1、元组tuple，和列表类似，不同之处

a. 元组的元素不能修改，列表可以

b. 元组使用小括号，列表使用中括号

2、元组的基本操作，和字符串类型，索引下标从0开始，可以进行切片、组合等操作

算术运算 + += \* \*=

比较运算 > < >= <= != ==

in、not in 也支持

切片 也支持

切片赋值 不支持，因为元组的不可变对象，

可以通过索引去访问元组的元素，但是不可以改变

算术运算时，必须是元组和元组之间的操作

不能删除元组的元素内容，但是可以使用del语句来删除整个元组

3、列表的排序

3.1 sorted(L) 临时排序，对原列表的内容补改变，返回排序之后的元素组成的新列表，默认情况是升序排列，使用关键字传参的方式reverse=true可以指定降序排列

3.2 L.sort() 永久排序，对源列表的元素内容做修改，也可以使用关键字传参的方式reverse=true指定降序排列

3.3 L.reverse() 反转

对于元组是无法直接进行排序，可以将元组转换程列表，再进行排序

4、字典

通过下标来处理列表的元素时，容易出错，工作量比较大，当存储相同的数据时，使用列表或者元组比较合适,但是对于类型不一致的元素时，使用列表或元组就不适合了。

4.1 字典是一种可变的容器，可以存储任意类型的数据

4.2 字典中的每一个数据都用键(key)进行索引，而不是像序列那样使用下标

4.3 字典中的元素没有先后顺序，字典的存储是无序的

4.4 字典的数据都是以key-value键值对的方式进行的存储映射

4.5 字典的key值不能重复，且只能是不可变类型作为字典的键

5、创建字典的方式

dict = {"name":"郭德纲","age":55,"gender":True}

dict()

dict(可迭代对象) 可迭代对象的每一个元素，会被创建程一个键值对

dict(\*\*kwargs) 关键字传参

6、字典的增删改查（info是自定义的字典名字）

info = {}

info[“name”] = “张三” 增加

info[“name”] = “李四” 改

del info[“name”] 删

info.get(“name”) 查

7、字典的其他操作

len(d) 键值的个数

Info.keys() 返回所有的键值组成的列表

Info.values() 返回所有的value值组成的列表

Info.items() key和value封装成一个元组，若干个元组封装成一个列表返回