img_02

数学与计算机科学学院

**实**

**验**

**报**

**告**

**册**

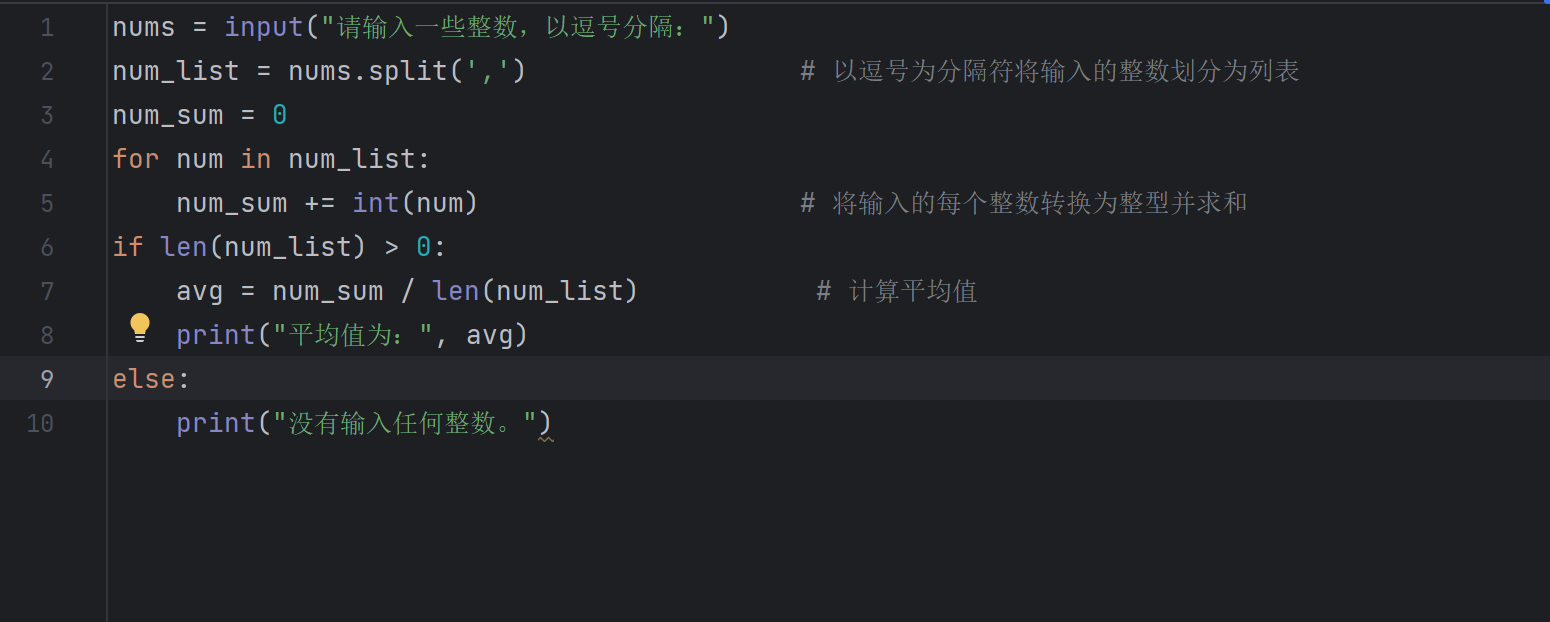
实验课程： python程序设计基础与应用

实验名称： Python内置函数

学号：20220522108 姓名：曾璠宇 班级： 2022级网络工程

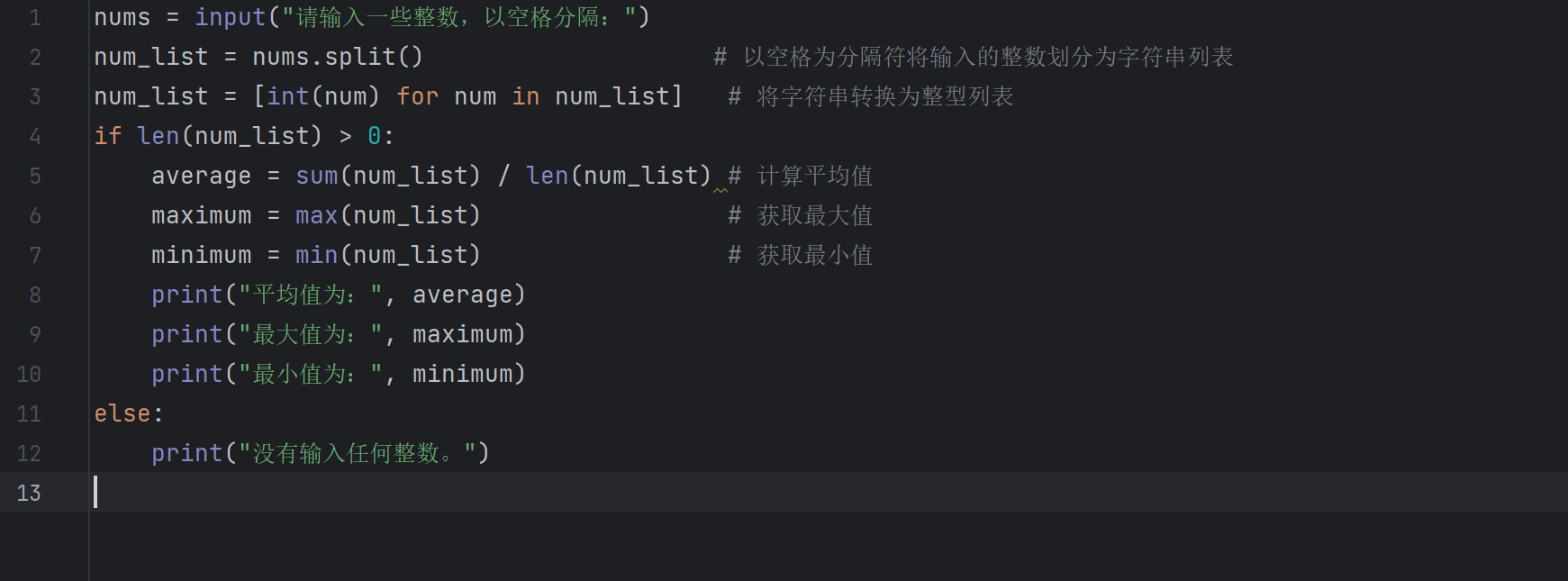
任课教师： 谢发荣 实验日期： 2023 年 9 月 17 日

|  |
| --- |
| **（一）实验内容**  1、输入一个整数，以列表的形式输出每一位，并输出列表的和。  2、输入一个整数，输出对应的二进制、八进制、十六进制。  3、输入一些整数，以逗号分隔，计算并输出这些数的平均值。  4、以列表的形式输入一些整数，计算并输出这些数的平均值、最大值、最小值。  5、以列表的形式输出一些整数，降序输出。  6、编写程序，输入两个各包含2个实数的列表，分别表示点的坐标，计算两点间的欧式距离和曼哈顿距离。  x(x1,x2)  y(y1,y2)  d=sqrt((x1-y1)^2+(x2-y2)^2)  7、输入一个列表，如果列表全为真，输出True，否则输出False  8、输入一个列表，如果列表中有一个为真，输出True,否则输出False  9、输入一个整数，输出整数分离后的和。例如输入1234，输出10  10、输入一个整数，输入分离后的整数，以逗号分隔。例如输入1234，输出：1,2,3,4  11、编写一个函数，测试字符串是否包含列表中某个元素作为子串  12、生成一个1到1e20的随机整数，对这个数进行分离，依次输出每位数，用逗号分隔。  13、用列表生成式，随机生成20个小于100的整数，分别计算平均值、最大值、最小值。  14、用列表生成式，随机生成4x5的列表，即该列表有5个元素，每个元素是一个列表，列表含有4个元素。  分别求最大的列表，和求和最大的列表，并输出。  15、任意输入3个英文单词，以空格分隔分隔，输出字符串长度最大的。  16、随机生成10个整数，分别按字符的字典顺序、按长度、转换为整数比较大小。  17、随机生成两个长度为10列表，将两个列表对应位置的数相加；将两个列表对应位置的数相乘；  将两个列表对应位置的数相乘后得到的列表求和（向量内积）。  18、用range函数生成首项为1，公差为3，小于50的等差数列。用reduce函数求该数列的和。  19、用range函数生成1到10的数字，计算这些数字的乘积。随机生成三个小于10的整数，计算这些数字的乘积。  并比较math.factorial  20、随机生成200个在[0到100]的整数，并把奇数放到一个列表中，偶数放到另一个列表中，计算奇数和偶数的个数。  21、随机生成两个长度为10列表，用zip函数把两个列表压缩。  22、输入一个表达式，输出结果。  23、编写程序，输入一个正整数，然后输出各位数字之和。  24、编写程序，输入一个字符串，输出翻转（首尾交换）后的字符串，要求使用内置函数来实现。例如，12345，输出54321  25、编写程序，输入一个包含若干整数的列表，输出列表中，最大值。  26、编写程序，输入一个包含若干整数的列表，把列表中所有整数转换为字符串，然后输出包含这些字符串的列表。  27、编写程序，输入一个包含若干任意数据的列表，输出该列表中等价于True的元素组成的列表。例如，输入[1,2,0,[]]，输出  [1,2]  28、编写程序，输入一个包含若干整数的列表，输出一个只包含奇数的新列表。  29、编写程序，输入一个包含若干整数的列表，输出一个只包含偶数的新列表。  30、编写程序，输入一个包含若干整数的列表，输出一个一个新列表，列表中奇数在前前偶数在后，并且奇数之间的相对顺序不变，偶数之间的相对顺序也不变。  31、输入一个包含若干自然数的列表，输出这些自然数的平均值，结果保留3小数。  32、输入一个包含若干自然数的列表，输出这些自然数降序排列后的新列表。  33、输入一个包含若干自然数的列表，输出一个新列表，新列表中每个元素为原列表中每个自然数的位数。  34、输入一个包含若干数字的列表，输出其中绝对值最在的数字。  35、输入一个包含若干整数的列表，输出这些整数的乘积。  36、输入两个包含若干整数的等长列表，把这两个列表看作两个向量，输出这两个向量的内积。 |
| **（二）实验目的**  1、熟悉各种python函数的使用方式  2、对python编程语言更加深入的理解 |
| **（三）实验类型**  验证性/设计性/综合性实验（设计性） |
| **（四）实验参考资料**  1、互联网查找  2、Python课本资料学习 |
| **（五）实验内容、步骤（思路）及效果**  1、源代码：    这段代码首先将输入的整数转换为字符串，然后使用列表推导式将每一位作为整数添加到列表 digits 中，最后计算列表中所有元素的和并输出。  **2、源代码**    首先将输入的整数分别转换为二进制、八进制、十六进制表示，并输出它们。然后，它使用max()和min()函数来计算整数的最大值和最小值，并输出它们。  **3、源代码** |



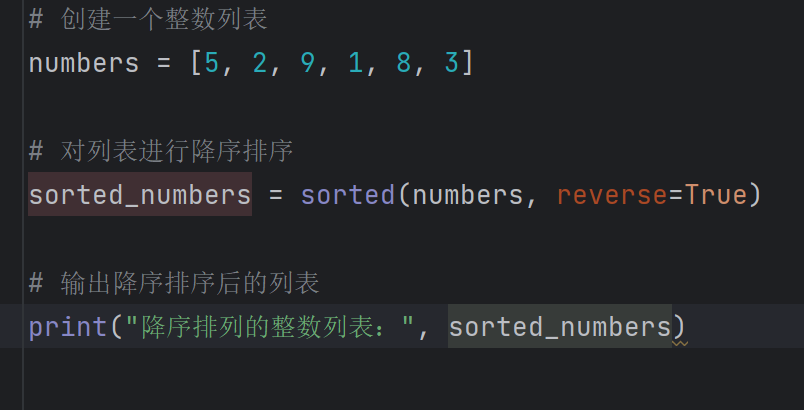
input() 函数获取输入的整数，并使用字符串的 split() 方法以逗号作为分隔符将输入的整数划分为列表，然后遍历该列表并求和，最后计算平均值并输出

4、源代码



首先将输入的一行字符串切割为一个字符串列表，然后将列表中的每个元素转换为整数，并存储在 num\_list 中。接下来，我们可以使用内置的 sum()、max() 和 min() 函数来计算列表的和、最大值和最小值，并将结果进行输出。

5、源代码



6、源代码



1. 首先导入了math模块，以便使用其中的数学函数。
2. 通过input函数分别获取第一个点和第二个点的坐标输入，并用逗号将输入的字符串分割成列表。
3. 使用列表推导式将字符串列表中的每个元素转换成浮点数类型，得到两个包含实数的列表x和y。
4. 根据欧式距离公式 d = sqrt((x1-y1)^2+(x2-y2)^2)，计算两点之间的欧式距离并将结果存储在变量euclidean\_distance中。这里使用了math.sqrt函数来计算平方根。
5. 根据曼哈顿距离公式 d = |x1-y1| + |x2-y2|，计算两点之间的曼哈顿距离并将结果存储在变量manhattan\_distance中。
6. 使用print函数输出欧式距离和曼哈顿距离的值。

7、源代码



首先定义了一个函数check\_all\_true，它接受一个列表作为参数，并使用all函数判断列表中的元素是否全部为真。

然后，通过input函数获取输入的列表，并将输入的字符串切割成多个元素。接着，使用列表推导式将输入的字符串列表中的每个元素转换为布尔值。

最后，调用check\_all\_true函数判断列表中的元素是否全部为真，并将结果输出。

8、源代码

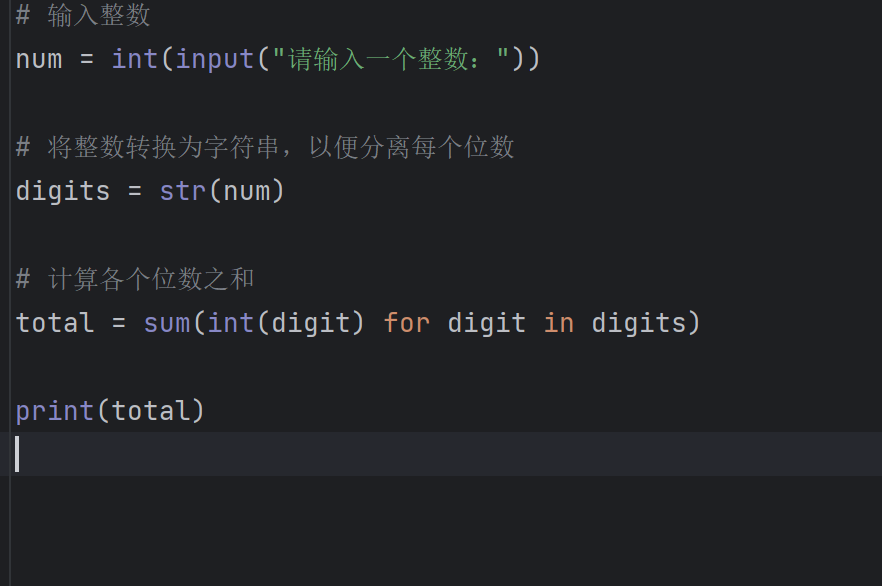


首先定义了一个函数check\_any\_true，它接受一个列表作为参数，并使用any函数判断列表中是否存在至少一个真值。

然后，通过input函数获取输入的列表，并将输入的字符串切割成多个元素。接着，使用列表推导式将输入的字符串列表中的每个元素转换为布尔值。

最后，调用check\_any\_true函数判断列表中是否存在至少一个真值，并将结果输出。

9、源代码



首先通过input函数获取输入的整数，并将其转换为整型。

接着，将整数转换为字符串，以便可以逐个分离出每个位数。

然后，使用列表推导式将字符串中的每个字符转换为整型，并使用sum函数计算所有位数之和。

最后，将结果输出。

10、源代码



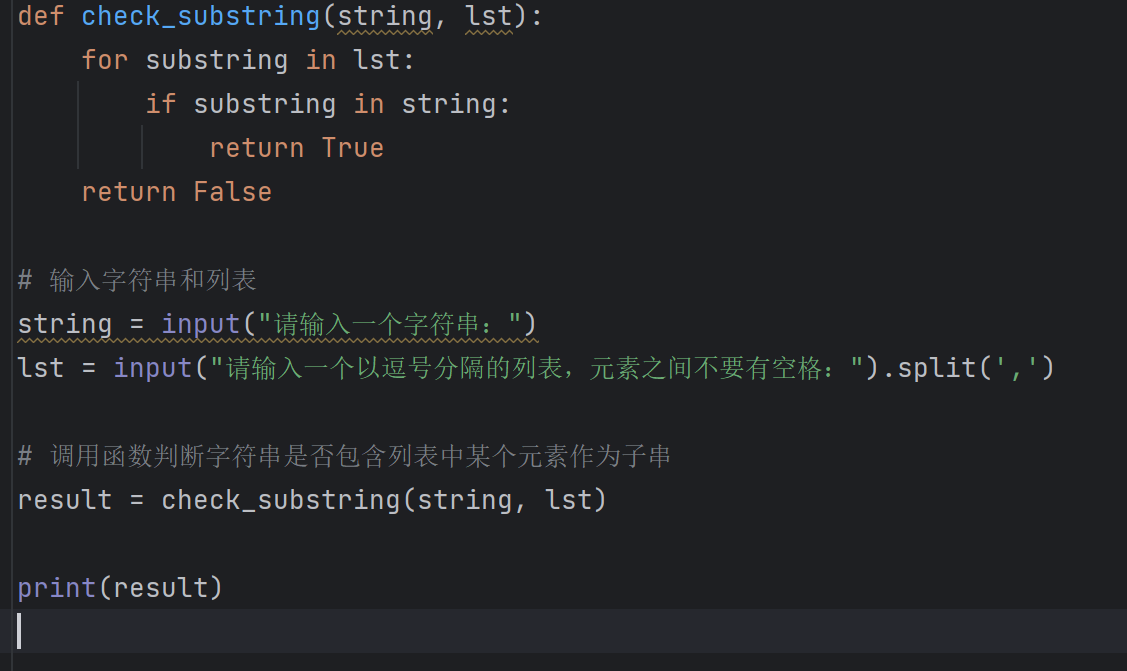
首先通过input函数获取输入的整数，并将其转换为整型。

接着，将整数转换为字符串，以便可以逐个分离出每个位数。

然后，使用join方法将各个位数连接起来，以逗号分隔。

最后，将结果输出。

11、源代码



定义了一个函数check\_substring，它接受一个字符串和一个列表作为参数。函数遍历列表中的每个元素，并使用in运算符判断字符串是否包含该元素作为子串。如果找到任何一个匹配的子串，则返回True，否则返回False。

然后，使用input函数获取输入的字符串和列表，并将输入的列表字符串切割成多个元素。

最后，调用check\_substring函数判断字符串是否包含列表中某个元素作为子串，并将结果输出。

12、源代码



首先导入random模块，以便使用其中的函数生成随机数。

 import random

 使用random.randint(a, b)函数生成1到1e20之间的随机整数。该函数的参数a和b分别表示生成随机整数的范围。本例中，a为1，b为10的20次方。

 num = random.randint(1, 10\*\*20)

 将生成的随机整数转换为字符串，以便分离每个位数。这里使用str()函数将整数转换为字符串类型。

 digits = str(num)

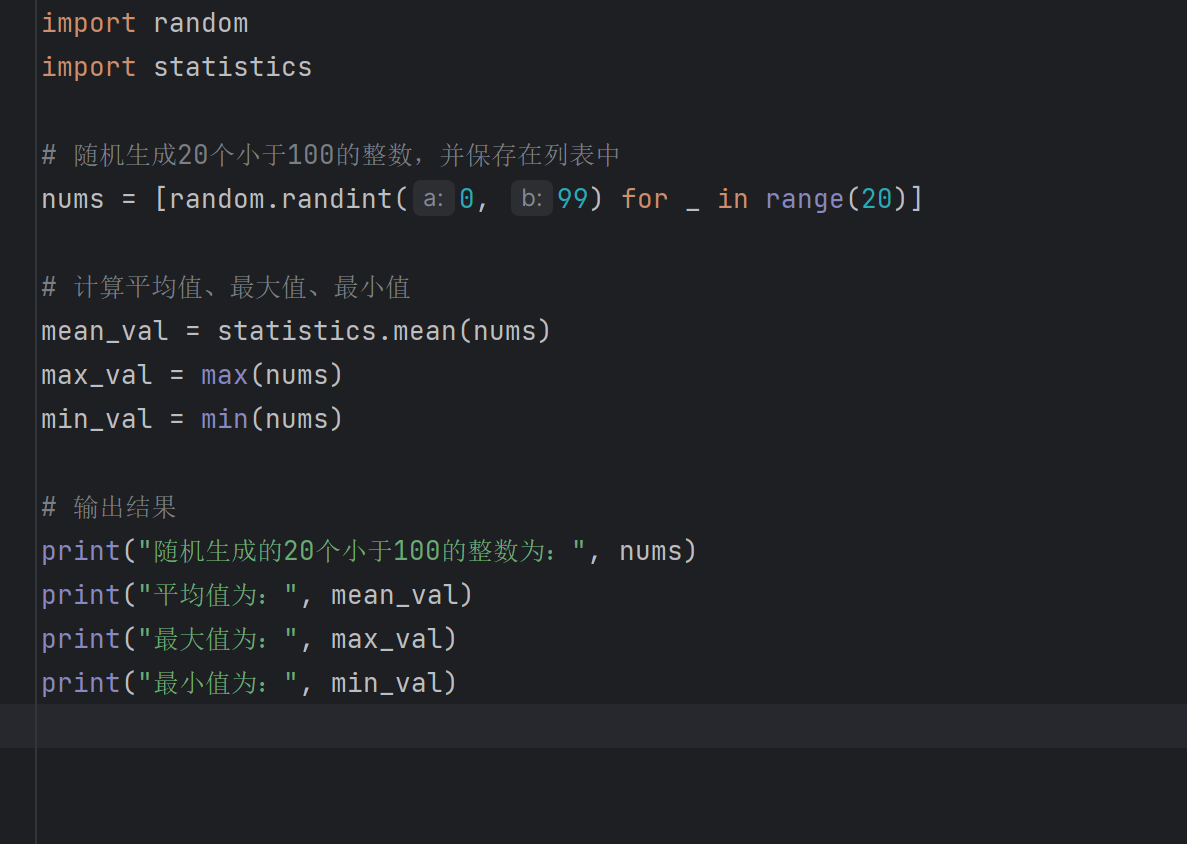
 使用join()方法以逗号分隔输出各个位数。在这里，首先需要将每个位数用,连接起来，然后才能输出。因此，使用join()方法将每个位数作为一个列表元素，以逗号进行连接。最终结果得到的是一个字符串。

 result = ",".join(digits)

 最后，将结果输出。

print(result)

13、源代码

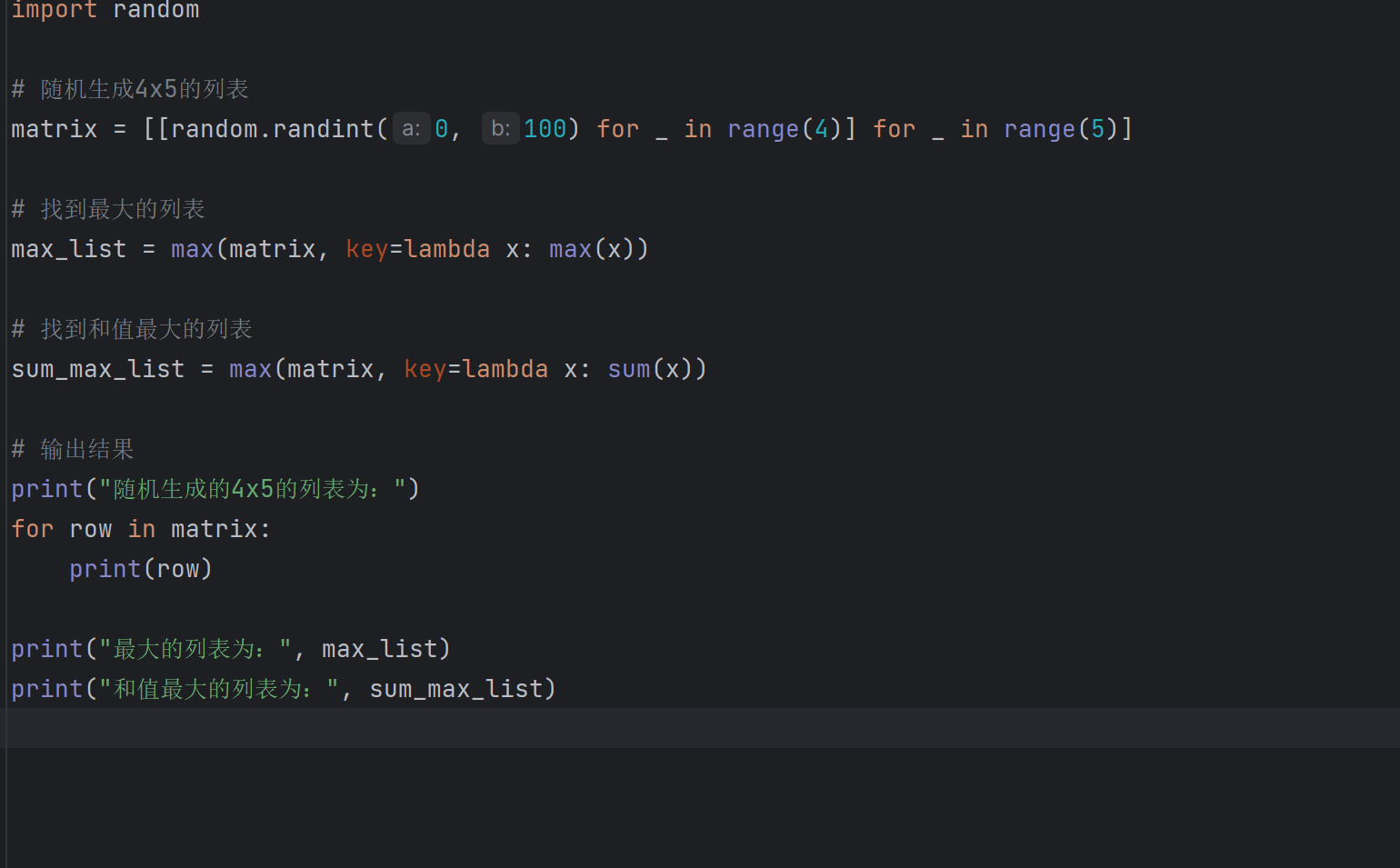


使用了列表生成式随机生成了一个包含20个小于100的整数的列表，这里用到的是Python的random模块中的randint()函数，用来生成指定范围内的随机整数。

然后，使用statistics模块中的mean()函数计算列表中所有元素的平均值；同时使用内置的max()函数和min()函数分别求出列表中的最大值和最小值。这里要注意，max()和min()函数适用于任何可迭代对象，不仅仅限于列表。

最后，通过打印输出结果，将生成的随机数列表、平均值、最大值、最小值都展示出来。

14、源代码

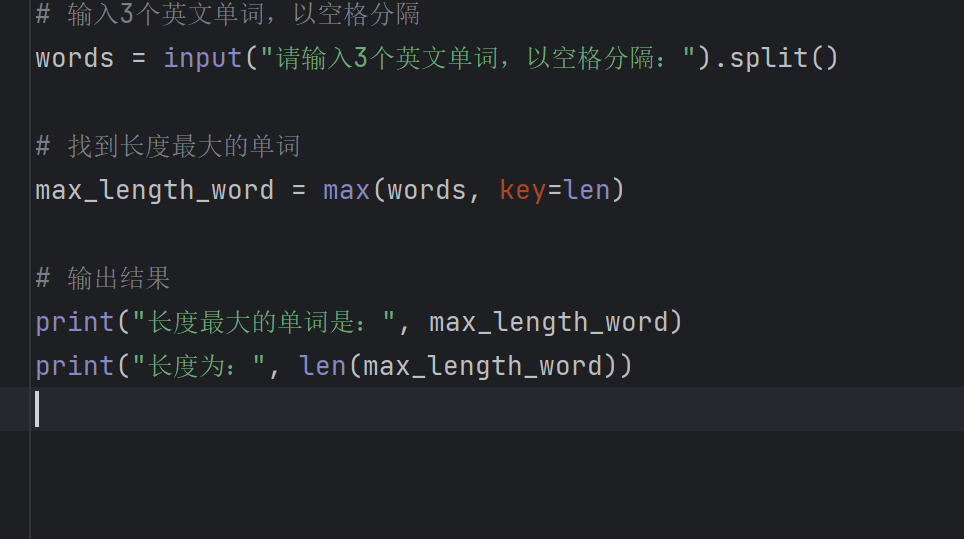


首先使用了列表生成式来生成一个4x5的列表matrix，其中每个元素都是一个包含4个随机整数的列表。

然后，使用内置函数max()结合lambda表达式来找到最大的列表和和值最大的列表。max()函数接受两个参数，第一个参数是要比较的可迭代对象，第二个参数key是一个函数，用于指定如何比较元素。在这里，我们使用lambda表达式作为key，lambda x: max(x)用于找到列表中的最大值，lambda x: sum(x)用于计算列表中元素的和。

最后，通过打印输出结果，将生成的4x5的列表、最大的列表和和值最大的列表都展示出来。

15、源代码

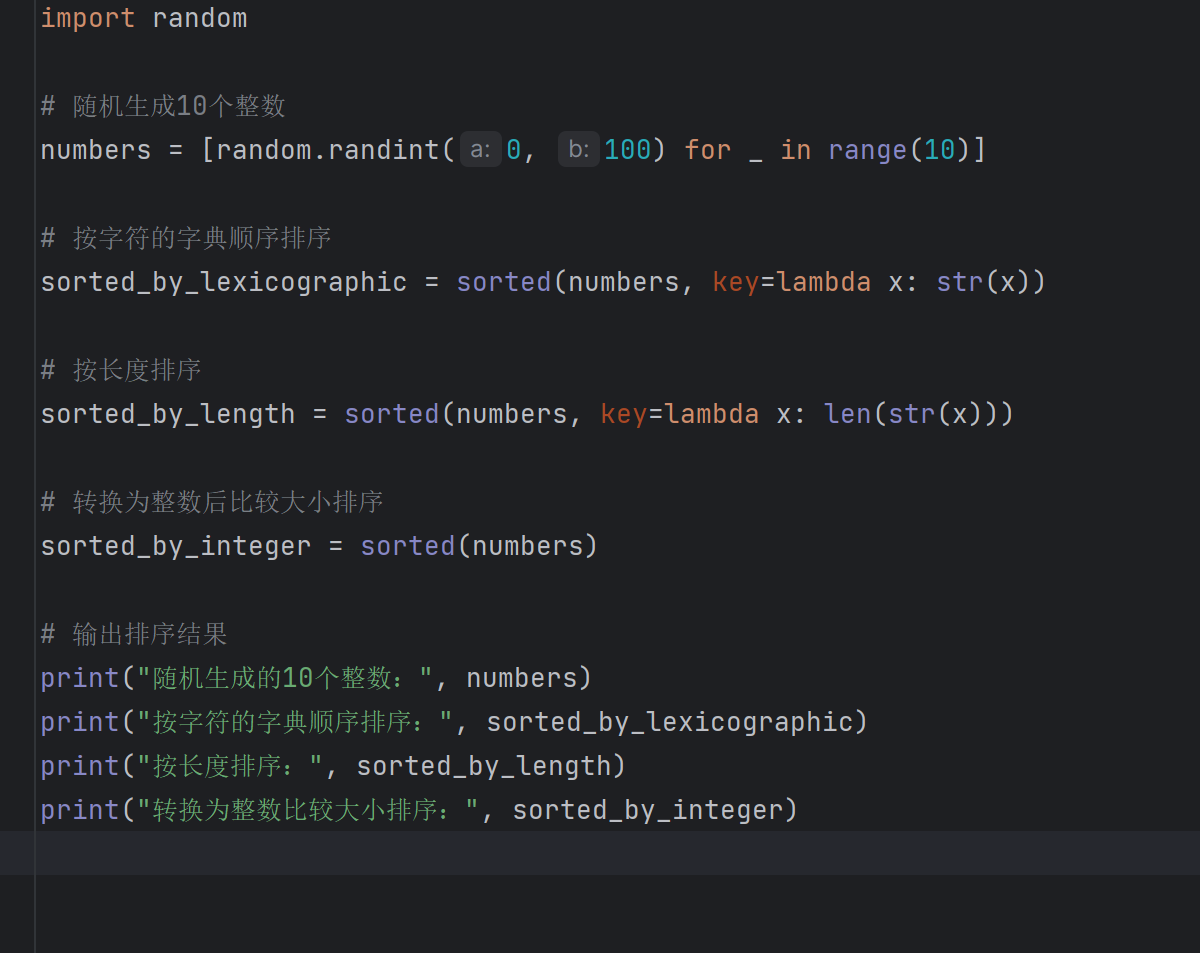


使用input()函数来获取用户输入的3个英文单词，这里使用split()方法将输入的字符串按照空格进行分割，并保存在列表words中。

然后，使用内置函数max()结合lambda表达式来找到列表中长度最大的单词。max()函数的第一个参数是要比较的可迭代对象，第二个参数key是一个函数，用于指定如何比较元素。在这里，我们使用lambda表达式key=len，表示以单词的长度作为比较依据。

最后，通过打印输出结果，将长度最大的单词和其长度展示出来。

16、源代码

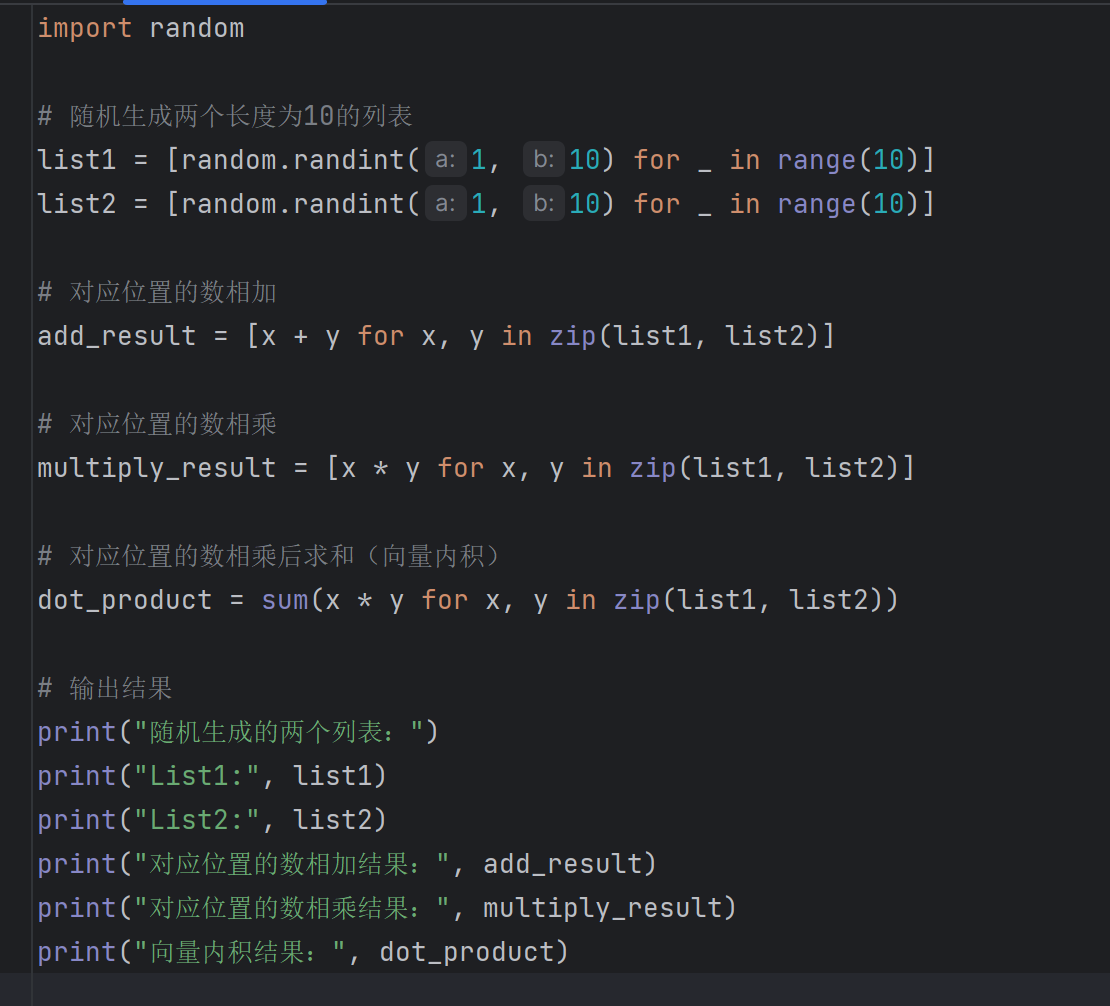


首先使用列表生成式和random模块生成了一个包含10个随机整数的列表numbers。

然后，使用内置函数sorted()结合lambda表达式来进行排序。sorted()函数的第一个参数是要排序的可迭代对象，第二个参数key是一个函数，用于指定如何比较元素。在这里，我们使用lambda表达式作为key，分别使用str(x)表示按字符的字典顺序排序，len(str(x))表示按长度排序，而默认情况下直接使用整数本身的值进行比较。使用sorted()函数返回的结果将是一个排序后的新列表。

最后，通过打印输出结果，将生成的10个随机整数和三种排序方式的结果都展示出来。

17、源代码



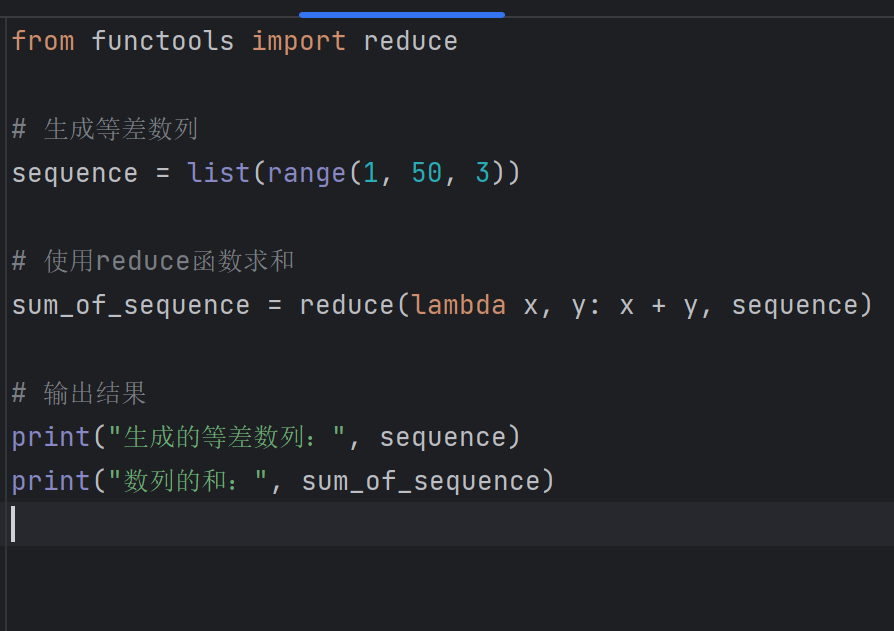
使用列表推导式和random模块生成了两个长度为10的随机列表list1和list2，其中random.randint(1, 10)用于生成1到10之间的随机整数。

然后，使用zip()函数将两个列表对应位置的数配对起来，然后通过列表推导式分别进行对应位置的数相加和相乘操作。

最后，使用内置函数sum()对相乘后的结果进行求和，得到向量的内积。

通过打印输出结果，将生成的两个随机列表和三种操作的结果都展示出来。

18、源代码



使用range函数生成了一个等差数列，其中参数1表示首项为1，参数50表示数列中最大的元素（不包含），参数3表示公差为3。通过将range对象转换为列表，得到了所需的等差数列。

然后，使用functools模块中的reduce函数对列表进行累加求和。reduce函数接受一个二元操作函数和一个可迭代对象作为参数，将前两个元素传递给二元操作函数进行计算，然后将结果与下一个元素进行运算，依次类推，最终得到一个值。

最后，通过打印输出结果，展示生成的等差数列和数列的和。

19、源代码



首先使用math库中的prod函数结合range函数生成了1到10的数字，并计算它们的乘积。

接着，使用random模块中的randint函数生成了三个小于10的随机整数，并将它们存储在列表random\_numbers中。然后，使用math库中的prod函数计算这三个随机整数的乘积。

最后，通过打印输出结果，展示了1到10的数字乘积、随机生成的三个小于10的数字以及它们的乘积。

20、源代码



首先初始化了两个空列表odd\_numbers和even\_numbers，用于存放奇数和偶数。

然后，使用for循环生成200个随机整数，并判断每个随机数的奇偶性。如果是偶数，则将其添加到even\_numbers列表中；如果是奇数，则将其添加到odd\_numbers列表中。

最后，通过计算odd\_numbers和even\_numbers列表的长度，分别得到奇数和偶数的个数。

21、源代码

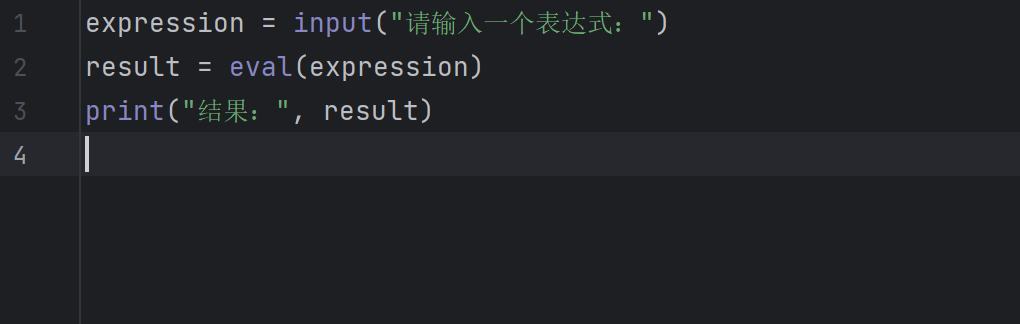


首先使用列表推导式生成了两个长度为10的随机列表list1和list2，其中每个元素都是在范围1到100内的随机整数。

然后，通过使用zip函数将两个列表进行压缩，得到一个包含元组的新列表zipped\_list。每个元组由list1和list2中对应位置的元素组成。

最后，通过打印输出结果，展示了原始的两个列表list1和list2，以及压缩后的列表zipped\_list

22、源代码

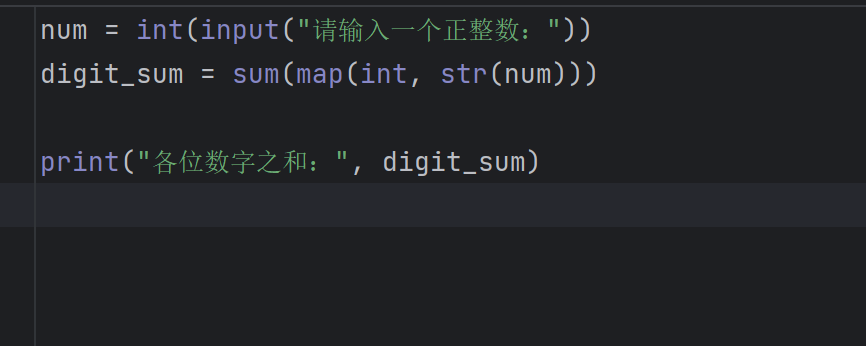


首先使用input()函数接收用户输入的表达式，并将其保存在变量expression中。

然后，使用eval()函数对expression进行求值，并将结果保存在变量result中。

最后，通过打印输出结果，显示表达式的计算结果。

23、源代码

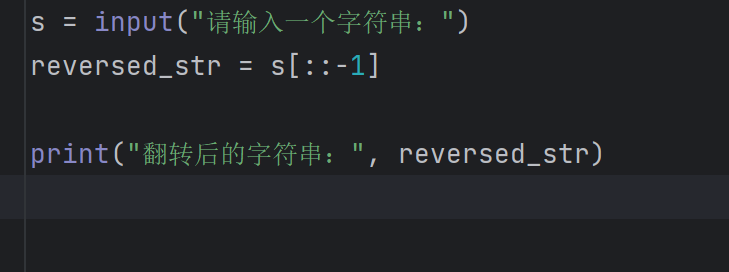


首先使用input()函数获取用户输入的正整数，并将其转换为整型赋值给变量num。

然后，使用map()函数将num转换为字符串，并通过int函数将每个字符转换为相应的整数。然后，使用sum()函数对得到的整数列表求和，得到各位数字之和，并将结果保存在变量digit\_sum中。

最后，通过打印输出结果，展示各位数字之和

24、源代码

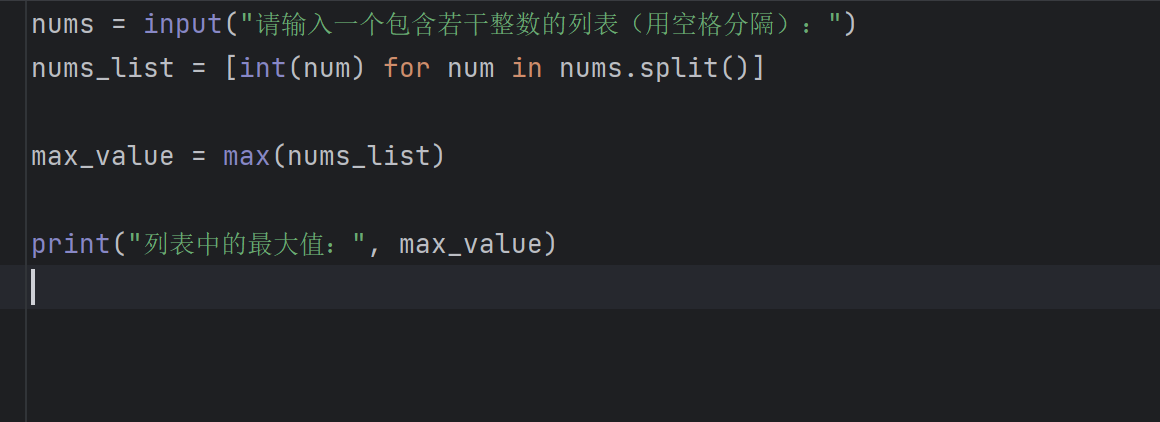


首先使用input()函数获取用户输入的字符串，并将其保存在变量s中。

然后，通过使用字符串的切片操作[::-1]，可以将字符串逆序翻转。该切片操作会生成一个新的字符串，其中步长为-1表示从末尾向前遍历。

最后，通过打印输出结果，展示翻转后的字符串。

25、源代码



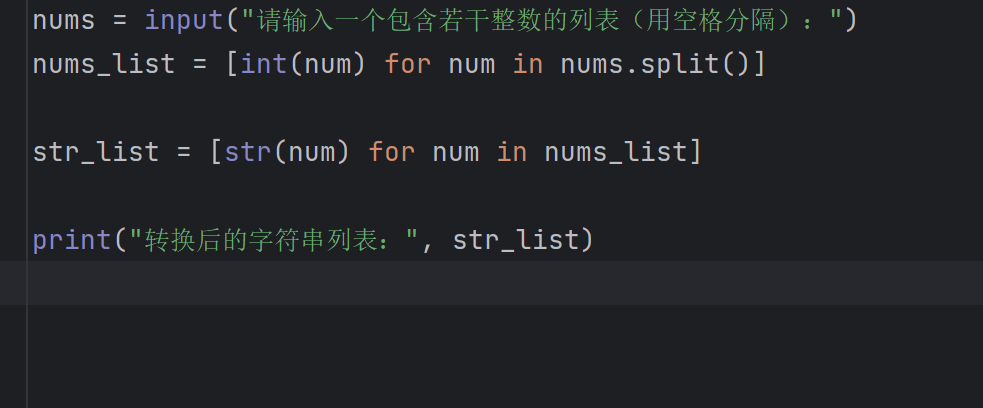
首先使用input()函数获取用户输入的一串数字，并将其保存在变量nums中。

然后，通过列表推导式将输入的数字字符串分割成若干整数，并保存在列表nums\_list中。其中，num为每个数字的字符串形式，通过int()函数将其转换为整数。

接下来，使用max()函数找到列表nums\_list中的最大值，并将结果保存在变量max\_value中。

最后，通过打印输出结果，展示列表中的最大值。

26、源代码



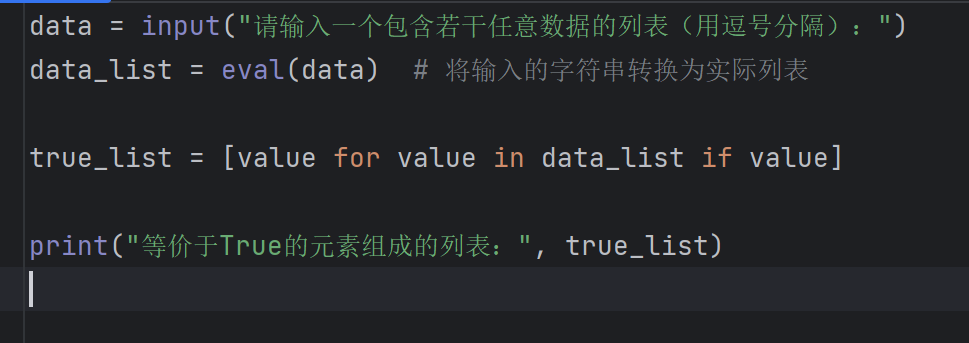
首先使用input()函数获取用户输入的一串数字，并将其保存在变量nums中。

然后，通过列表推导式将输入的数字字符串分割成若干整数，并保存在列表nums\_list中。其中，num为每个数字的字符串形式，通过int()函数将其转换为整数。

接下来，使用另一个列表推导式将列表nums\_list中的每个整数转换为字符串形式，并生成一个新的字符串列表str\_list。

最后，通过打印输出结果，展示转换后的字符串列表。

27、源代码



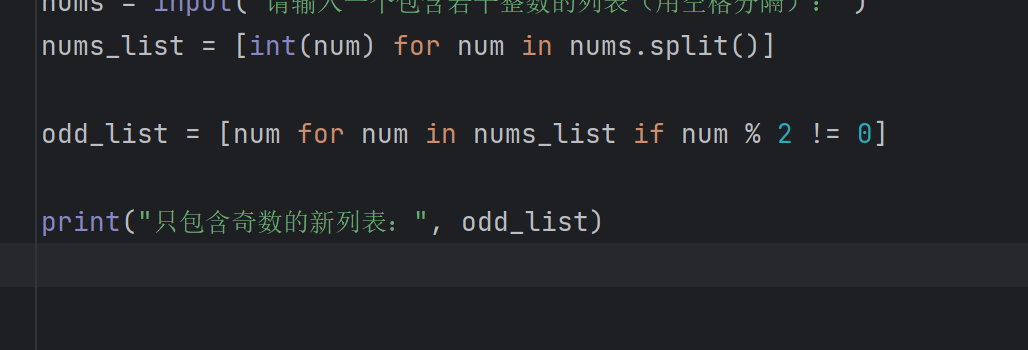
首先使用input()函数获取用户输入的一串数据，并将其保存在变量data中。

然后，通过使用eval()函数将输入的字符串转换为实际的列表对象，保存在变量data\_list中。注意，使用eval()函数需谨慎，确保输入的字符串是安全的且符合语法规则。

接下来，使用列表推导式遍历data\_list中的每个元素，筛选出等价于True的元素，并生成一个新的列表true\_list。

最后，通过打印输出结果，展示等价于True的元素所组成的列表。

28、源代码



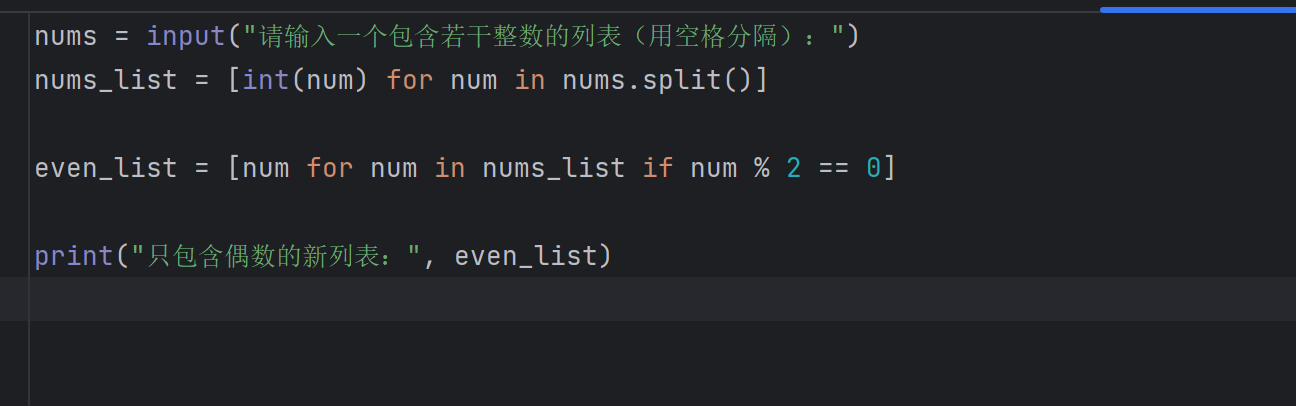
首先使用input()函数获取用户输入的一串数据，并将其保存在变量data中。

然后，通过使用eval()函数将输入的字符串转换为实际的列表对象，保存在变量data\_list中。注意，使用eval()函数需谨慎，确保输入的字符串是安全的且符合语法规则。

接下来，使用列表推导式遍历data\_list中的每个元素，筛选出等价于True的元素，并生成一个新的列表true\_list。

最后，通过打印输出结果，展示等价于True的元素所组成的列表。

29、源代码



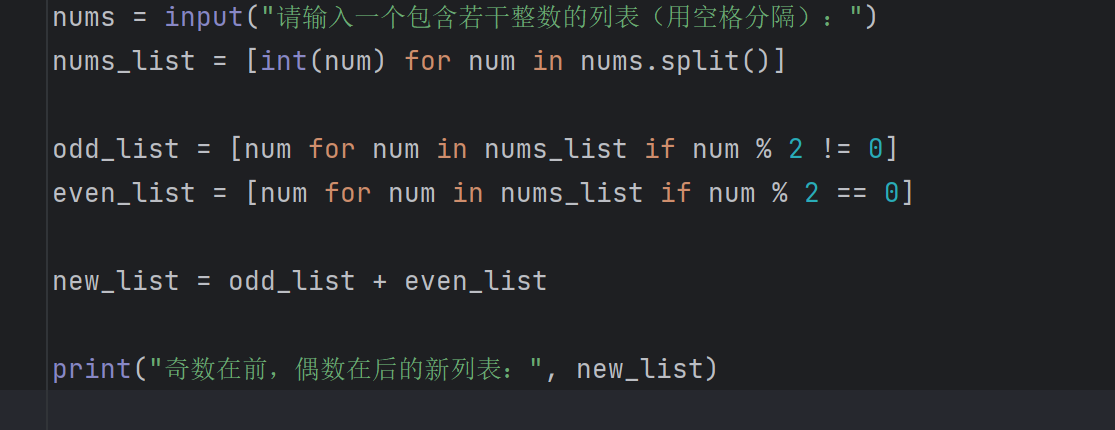
首先使用input()函数获取用户输入的一串数字，并将其保存在变量nums中。

然后，通过列表推导式将输入的数字字符串分割成若干整数，并保存在列表nums\_list中。其中，num为每个数字的字符串形式，通过int()函数将其转换为整数。

接下来，使用列表推导式遍历nums\_list中的每个整数，筛选出所有的偶数，并生成一个新的列表even\_list。在判断偶数的逻辑中，使用了对2取余的操作num % 2，如果结果等于0，则说明该数为偶数。

最后，通过打印输出结果，展示只包含偶数的新列表。

30、源代码



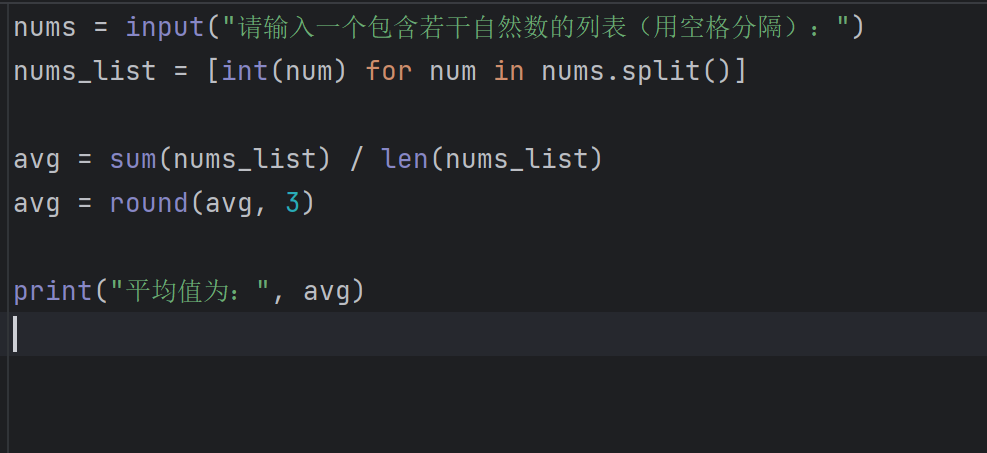
首先使用input()函数获取用户输入的一串数字，并将其保存在变量nums中。

然后，通过列表推导式将输入的数字字符串分割成若干整数，并保存在列表nums\_list中。其中，num为每个数字的字符串形式，通过int()函数将其转换为整数。

接下来，使用列表推导式遍历nums\_list中的每个整数，筛选出所有的奇数并保存在列表odd\_list中，同时筛选出所有的偶数并保存在列表even\_list中。

最后，使用+运算符将odd\_list和even\_list拼接起来，得到一个新的列表new\_list，其中奇数在前，偶数在后。

31、源代码



首先使用input()函数获取用户输入的一串数字，并将其保存在变量nums中。

然后，通过列表推导式将输入的数字字符串分割成若干整数，并保存在列表nums\_list中。其中，num为每个数字的字符串形式，通过int()函数将其转换为整数。

接下来，使用sum()函数计算nums\_list中所有自然数的总和，并除以列表长度，得到平均值。

使用round()函数将平均值保留3位小数，传入两个参数，第一个参数为要保留小数位的数，第二个参数为保留的位数。

最后，通过打印输出结果，展示计算得到的平均值。

32、源代码



首先使用input()函数获取用户输入的一串数字，并将其保存在变量nums中。

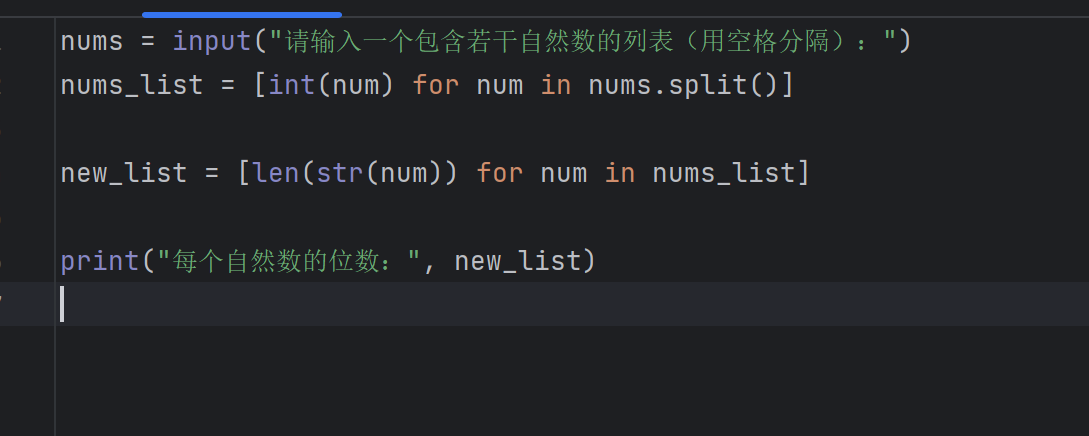
然后，通过列表推导式将输入的数字字符串分割成若干整数，并保存在列表nums\_list中。其中，num为每个数字的字符串形式，通过int()函数将其转换为整数。

接下来，使用sorted()函数对nums\_list进行降序排序，传入两个参数，第一个参数为待排序的列表，第二个参数reverse=True表示降序排序。

将排序结果保存到新的列表new\_list中。

最后，通过打印输出结果，展示降序排列后的新列表。

33、源代码



首先使用input()函数获取用户输入的一串数字，并将其保存在变量nums中。

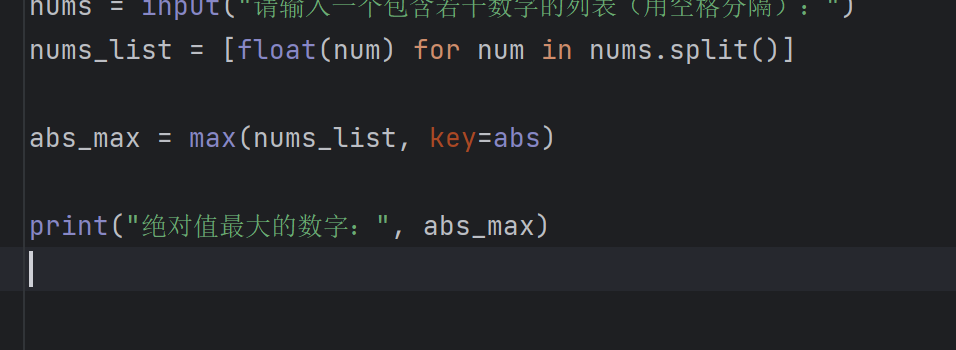
然后，通过列表推导式将输入的数字字符串分割成若干整数，并保存在列表nums\_list中。其中，num为每个数字的字符串形式，通过int()函数将其转换为整数。

接下来，使用列表推导式遍历nums\_list中的每个元素，并使用len()函数获取每个自然数的位数。len(str(num))将整数转换为字符串后再计算长度，即得到位数。

将位数结果保存到新的列表new\_list中。

最后，通过打印输出结果，展示每个自然数的位数

34、源代码



首先使用input()函数获取用户输入的一串数字，并将其保存在变量nums中。

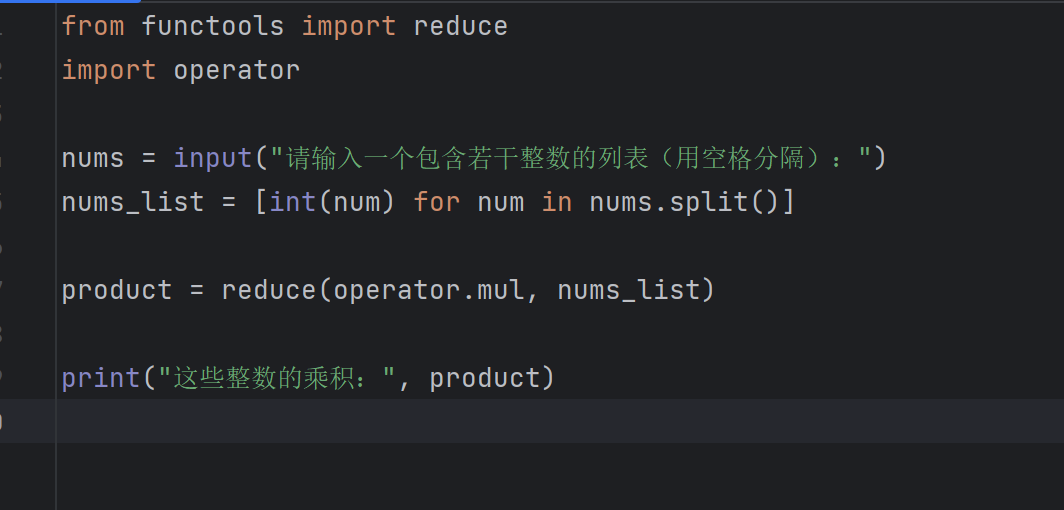
然后，通过列表推导式将输入的数字字符串分割成若干浮点数，并保存在列表nums\_list中。其中，num为每个数字的字符串形式，通过float()函数将其转换为浮点数。

接下来，使用max()函数找到nums\_list中绝对值最大的数字。通过key=abs参数，指定以绝对值为比较依据进行比较。

将绝对值最大的数字保存在变量abs\_max中。

最后，通过打印输出结果，展示绝对值最大的数字。

35、源代码



首先使用input()函数获取用户输入的一串数字，并将其保存在变量nums中。

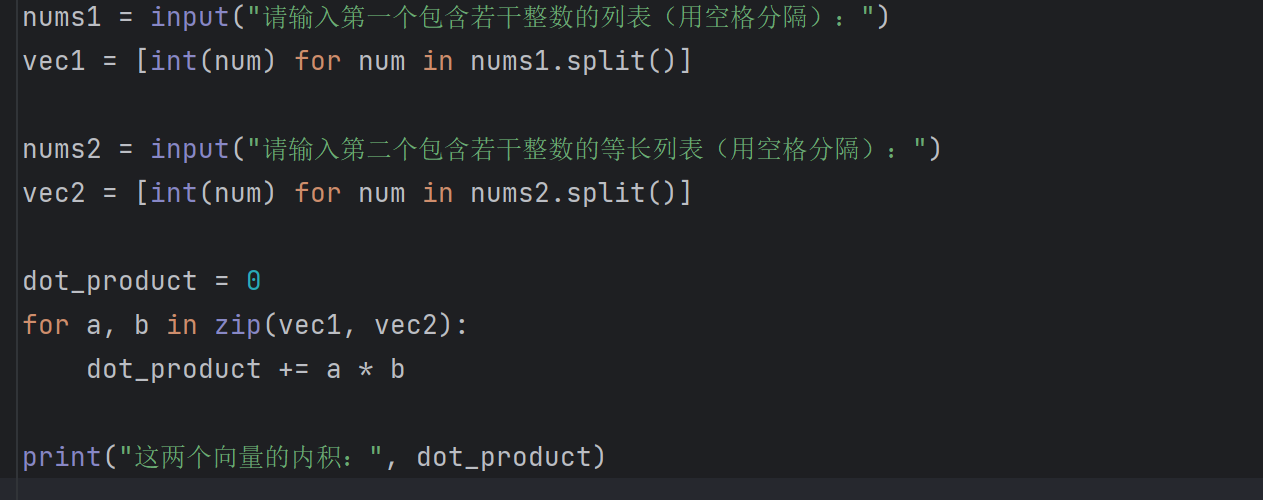
然后，通过列表推导式将输入的数字字符串分割成若干整数，并保存在列表nums\_list中。其中，num为每个数字的字符串形式，通过int()函数将其转换为整数。

接下来，使用reduce()函数对nums\_list中的整数依次进行累积乘法运算。reduce()函数需要两个参数，第一个参数为用于进行乘法运算的函数，这里使用operator.mul表示乘法运算；第二个参数为待操作的列表，即nums\_list。

将乘积结果保存在变量product中。

最后，通过打印输出结果，展示整数的乘积。

36、源代码



首先使用input()函数获取用户输入的两个包含若干整数的等长列表，并将它们分别保存在变量nums1和nums2中。

然后，通过列表推导式将每个数字字符串转换为整数，并保存在两个列表vec1和vec2中。

接下来，通过zip()函数将vec1和vec2中的对应元素配对，并遍历每个配对产生的元组(a, b)。在循环中，将每个元组中的两个数相乘并累加到变量dot\_product上。

最后，通过打印输出结果，展示两个向量的内积。

|  |
| --- |
| **（六）实验小结与心得**  问题1：如何将一个字符串列表按照字母顺序进行排序？  实验小结： 通过使用Python的sorted()函数和join()方法，我们可以轻松地将一个字符串列表按照字母顺序进行排序并将结果以字符串形式输出。  心得： 这个问题教会了我们如何使用Python中的内置函数来解决字符串列表排序的问题。通过灵活运用列表的排序方法，我们可以快速有效地对数据进行处理。  问题2：如何计算一个字符串中某个字符出现的次数？  实验小结： 通过使用Python的字符串方法count()，我们可以方便地计算一个字符串中特定字符出现的次数。  心得： 解决这个问题让我们意识到在Python中有很多内置方法和函数可以帮助我们处理字符串。count()方法是一种简单而强大的工具，可以帮助我们快速获取字符出现的次数。  问题3：如何计算两个向量的内积？  实验小结： 通过使用zip()函数和一个简单的循环，我们可以计算两个向量的内积。  心得： 解决这个问题让我们了解到如何使用zip()函数将两个向量中对应的元素配对，并通过循环和累加运算来计算内积。这个问题展示了如何利用Python的一些基本操作和函数解决数值计算问题。  总结： 通过解决这些编程问题，我们巩固了对Python中常用字符串方法、列表操作和数值计算的理解。同时，通过实际编码和测试，我们提高了问题解决能力和代码调试技巧。这些问题的实验让我们更加熟悉了Python的基本功能和语法，为进一步的编程学习打下了坚实的基础。 |