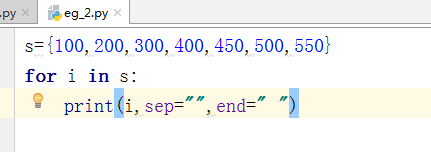




（6）集合的遍历

#格式

for 变量 in 非空集合



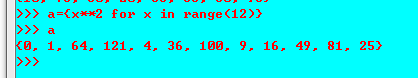
（7）集合的推导式

#语法

{表达式 for x in 可迭代对象 if 真值表达式}

#作用

用可迭代对象创建集合



**2、固定集合(frozenset)**

（1）概念

固定集合是不可变的，无序的，含有一定元素的集合

（2）作用

固定集合可以作为字典的键，也可以作为集合的值

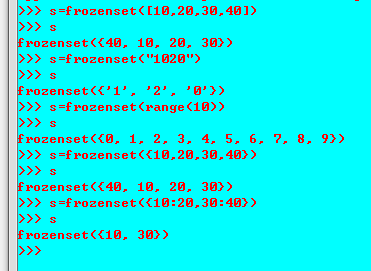
（3）创建空的固定集合

frozenset()



（4）创建非空的固定集合

frozenset(iterable)



（5）固定集合中的运算

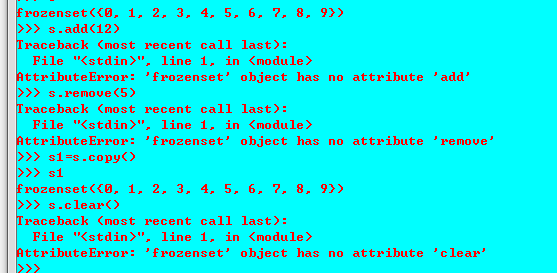
等同于集合中的用法

（6）固定集合的方法

相当于集合的全部方法去掉修改集合的方法

固定集合可以作为集合的值，也可以作为字典的键

集合中常用的方法都不可以使用（例如：增删改查），只能使用复制（s.copy）



**3、元组**

（1）概念

#元组是不可变序列

#元组是可以存放任意元素的，他们之间有先后顺序

（2）表示方法

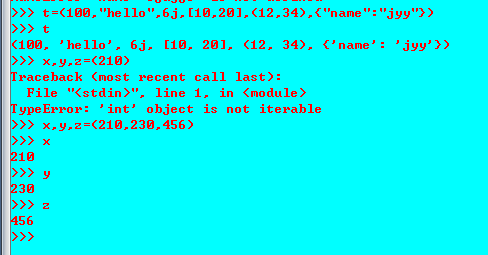
用小括号()括起来，单个元素括起来后加逗号，区分是单个对象还是元组

#空元组

t=()

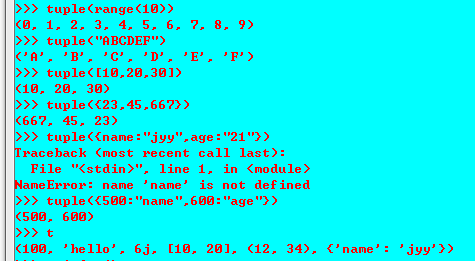
t=tuple()

#非空元组



（3）元组的构造函数

tuple(iterable)



（4）元组的运算

+ += \*\* = < <= > >= == != in not in

使用方法和列表完全相同

（5）元组的索引与切片

元组的索引与切片规则与列表的完全相同，但是由于元组是不可变序列，所以不具备索引复制与切片赋值规则

如果想对元组进行更改操作，可以先把元组转换成列表，操作完成之后再转换换回元组

（6）元组的常用方法

|  |  |
| --- | --- |
| 方法 |  |
| t.index(v,[begin],[end]) | 返回对应元素下标索引，元素不存在时会触发valueError |
| t.count() | 返回元组中对应元素的个数 |

