Python是一种计算机程序设计语言。

代码少的代价是运行速度慢。

Python是用来编写应用程序的高级编程语言。

Python就为我们提供了非常完善的基础代码库，覆盖了网络、文件、GUI、数据库、文本等大量内容，被形象地称作“内置电池（batteries included）”。用Python开发，许多功能不必从零编写，直接使用现成的即可。

除了内置的库外，Python还有大量的第三方库，也就是别人开发的，供你直接使用的东西。当然，如果你开发的代码通过很好的封装，也可以作为第三方库给别人使用。

Python适合开发哪些类型的应用呢？首选是网络应用，包括网站、后台服务等等；其次是许多日常需要的小工具，包括系统管理员需要的脚本任务等等；另外就是把其他语言开发的程序再包装起来，方便使用。

Python的定位是“优雅”、“明确”、“简单”，所以Python程序看上去总是简单易懂。

Python有哪些缺点呢？第一个缺点就是运行速度慢，和C程序相比非常慢，因为Python是解释型语言，你的代码在执行时会一行一行地翻译成CPU能理解的机器码，这个翻译过程非常耗时，所以很慢。而C程序是运行前直接编译成CPU能执行的机器码，所以非常快。 第二个缺点就是代码不能加密。如果要发布你的Python程序，实际上就是发布源代码，这一点跟C语言不同，C语言不用发布源代码，只需要把编译后的机器码（也就是你在Windows上常见的xxx.exe文件）发布出去。要从机器码反推出C代码是不可能的，所以，凡是编译型的语言，都没有这个问题，而解释型的语言，则必须把源码发布出去。

安装Python：安装后，你会得到Python解释器（就是负责运行Python程序的），一个命令行交互环境，还有一个简单的集成开发环境。

当我们从Python官方网站下载并安装好Python 3.5后，我们就直接获得了一个官方版本的解释器：CPython。这个解释器是用C语言开发的，所以叫CPython。在命令行下运行python就是启动CPython解释器。

请注意区分命令行模式和Python交互模式。

有同学问，能不能像.exe文件那样直接运行.py文件呢？在Windows上是不行的，但是，在Mac和Linux上是可以的，方法是在.py文件的第一行加上一个特殊的注释：#!/usr/bin/env python3。然后，通过命令给hello.py以执行权限：$ chmod a+x hello.py。

输入和输出

直接输入python进入交互模式，相当于启动了Python解释器，但是等待你一行一行地输入源代码，每输入一行就执行一行。直接运行.py文件相当于启动了Python解释器，然后一次性把.py文件的源代码给执行了，你是没有机会以交互的方式输入源代码的。

print()函数可以接受多个字符串，用逗号“,”隔开，就可以连成一串输出，print('The quick brown fox', 'jumps over', 'the lazy dog')。print()会依次打印每个字符串，遇到逗号“,”会输出一个空格。

Python提供了一个input()，可以让用户输入字符串，并存放到一个变量里。比如输入用户的名字：>>> name = input()。input()可以让你显示一个字符串来提示用户，于是我们把代码改成：name = input('please enter your name: ')

在计算机程序中，变量不仅可以为整数或浮点数，还可以是字符串。

我们把输入输出统称为Input/Output，或者简写为IO。input()和print()是在命令行下面最基本的输入和输出。用户也可以通过其他更高级的图形界面完成输入和输出，比如，在网页上的一个文本框输入自己的名字，点击“确定”后在网页上看到输出信息。

编译器或者解释器就是负责把符合语法的程序代码转换成CPU能够执行的机器码，然后执行。

以#开头的语句是注释，注释是给人看的，可以是任意内容，解释器会忽略掉注释。其他每一行都是一个语句，当语句以冒号:结尾时，缩进的语句视为代码块。Python使用缩进来组织代码块，请务必遵守约定俗成的习惯，坚持使用4个空格的缩进。

Python程序是大小写敏感的，如果写错了大小写，程序会报错。

数据类型和变量

不同的数据，需要定义不同的数据类型。在Python中，能够直接处理的数据类型有以下几种：**整数**，**浮点数**也就是小数，**字符串**。

转义字符\可以转义很多字符，比如\n表示换行，\t表示制表符，字符\本身也要转义，所以\\表示的字符就是\。为了简化，Python还允许用r''表示''内部的字符串默认不转义。

如果字符串内部有很多换行，用\n写在一行里不好阅读，为了简化，Python允许用'''...'''的格式表示多行内容。

**布尔值**，只有True、False。布尔值可以用and、or和not运算。

**空值**是Python里一个特殊的值，用None表示。

此外，Python还提供了**列表、字典等**多种数据类型，还允许创建自定义数据类型。

变量不仅可以是数字，还可以是任意数据类型。变量在程序中就是用一个变量名表示了，变量名必须是大小写英文、数字和\_的组合，且不能用数字开头。

变量本身类型不固定的语言称之为动态语言a = 123，与之对应的是静态语言。静态语言在定义变量时必须指定变量类型int a = 123。

Python支持多种数据类型，在计算机内部，可以把任何数据都看成一个“对象”，而变量就是在程序中用来指向这些数据对象的，对变量赋值就是把数据和变量给关联起来。

字符串和编码

计算机只能处理数字，如果要处理文本，就必须先把文本转换为数字才能处理。

8个比特（bit）（8位）作为一个字节（byte）。ASCII [ˈæski]编码是1个字节，而Unicode编码通常是2个字节。把Unicode编码转化为“可变长编码”：UTF-8编码。在计算机内存中，统一使用Unicode编码，当需要保存到硬盘或者需要传输的时候，就转换为UTF-8编码。用记事本编辑的时候，从文件读取的UTF-8字符被转换为Unicode字符到内存里，编辑完成后，保存的时候再把Unicode转换为UTF-8保存到文件。浏览网页的时候，服务器会把动态生成的Unicode内容转换为UTF-8再传输到浏览器。

在最新的Python 3版本中，字符串是以Unicode编码的，由于Python的字符串类型是str，在内存中以Unicode表示，一个字符对应若干个字节。如果要在**网络上传输，或者保存到磁盘**上，就需要把str变为以字节为单位的bytes。Python对bytes类型的数据用带b前缀的单引号或双引号表示：x = b'ABC'。要注意区分'ABC'和b'ABC'，前者是str，后者虽然内容显示得和前者一样，但bytes的每个字符都只占用一个字节。以Unicode表示的str通过**encode()**方法可以编码为指定的bytes。反过来，如果我们**从网络或磁盘上读取**了字节流，那么读到的数据就是bytes。要把bytes变为str，就需要用**decode()**方法。

在Python中，采用的格式化方式和C语言是一致的，用%实现，%运算符就是用来格式化字符串的。在字符串内部，%s表示用字符串替换，%d表示用整数替换，有几个%?占位符，后面就跟几个变量或者值，顺序要对应好。如果只有一个%?，括号可以省略。%f 浮点数；

%x 十六进制整数。其中，格式化整数和浮点数还可以指定是否补0和整数与小数的位数。如果你不太确定应该用什么，%s永远起作用，它会把任何数据类型转换为字符串。转义用%%来表示一个%。

使用list和tuple

Python内置的一种数据类型是列表：list。list是一种有序的集合，可以随时添加和删除其中的元素。用len()函数可以获得list元素的个数。用索引来访问list中每一个位置的元素，记得**索引是从0开始的**。

最后一个元素的索引是len(classmates) - 1。如果要取最后一个元素，除了计算索引位置外，还可以用-1做索引，直接获取最后一个元素。以此类推，可以获取倒数第2个、倒数第3个。可以往list中追加元素到末尾：classmates.append('Adam')。也可以把元素插入到指定的位置，比如索引号为1的位置：>>> classmates.insert(1, 'Jack')。要删除list末尾的元素，用pop()方法：>>> classmates.pop()。要删除指定位置的元素，用pop(i)方法，其中i是索引位置。

另一种有序列表叫元组：tuple。tuple和list非常类似，但是**tuple一旦初始化就不能修改**，也没有append()，insert()这样的方法。其他获取元素的方法和list是一样的，你可以正常地使用classmates[0]，classmates[-1]，但不能赋值成另外的元素。因为tuple不可变，所以代码更安全。如果可能，能用tuple代替list就尽量用tuple。只有1个元素的tuple定义时必须加一个逗号,，来消除歧义：>>> t = (1,)

list和tuple是Python内置的有序集合，一个可变，一个不可变。根据需要来选择使用它们。

条件判断

if语句执行有个特点，它是从上往下判断，如果在某个判断上是True，把该判断对应的语句执行后，就忽略掉剩下的elif和else。

input()返回的数据类型是str。

循环

Python的循环有两种，一种是for...in循环，依次把list或tuple中的每个元素迭代出来，for x in ...循环就是把每个元素代入变量x，然后执行缩进块的语句。

Python提供一个range()函数，可以生成一个整数序列，再通过list()函数可以转换为list。比如range(5)生成的序列是从0开始小于5的整数：>>> list(range(5))

第二种循环是while循环，只要条件满足，就不断循环，条件不满足时退出循环。在循环中，**break语句**可以提前退出循环。在循环过程中，也可以通过**continue语句**，跳过当前的这次循环，**直接开始下一次循环**。

使用dict和set

Python内置了字典：dict的支持，dict全称dictionary，在其他语言中也称为map，使用键-值（key-value）存储，具有极快的查找速度。把数据放入dict的方法，除了初始化时指定（>>> d = {'Michael': 95, 'Bob': 75, 'Tracy': 85}）外，还可以通过key放入：>>> d['Adam'] = 67。由于一个key只能对应一个value，所以，多次对一个key放入value，后面的值会把前面的值冲掉。

如果key不存在，dict就会报错。要避免key不存在的错误，有两种办法，一是通过in判断key是否存在：>>> 'Thomas' in d；二是通过dict提供的**get方法**，如果key不存在，可以返回None，或者自己指定的value：>>> d.get('Thomas') >>> d.get('Thomas', -1)。要删除一个key，用pop(key)方法，对应的value也会从dict中删除。要删除一个key，用**pop(key)方法**，对应的value也会从dict中删除。

**dict的key必须是不可变对象。**在Python中，字符串、整数等都是不可变的，因此，可以放心地作为key。而list是可变的，就不能作为key。

**set**和dict类似，也是一组key的集合，但不存储value。由于key不能重复，所以，在set中，没有重复的key。要创建一个set，需要提供一个list作为输入集合：>>> s = set([1, 2, 3])。注意，传入的参数[1, 2, 3]是一个list，而显示的{1, 2, 3}只是告诉你这个set内部有1，2，3这3个元素，显示的顺序也不表示set是有序的。重复元素在set中自动被过滤，通过add(key)方法可以添加元素到set中，可以重复添加，但不会有效果。通过remove(key)方法可以删除元素。

调用函数

abs函数，max函数max()，数据类型转换函数，比如int()函数：>>> int('123')>>> int(12.34)>>> float('12.34')>>> str(1.23)>>> str(100)>>> bool(1)>>> bool('')。

函数名其实就是指向一个函数对象的引用，完全可以把函数名赋给一个变量，相当于给这个函数起了一个“别名”：>>> a = abs # 变量a指向abs函数 >>> a(-1) # 所以也可以通过a调用abs函数。

如果函数调用出错，一定要学会看错误信息，所以英文很重要！

定义函数

在Python中，定义一个函数要使用def语句，依次写出函数名、括号、括号中的参数和冒号:，然后，在缩进块中编写函数体，函数的返回值用return语句返回。

请注意，函数体内部的语句在执行时，**一旦执行到return时，函数就执行完毕，并将结果返回**。因此，函数内部通过条件判断和循环可以实现非常复杂的逻辑。如果没有return语句，函数执行完毕后也会返回结果，只是结果为None。return None可以简写为return。

Python的函数返回多值其实就是返回一个tuple，但写起来更方便。