Python一本通

信息技术学考教程

第7章　字符串

Python中的字符串(string)类型是指用单引号、双引号或三引号括号起来的数据。现在开始介绍字符串的基本操作。

一、字符串运算符

字符串运算符用于对字符串进行运算，包括

连接运算符(+)、复制运算符(*)、成员运算符(in/not in)三种。

1.连接运算符(+)用于连接字符串。

例如：

```
>>>"Hello" + ",Python!"
'Hello,Python!'
```

一、字符串运算符

字符串运算符用于对字符串进行运算，包括

连接运算符(+)、复制运算符(*)、成员运算符(in/not in)三种。

2.复制运算符(*)用于复制字符串。

例如：

```
>>>"Welcome!" * 3
'Welcome!Welcome!Welcome!'
```

一、字符串运算符

字符串运算符用于对字符串进行运算，包括

连接运算符(+)、复制运算符(*)、成员运算符(in/not in)三种。

3.成员运算符(in/not in)用于判断是否为字符串的子串。

例如：

```
>>>"y" in "Python"
True
>>>"th" not in "Python"
False
```

【说明】

成员运算符in表示如果字符串包含指定的字符，结果返回True；否则，返回False。成员运算符not in表示如果字符串中不包含指定的字符，结果返回True；否则，返回False。

二、字符串的索引

字符串的索引是指返回字符串的单个字符。

字符串索引的格式如下：

变量名[索引下标]

【说明】

(1)字符串有两种索引方式，分别是正索引(左→右)和负索引(右→左)。

(2)正索引时，索引下标(左→右)从0开始编号(0,1,2,3,…,n-1)，依次递增。

(3)负索引时，索引下标(右→左)从-1开始编号(-1,-2,-3,…,-n)，依次递减。

二、字符串的索引

例如：

```
>>>st= "Hello Python"                    #定义一个字符串变量
>>>print(st[0],st[4],st[6],st[9])        #正索引下标从0开始编号
H o P h

>>>print(st[-1],st[-6],st[-5])           #负索引下标从-1开始编号
n P y
```

字符串的正索引和负索引访问字符串的方式，如下表所示：

字符串st	H	e	l	l	o		P	y	t	h	o	n
正索引(左→右)	0	1	2	3	4	5	6	7	8	9	10	11
负索引(右→左)	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

三、字符串的切片

字符串的切片是指返回字符串中一段字符子串。

字符串切片的格式如下：

变量名[下标1:下标2:步长]

【说明】

(1)下标1表示切片的起始位置，默认值为0。

(2)下标2表示切片的终止位置(但不包括这个位置)，也就是说字符串的切片操作只能访问到终止值前面一个元素。

(3)步长表示访问字符的间隔，默认值为1。

例如：

```
>>>st= "Hello Python"          #定义一个字符串变量
>>>st[0:10:3]                  #下标从0到10(不包含10)为止，步长为3
'HlPh'
```

四、字符串操作函数

函　数	描　述
s.index(sub[,start,end])	返回子串sub在s里第一次出现的位置
s.find(sub[,start,end])	与index函数一样，但如果找不到会返回-1
s.replace(old,new[,count])	将s里所有old子串替换为new子串，count指定替换多少个子串
s.count(sub[,start,end])	统计s里有多少个sub子串
s.split()	用分隔符将字符串分开，默认分隔符是空格
s.lower()	将字符串中的大写字母变成小写字母
s.upper()	将字符串中的小写字母变成大写字母
sep.join(sequence)	把序列sequence中的字符串用连接符sep连接起来

五、字符串操作实例

(1)index举例

```
>>> s = "Python"
>>> s.index('P')
0
>>> s.index('h',1,4)
3
```

(2)find()举例

```
>>> s = "Python"
>>> s.find('s')
-1
>>> s.find('t',1)
2
```

五、字符串操作实例

(3)replace()举例

```
>>> s = "Python"
>>> s.replace('h','i')
'Pytion'
```

(4)count()举例

```
>>> s = "Python"
>>> s.count('t')
1
```

五、字符串操作实例

(5)split()举例

```
>>> s = "hello Python i like it"
>>> s.split()
['hello', 'Python', 'i', 'like', 'it']
```

(6)join()举例

```
>>> li = ['apple','peach','banana','pear']
>>> sep = ','
>>> sep.join(li)                #连接列表元素
'apple,peach,banana,pear'
```

六、字符串应用举例

【例7.1】统计单词个数

【问题描述】

输入一段字符，统计其中单词的个数，单词之间用空格分隔。

【输入格式】

输入一段英文单词组成字符。

【输出格式】

输出单词的个数。

【样例输入】

```
Python is an object-oriented programming language
```

【样例输出】

```
6
```

【题目分析】

一个不含空格字符的字符串就是一个单词。将连续的若干个空格看作一个空格，因此，单词的个数可以由空格数来决定。如果当前字符不是空格，而它的前一个字符是空格，便认为是新单词的开始，累计单词个数的变量加1；如果当前字符不是空格，而它的前一个字符也不是空格，则认为是当前单词的继续，累计单词个数的变量保持不变。

【代码实现】

```
str=input()                          #输入一段英文字符
flag=0
count=0
for c in str:
    if c==" ":
        flag=0
    else:
        if flag==0:
            flag=1
            count=count+1            #单词个数加1
print(count)                         #输出单词个数
```

【例7.2】统计几种字符个数

【问题描述】

输入一行字符，分别统计英文字母、空格、数字和其他字符的4种个数。

【输入格式】

输入一行字符，包含英文字母、空格、数字和其他字符。

【输出格式】

输出字符统计的个数，每行1种。

【样例输入】

Python 3.6.0中文版

【样例输出】

6
1
3
5

【题目分析】

根据字符串中每个字符的ASCII码值判断其类型。数字0~9对应的ASCII码值为48~57，大写字母A~Z对应的ASCII码值为65~90，小写字母A~Z对应的ASCII码值为97~122。使用ord()函数将字符串转换换为ASCII码值。

【例7.2】统计几种字符个数

【代码实现】

```
a_list=list(input())          #输入一行字符串
letter=[]
space=[]
number=[]
other=[]
for i in range(len(a_list)):
    if ord(a_list[i]) in range(65,91) or ord(a_list[i]) in range(97,123):
        letter.append(a_list[i])
    elif a_list[i]==' ':
        space.append(' ')
    elif ord(a_list[i]) in range(48,58):
        number.append(a_list[i])
    else:
        other.append(a_list[i])
print(len(letter))            #英文字母个数
print(len(space))             #空格个数
print(len(number))            #数字个数
print(len(other))             #其他字符个数
```

【例7.3】替换字母

【题目描述】

在应用计算机编辑文档的时候，我们经常遇到替换任务。如把文档中的“电脑”都替换成“计算机”。现在请你编程模拟一下这个操作。

【输入格式】

输入两行内容，第1行是原文(长度不超过200个字符)，第2行包含以空格分隔的两个字符A和B，要求将原文中所有的字符A都替换成字符B，注意：区分大小写字母。

【输出格式】

一行，输出替换后的结果。

【样例输入】

```
I love China. I love Beijing.
I U
```

【样例输出】

```
U love China. U love Beijing.
```

【代码实现】

```
# 读入原文
article=input()
# 读入两个字符串
A,B=input().split()
# 将article中的A字符串换成B字符串
print(article.replace(A,B))
```

【例7.4】字符串判等【1.7编程基础之字符串17】

【问题描述】

判断两个由大小写字母和空格组成的字符串在忽略大小写，且忽略空格后是否相等。

【输入格式】

两行，每行包含一个字符串。

【输出格式】

若两个字符串相等，输出YES，否则输出NO。

【样例输入】

```
a A bb BB ccc CCC
Aa BBbb CCCccc
```

【样例输出】

```
YES
```

【代码实现】

```
# input().split(): 将读入的字符串用空格分开
# ''.join() 删去字符串中的空格
A=''.join(input().split())
B=''.join(input().split())
# A.upper(): 将A中字符全部转化为大写
if A.upper()==B.upper():
    print("YES")
else:
    print("NO")
```

【例7.5】验证子串【1.7编程基础之字符串18】

【问题描述】

输入两个字符串，验证其中一个串是否为另一个串的子串。

【输入格式】

输入两个字符串， 每个字符串占一行，长度不超过200且不含空格。

【输出格式】

若第一个串s1是第二个串s2的子串，则输出(s1) is substring of (s2)

否则，若第二个串s2是第一个串s1的子串，输出(s2) is substring of (s1)

否则，输出 No substring。

【样例输入】

abc

dddncabca

【样例输出】

abc is substring of dddncabca

【代码实现】

```
s1=input()
s2=input()
if s1.find(s2)!=-1:
    # s2是s1的子串
    print("%s is substring of %s"%(s2,s1))
elif s2.find(s1)!=-1:
    # s1是s2的子串
    print("%s is substring of %s"%(s1,s2))
else:
    print('No substring')
```

【课堂练习】

练7.1　统计数字字符个数【1.7编程基础之字符串01】

【问题描述】
　　输入一行字符，统计出其中数字字符的个数。
【输入格式】
　　一行字符串，总长度不超过255。
【输出格式】
　　输出为1行，输出字符串里面数字字符的个数。
【样例输入】
　　Peking University is set up at 1898.
【样例输出】
　　4
【代码实现】

【代码实现】

```
st=input()
num=0
# 枚举st中的字符
for i in st:
    # 判断是否为数字字符
    if i>='0' and i<='9':
        num+=1
print(num)
```

练7.2　找第一个只出现一次的字符【1.7编程基础之字符串02】

【问题描述】

给定一个只包含小写字母的字符串，请你找到第一个仅出现一次的字符。如果没有，输出no。

【输入格式】

一个字符串，长度小于100000。

【输出格式】

输出第一个仅出现一次的字符，若没有则输出no。

【样例输入】

abcabd

【样例输出】

c

【代码实现】

```
st=input()
num={}    # num是一个字典，用于记录每个字符出现次数

for i in st:    # 枚举字符串st中的每个字符
    if i not in num:    # 如果i不在字典中
        num[i]=0        # 在字典中加入字符i
    num[i]+=1    # 字符i出现次数加一

flag=False
for i in st:
    if num[i]==1:    # i是第一个仅出现一次的字符
        print(i)
        flag=True
        break
if not flag:
    print("no")
```

练7.3　基因相关性【1.7编程基础之字符串03】

【问题描述】

为了获知基因序列在功能和结构上的相似性，经常需要将几条不同序列的DNA进行比对，以判断该比对的DNA是否具有相关性。

现比对两条长度相同的DNA序列。定义两条DNA序列相同位置的碱基为一个碱基对，如果一个碱基对中的两个碱基相同的话，则称为相同碱基对。接着计算相同碱基对占总碱基对数量的比例，如果该比例大于等于给定阈值时则判定该两条DNA序列是相关的，否则不相关。

【输入格式】

有三行，第一行是用来判定出两条DNA序列是否相关的阈值，随后2行是两条DNA序列(长度不大于500)。

【输出格式】

若两条DNA序列相关，则输出“yes”，否则输出“no”。

【样例输入】

```
0.85
ATCGCCGTAAGTAACGGTTTTAAATAGGCC
ATCGCCGGAAGTAACGGTCTTAAATAGGCC
```

【样例输出】

```
yes
```

【代码实现】

```
x=float(input())
s1=input()
s2=input()
tot=0
# len(s1):字符串长度
# 从头枚举字符串的碱基是否相同
for i in range(len(s1)):
    if (s1[i]==s2[i]):
        tot+=1
# tot/len为相同碱基对
# 占总碱基对数量的比例
if (tot/len(s1)>=x):
    print("yes")
else:
    print("no")
```

练7.5　输出亲朋字符串【1.7编程基础之字符串05】

【问题描述】

编写程序，求给定字符串s的亲朋字符串s1。

亲朋字符串s1定义如下：给定字符串s的第一个字符的ASCII值加第二个字符的ASCII值，得到第一个亲朋字符；　给定字符串s的第二个字符的ASCII值加第三个字符的ASCII值，得到第二个亲朋字符；依此类推，直到给定字符串s的倒数第二个字符。亲朋字符串的最后一个字符由给定字符串s的最后一个字符ASCII值加s的第一个字符的ASCII值。

【输入格式】

输入一行，一个长度大于等于2，小于等于100的字符串。字符串中每个字符的ASCII值不大于63。

【输出格式】

输出一行，为变换后的亲朋字符串。输入保证变换后的字符串只有一行。

【样例输入】

1234

【样例输出】

cege

【代码实现】

```
a=input()
n=len(a)
# 处理最后一位字符的情况
a=a+a[0]
b=''
for i in range(n):
    # ord函数将字符转化为Unicode码
    x=ord(a[i])+ord(a[i+1])
    # chr函数将整数看作Unicode码转化为字符
    b=b+chr(x)
    # 将字符接在b的末尾
print(b)
```

练7.6　合法C标识符【1.7编程基础之字符串06】

【问题描述】

给定一个不包含空白符的字符串，请判断是否是C语言合法的标识符号(注：题目保证这些字符串一定不是C语言的保留字)。

C语言标识符要求：

1.非保留字；

2.只包含字母、数字及下划线（“_”）。

3.不以数字开头。

【输入格式】

一行包含一个字符串，字符串中不包含任何空白字符，且长度不大于20。

【输出格式】

一行，如果它是C语言的合法标识符，则输出yes，否则输出no。

【样例输入】

RKPEGX9R;TWyYcp

【样例输出】

no

【代码实现】

```
s=input()
flag=True
if s[0]>='0' and s[0]<='9':# 判断是否以数字开头
    flag=False
for c in s:
    if c>='0' and c<='9':  # 判断是否是数字
        continue
    if c>='A' and c<='Z':  # 判断是否是大写字母
        continue
    if c>='a' and c<='z':  # 判断是否是小写字母
        continue
    if c=='_':             # 判断是否是下划线
        continue
    flag=False
    break
if flag:
    print("yes")
else:
    print('no')
```

练7.7　配对碱基链【1.7编程基础之字符串07】

【问题描述】

脱氧核糖核酸(DNA)由两条互补的碱基链以双螺旋的方式结合而成。而构成DNA的碱基共有4种，分别为腺嘌呤(A)、鸟嘌呤(G)、胸腺嘧啶(T)和胞嘧啶(C)。我们知道，在两条互补碱基链的对应位置上，腺嘌呤总是和胸腺嘧啶配对，鸟嘌呤总是和胞嘧啶配对。你的任务就是根据一条单链上的碱基序列，给出对应的互补链上的碱基序列。

【输入格式】

一个字符串，表示一条碱基链。这个字符串只含有大写字母A、T、G、C，分别表示腺嘌呤、胸腺嘧啶、鸟嘌呤和胞嘧啶。字符串长度不超过255。

【输出格式】

一个只含有大写字母A、T、G、C的字符串，为与输入的碱基链互补的碱基链。

【样例输入】

ATATGGATGGTGTTTGGCTCTG

【样例输出】

TATACCTACCACAAACCGAGAC

【代码实现】

```
s=input()
t=''
for c in s:
    # 判断碱基的配对碱基
    if c=='A':
        t+='T'
    if c=='T':
        t+='A'
    if c=='C':
        t+='G'
    if c=='G':
        t+='C'
print(t)
```

练7.8　密码翻译【1.7编程基础之字符串09】

【问题描述】

在情报传递过程中，为了防止情报被截获，往往需要对情报用一定的方式加密，简单的加密算法虽然不足以完全避免情报被破译，但仍然能防止情报被轻易的识别。我们给出一种最简的的加密方法，对给定的一个字符串，把其中从a-y，A-Y的字母用其后继字母替代，把z和Z用a和A替代，其他非字母字符不变，则可得到一个简单的加密字符串。

【输入格式】

输入一行，包含一个字符串，长度小于80个字符。

【输出格式】

输出每行字符串的加密字符串。

【样例输入】

Hello! How are you!

【样例输出】

Ifmmp! Ipx bsf zpv!

【代码实现】

```
s=input()
t=''
for c in s:
    if c>='A' and c<'Z' or c>='a' and c<'z':
        # 用后继字母替代
        t+=chr(ord(c)+1)
    elif c=='Z':
        # 把Z用A替代
        t+='A'
    elif c=='z':
        # 把z用a替代
        t+='a'
    else:
        t+=c
print(t)
```

练7.9　将字符串中的小写字母转换成大写字母【1.7编程基础之字符串13】

【问题描述】

给定一个字符串，将其中所有的小写字母转换成大写字母。

【输入格式】

输入一行，包含一个字符串（长度不超过100，可能包含空格）。

【输出格式】

输出转换后的字符串。

【样例输入】

helloworld123Ha

【样例输出】

HELLOWORLD123HA

【代码实现】

```
s=input()
t=''
for c in s:
    if c>='a' and c<='z':
        # 将c转化为大写字母
        t+=c.upper()
    else:
        # 其余字符不变
        t+=c
print(t)
```

练7.10　整理药名【1.7编程基础之字符串15】

【问题描述】

医生在书写药品名的时候经常不注意大小写，格式比较混乱。现要求你写一个程序将医生书写混乱的药品名整理成统一规范的格式，即药品名的第一个字符如果是字母要大写，其他字母小写。如将ASPIRIN、aspirin整理成Aspirin。

【输入格式】

第一行一个数字n，表示有n个药品名要整理，n不超过100。

接下来n行，每行一个单词，长度不超过20，表示医生手书的药品名。药品名由字母、数字和-组成。

【输出格式】

n行，每行一个单词，对应输入的药品名的规范写法。

【样例输入】

```
4
AspiRin
cisapride
2-PENICILLIN
Cefradine-6
```

【样例输出】

```
Aspirin
Cisapride
2-penicillin
Cefradine-6
```

【代码实现】

```
n=int(input())
for i in range(n):
    t=""
    s=input()
    c=s[0]
    if c>='a' and c<='z':
        # 将c转化为大写字母
        t=c.upper()
    else:
        # 其余字符不变
        t=c
    for c in s[1:]:
        if c>='A' and c<='Z':
            # 将c转化为小写字母
            t+=c.lower()
        else:
            # 其余字符不变
            t+=c
    print(t)
```