1.变量：每个变量都存储了一个值 ——与变量相关联的信息。

2.字符串：就是一系列字符。在Python中，用引号括起的都是字符串，其中的引号可以是单引号，也可以是双引号。

3.Python将带小数点的数字都称为浮点数 。

4. 列表：由一系列按特定顺序排列的元素组成。

5.元组：Python将不能修改的值称为不可变的，而不可变的列表被称为元组。

6. 每条if 语句的核心都是一个值为True 或False 的表达式，这种表达式被称为条件测试 。

True :真

False: 假

7.使用”and”检查多个条件。

要检查是否两个条件都为True ，可使用关键字and 将两个条件测试合而为一；如果每个测试都通过了，整个表达式就为True ；如果至少有一个测试没有通过，整个表达式就为False 。

8. 使用or 检查多个条件。

关键字or 也能够让你检查多个条件，但只要至少有一个条件满足，就能通过整个测试。仅当两个测试都没有通过时，使用or 的表达式才为False。

9. 使用"not in"检查特定值是否不包含在列表中。

10. 布尔表达式 ，它不过是条件测试的别名。与条件表达式一样，布尔表达式的结果要么为True ，要么为False 。

11. 在Python中，字典是一系列键—值对。每个键都与一个值相关联，你可以使用键来访问与之相关联的值。与键相关联的值可以是数字、字符串、列表乃至字典。

键—值 对是两个相关联的值。指定键时，Python将返回与之相关联的值。键和值之间用冒号分隔，而键—值对之间用逗号分隔。

12. 要立即退出while 循环，不再运行循环中余下的代码，也不管条件测试的结果如何，可使用break 语句。

13. 要返回到循环开头，并根据条件测试结果决定是否继续执行循环，可使用continue 语句，它不像break 语句那样不再执行余下的代码并退出整个循环。

14. 函数是带名字的代码块，用于完成具体的工作。

要执行函数定义的特定任务，可调用该函数。需要在程序中多次执行同一项任务时，你无需反复编写完成该任务的代码，而只需调用执行该任务的函数，让Python运行其中的代码。

15. 函数返回的值被称为返回值。return 语句将值返回到调用函数的代码行。

语法：

16. 类中的函数称为方法。

xxx.upper() #大写

xxx.lower() #小写

xxx.title() #首字母大写

剔除空白：

xxx.lstrip() #开头空白

xxx.rstrip() #末尾空白

xxx.strip() #两端空表

str(xxx) #避免类型错误

xxx.insert() #可在列表的任何位置添加新元素

xxx.append() #将元素添加到列表末尾

del xxx[0] #删除任意位置元素

xxx.pop() #删除列表末尾的元素

xxx.pop(0) #删除列表任意位置的元素

remove(xxx) #删除元素的值

xxx.sort() #按字母顺序排列

xxx.sort(reverse=True) #相反顺序排序

xx.reverse() #按相反的顺序排列,倒着打印列表

len(xxx) #确认列表长度

for xxx in range(1,5) #生成一系列的数字

#使用函数list() 将range() 的结果直接转换为列表

xxx = list(range(1,6))

items() #返回一个键—值对列表。

keys() #不需要使用字典中的值时使用

sorted() #以特定的顺序返回元素，按特定顺序排列的键列表的副本

values() #它返回一个值列表，而不包含任何键。

set() #去除重复的元素

remove() #删除列表中的特定值

python数据类型：

基本类型：

数字：

整数：int

长整型：long （python3无！）

浮点型：float

字符串：

文本：str

字节：bytes

布尔：True/False

数据集：

列表：list

元组：tuple

字典：dict (有序字典,无序字典）

集合：set (有序集合,无序集合）

>（大于）

<(小于)

==（等于）

>=（大于等于）

<=（小于等于）