

实验3 序列

ANDYWWW

1. 给出一个字符串，不区分大小写，字符串中可能包含'A'-'Z','a'-'z',' '(空格)等字符。输出字母a（包括大小写）出现的次数。

```
s = input('Please input the string:\n')
s_lower = s.lower()
n = s_lower.count('a')
print(n)
```

注意区分'a'与a:
'a'是一个字符串,
但是a是一个变量名!

由于包含大小写，最简单的统计办法为将所有大小写统一（大写函数upper），当然也可以分别计算a和A的个数。

字符串的lower函数与upper函数

```
Help on built-in function lower:  
lower(...) method of builtins.str instance  
  S.lower() -> str
```

```
Return a copy of the string S converted to lowercase.
```

根据帮助，其作用为返回一个S全部小写的新字符串，并不改变原字符串，使用时必须将这个返回值赋值才有作用；upper函数同理
当然，在交互式界面可以直接输出返回值，但是写在文档中必须将返回值print出来才可以体现。

字符串的count函数

Help on built-in function count:

count(...) method of builtins.str instance

S.count(sub[, start[, end]]) -> int

Return the number of non-overlapping occurrences of substring sub in string S[start:end]. Optional arguments start and end are interpreted as in slice notation.

在帮助里，由中括号[]包起来的部分可以不输入。其作用为寻找在S[start:end]内sub字符串的个数，返回值为字符串的个数，因此必须print才可以体现这个函数的作用。

2. 编写程序，将给定的字符串序列，按照字符ASCII码顺序从小到大排序后输出。

```
s = input('Please input the string:\n')
lst = sorted(s)
res = ''.join(lst)
print(res)
```

sorted函数返回值是一个排好序的列表，如果想要输出一个字符串利用join函数即可

针对可迭代对象的sorted函数

```
sorted(iterable, /, *, key=None, reverse=False)
```

Return a new list containing all items from the iterable in ascending order.

A custom key function can be supplied to customize the sort order, and the reverse flag can be set to request the result in descending order.

可迭代对象类都有__next__方法，常见的可迭代对象有字符串、列表、元组、range等。

sorted函数的作用为将可迭代对象进行升序，并返回一个列表。如果需要降序只需要变为reverse = True即可；如果不直接根据元素本身排序，而是对某个变化后的对象进行排序，则利用key即可，例如如果排序时不关乎大小写，只需要key=lambda x:x.lower()即可。

匿名函数lambda

sorted参数里的key多为匿名函数，简单而言lambda函数为一个简单的函数，其中冒号之前的内容为形参，冒号后为返回值；lambda函数只能接简单的判断结构，如果需要复杂的函数利用def编写后再套入lambda函数即可

lambda xxx:yyy

lambda 参数:返回值

例: ① `f = lambda x:x+1`

则f(1)返回值为2

② `f = lambda x:x.lower()`

则f('aBCde')返回值为abcde

③ `f = lambda x,y:[y,x]`

则f(1,2)返回值为[2,1]

④ `f = lambda x: "yes" if x>1 else "no"` 则f(2)返回值为"yes"

字符串的join函数

join(...) method of builtins.str instance
S.join(iterable) -> str

Return a string which is the concatenation of the strings in the iterable. The separator between elements is S.

参数为可迭代对象，返回值为将可迭代对象每一项用S衔接的一个字符串。

例如','.join([1,2,3])的返回值为'1,2,3'， ','.join('1,2,3')为'1,,,2,,,3'。

字符串的split函数

Help on built-in function split:

split(...) method of builtins.str instance

S.split(sep=None, maxsplit=-1) -> list of strings

Return a list of the words in S, using sep as the delimiter string. If maxsplit is given, at most maxsplit splits are done. If sep is not specified or is None, any whitespace string is a separator and empty strings are removed from the result.

视sep为分隔符，去掉maxsplit（默认为所有）个分隔符，返回值为去掉分隔符后得到的列表。例如'1 2 3 4'.split(' ') = ['1','2','3','4']

3. 小张举办生日宴会，想知道小赵是第几个到的。

```
lst = input("Input the names:").split()
```

```
lst.insert(0, "friends")
```

```
lst.insert(len(lst), "#") 或 lst.append("#")
```

```
print(lst.index("xiaozhao"))
```

注意字符串内建函数split()
用法、列表函数insert用法。

列表的index函数

Help on built-in function index:

index(...) method of builtins.list instance

L.index(value, [start, [stop]]) -> integer -- return first index of value.

Raises ValueError if the value is not present.

从L[start:stop]中寻找value，返回第一个元素为value的索引；如果不指定start和stop默认为列表开头与结尾

4.列表中存放了20个整数，分别表示20个评委的评分，请编写程序，去掉其中的最高分和最低分，求剩下18个分数的平均值。

方法一

```
lst = list(eval(input()))
lst.sort()
lst.pop()
lst.pop(0)
mean = sum(lst) / len(lst)
print(mean)
```

方法二

```
lst = list(eval(input()))
mean = (sum(lst) - max(lst) - min(lst)) / (len(lst) - 2)
print(mean)
```

列表的pop函数

Help on built-in function pop:

pop(...) method of builtins.list instance

L.pop([index]) -> item -- remove and return item at index (default last).

Raises IndexError if list is empty or index is out of range.

pop函数用于移除L[index]的元素，默认为最后一个元素，并返回该移除的元素。

eval函数——让你的输入更加灵活

Help on built-in function eval in module builtins:

```
eval(source, globals=None, locals=None, /)
```

Evaluate the given source in the context of globals and locals.

The source may be a string representing a Python expression or a code object as returned by compile(). The globals must be a dictionary and locals can be any mapping, defaulting to the current globals and locals. If only globals is given, locals defaults to it.

一个字符串可能代表一个Python的表达式，利用eval函数可将其还原成表达式，例如`eval('[1,2,3]')` = `[1,2,3]`，`eval('1 != 1')`返回值为`False`。注意一般不能是一个赋值语句！ 其反函数为`repr`函数

eval函数——让你的输入更加灵活

```
lst = input('Please input the string:\n')
```

键入[1, 2, 3, 5, 7, 6]

```
lst = '[1, 2, 3, 5, 7, 6]'
```

 类型为str

```
lst = eval(lst)
```

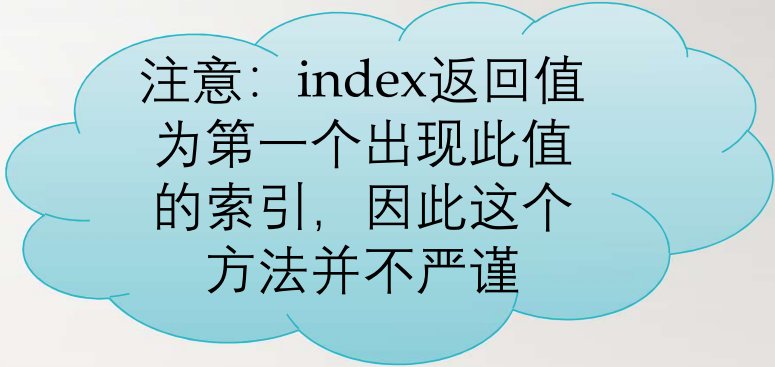
```
lst = [1, 2, 3, 5, 7, 6]
```

 类型为list

5.列表中存放了某次考试学生的考试成绩，请编写程序分别求出不及格学生（<60）和甲等（>85）学生的平均成绩。

方法I

```
scores = list(eval(input()))  
scores.sort()  
a = scores.index(85)  
d = scores.index(60)  
a_ave = scores[a:] / len(scores[a:])  
d_ave = scores[0:d] / len(scores[0:d])  
print(a_ave)  
print(d_ave)
```

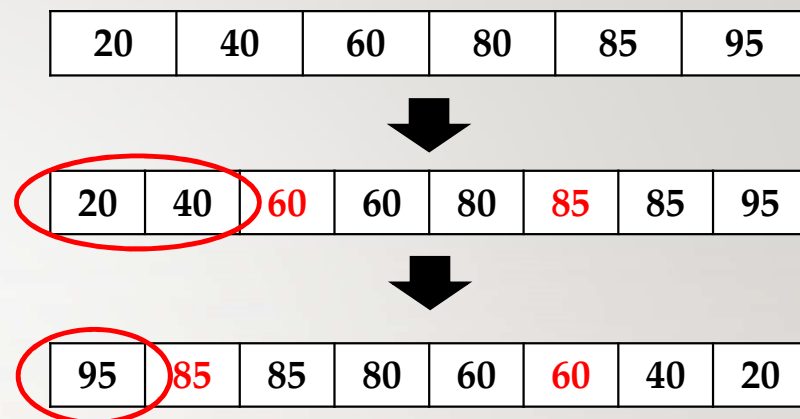


注意：index返回值为第一个出现此值的索引，因此这个方法并不严谨

方法II

如果不明确有60、85分

```
scores = list(eval(input()))
scores.append(60); scores.append(85)
scores.sort()
d = scores.index(60)
d_ave = sum(scores[:d]) / len(scores[:d])
print("The number of D is:",d_ave)
scores.reverse()
a = scores.index(85)
a_ave = sum(scores[:a]) / len(scores[:a])
print("The number of A is:",a_ave)
```



如果 \geq 某个分数，升序求索引；如果 $>$ 某个分数，降序求索引

方法III

```
grade = list(eval(input()))
j = k = 0; grade_j = grade_k = 0
for i in grade:
    if i < 60:
        j += 1
    grade_j += i
    if i >= 85:
        k += 1
    grade_k += i
print("The number of failed is:", grade_j / j)
print("The number of outstanding is:", grade_k / k)
```

循环语句，学到后面可以参考一下

注意内建函数sorted()函数与列表的内建函数sort()alist.sort()会直接对alist进行变化，而sorted函数并不会。

6.请写出可生成以下4个数列的range()函数

- | | |
|--|-------------------------|
| (1) [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] | <u>range(1, 11)</u> |
| (2) [2, 4, 6, 8, 10, 12, 14, 16, 18] | <u>range(2, 20, 2)</u> |
| (3) [0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50] | <u>range(0, 51, 5)</u> |
| (4) [10, 9, 8, 7, 6, 5, 4, 3, 2, 1] | <u>range(10, 0, -1)</u> |

Syntax: range(初始值[包括],结束值[不包括],步长)

问题：①注意不包括结束值，即使是步长为负也是一样。

②题目要求全部用range函数表示，有些同学第4题用了reverse，虽然结果是对的，但是不符合题目要求。

变量名的定义

不要使用诸如list、str、int等函数名作为变量名！这样会导致使用list等函数时函数无法使用！变量名也不能以数字开头！诸如1st是错误的！

```
In[3]: list(range(1,11))
Out[3]: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
In[4]: type(list)
Out[4]: type
In[5]: list=[1,2,3]
In[6]: list(range(1,11))
Traceback (most recent call last):
  File "C:\ProgramData\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py", line 2862, in run_code
    exec(code_obj, self.user_global_ns, self.user_ns)
  File "<ipython-input-6-4fcf9298909f>", line 1, in <module>
    list(range(1,11))
TypeError: 'list' object is not callable
In[7]: type(list)
Out[7]: list
```

在In[5]后，list就不再是函数了，而是一个列表！