字符串、列表、字典、元组、集合

1、什么是函数：是可以重复执行的语句块，独立的代码相当于一个整体。

作用：封装语句块，提高重用性

定义用户级别的函数

2、语法：def 函数名(参数列表):

语句块

说明：函数的名字的命名规则和变量相同(小写，单词之间用\_)

函数有自己的命名空间，在函数的外部不可以访问函数内部的变量，在函数外部可以访问函数外部的全局变量

函数的参数列表可以为空

语句部分也可以为空，但是要补充pass语句

3、函数的调用

函数的调用是一个表达式

如果函数中没有return语句，此函数调用完毕后返回None

4、return语句

函数中遇到return语句的时候，就不继续往下执行，当需要返回多个数据的时候，可以用列表等数据类型进行返回

练习：定义一个函数，接收两个整数，返回其中较大的那个数

5、函数的嵌套：一个函数中调用了另一个函数

练习：

1、写一个函数，此函数显示两个参数的相关信息：最大值、和值、乘积，从第一个数到第二个数之间所有的偶数

2、猴子吃桃问题，有一个猴子摘了很多桃，第一天：吃了一半，又多吃一个

第二天：吃了一半，又多吃一个

… …

第10天时,发现只有一个了，问第一天一个摘了多少。

3、完全数，6=1+2+3，求1000以内的所有完全数 28=1+2+4+7+14

6、全局变量和局部变量

当函数内部的局部变量和外部的全局变量重名的时候，在函数内部使用的实际上是局部变量，在函数内部如果像修改那个全局变量，实际上是修改了重名的局部变量。

解决办法：在修改全局变量的位置加一个关键字，global，

对于字典、列表等数据类型，在函数中修改其元素内容，不加global，也是修改的全局变量，因为仅仅修改了其元素内容，如果对于整个字典、列表进行修改，就需要加上global。

7、Python函数调用时，参数的传递规则

7.1位置传参

实参(实际参数)和形参的对应关系按照位置依次进行匹配

7.2序列传参

在函数的调用过程中，使用\*将序列拆散后再按照位置传参的方式进行参数传递

7.3关键字传参

传参的时候，使用形参的关键字进行传递

形参进行匹配时，其顺序可以不受约束，只要保证，每一个形参都被匹配到就可以。

7.4字典关键字传参

实参为字典，使用\*\*拆解后，进行关键字传参

字典的key要和形参保持一致

字典的key要是字符串

综合使用：再能确定每一个形参都能被唯一匹配到的情况下，可以任意结合使用

8、Python函数定义时，参数列表的定义规则