1. 字典的形式为：{key1:item1, key2:item2,key3:item3…)}，例：score = {'Michael': 95, 'Bob': 75, 'Tracy': 85}，

1）创建字典：直接使用{}，或者使用dict()，

方法一：

temp = dict(name='xiaoming', age=18)

方法二：

temp = dict((('name', 'xiaoming'), ('age', 18)))

方法三：dict() zip() 函数联合使用

temp = dict(zip(['name', 'age'], ['xiaoming', 18]))

2）获取数据，dict\_name[key\_name]，如果key不存在会报错KeyError

3）使用get函数获取key对应的value，dict\_name.get(key\_name)，如果key不存在，则返回空

4）字典新增项， dict\_name[key\_name] = value

5）删除项，dict\_name.pop(key\_name)，删除键对应的值，但它会把对值的应输出后再删除

clear方法：清空字典内容，变量名.clear()

del方法：删除键对应的值，del空格 变量名[键名]，如果只写变量名就是删除此字典

6）返回字典的key值或者value值keys()，values()

7）遍历字典，items()，获取所有键值对

2. 集合的定义：集合里面的元素是不可重复的，s = {1,2,3,1,2,3,4,5}

例如：

s2 = {}

print(type(s2)) #此时的s2是一个字典（dict）

s3 = set([]) #正确定义空集合

集合支持的特性只有：成员操作符

s = {1,2,3}

print(1 in s) # True

print(1 not in s) # False

可以用for循环遍历集合

s = {1,2,3}

for i in s:

print(i,end = '|')

print('')

for a,b in enumerate(s): #将索引值及元素分别赋给a,b

print('index: %s,value: %s'%(a,b))

**常用方法：**

集合是无序的数据类型，添加顺序和在集合中存储的顺序是不一定相同的，增加一个元素到集合中add(value)，例如：

s = {6,7,8,9}

s.add(1)

print(s)

结果：

{1, 6, 7, 8, 9}

增加多个元素到集合中：update()

s = {6,7,8,9}

s.update({2,3,5})

print(s)

结果：

{2, 3, 5, 6, 7, 8, 9}

删除pop()

删除指定元素：remove

集合的并集：union或|

交集：intersection或&

差集：difference

对等差分:并集 - 交集,symmetric\_difference()

判断是否是超集：issuperset()

s3 = {1,2}

s4 = {1,2,3}

#s3是否为s4的超集

#超集：如果s3中的每一个元素都在集合s4中，且s4中可能包含s3中没有的元素

#那么s4就是s3的一个超集

print(s3.issuperset(s4))

print(s4.issuperset(s3))

结果：

False

True

判断子集：issubset()

isdisjoint()用来判断两个集合是否包含相同元素,没有返回True,否则返回False

3.判断语句（要求掌握多条件判断）

if 判断条件1:

执行语句1……

elif 判断条件2:

执行语句2……

elif 判断条件3:

执行语句3……

else:

执行语句4……

如果判断需要多个条件需同时判断时，可以使用 or （或），表示两个条件有一个成立时判断条件成功；使用 and （与）时，表示只有两个条件同时成立的情况下，判断条件才成功

4.三目表达式

a = 1

b = 2

h = ""

h = a-b if a>b else a+b

print(h)

5.循环语句

for iterating\_var in sequence:

statements(s)

在 python 中，for … else 表示这样的意思，for 中的语句和普通的没有区别，else 中的语句会在循环正常执行完（即 for 不是通过 break 跳出而中断的）的情况下执行，while … else 也是一样。

while 判断条件：

执行语句……

while 语句时还有另外两个重要的命令 continue，break 来跳过循环，continue 用于跳过该次循环，break 则是用于退出循环，此外"判断条件"还可以是个常值，表示循环必定成立