



第三章 字符串、队列和栈

课时1 字符串

课时目标

- 1.通过问题解决，理解字符串的概念和特性。
- 2.掌握字符串的基本操作，并能编程实现。

目录

CONTENTS

01 知识梳理

02 例题精析

03 随堂检测

04 巩固与提升

1

知识梳理

1. 字符串的概念

- (1) 字符串是由零个或多个字符组成的有限序列。
- (2) 字符串长度：字符串中所包含字符的个数。
- (3) 主要形式有：'Python'、"Python"、"""Python"""。

■ 归纳总结 ■

正确区分空串和空格串

空串是指不包含任何字符的串，其长度为零。而空格串是指由空格符组成的字符串，其长度为空格字符的个数。

2.字符串的特性

(1)有限序列性

字符串的数据元素个数是有限的。

(2)字符串的可比性

若比较的字符串中的字符为ASCII字符集，则按字符的ASCII码值进行比较。

3.字符串的基本操作

(1)子串判断

子串判断用于检查字符串a是否为字符串b的子串，可用in运算实现。

(2)求子串

求子串用于从字符串中取出其子串，可用切片实现。

(3)字符串的连接

字符串的连接用于把两个字符串连接在一起，用“+”号实现。

4. Python字符串常用函数和方法

Python字符串常用函数和方法

| 函数和方法 | 功能 | 实例 |
|--|--|---|
| <code>len(x)</code> | 统计字符串x中字符的个数 | <code>x = "Python"</code> <code>len(x)</code> 的值为6 |
| <code>x.find(y)</code> | 返回字符串x中子串y出现的首字符下标，若找不到，则输出 - 1 | <code>x = "Python"</code> <code>y = "on"</code> <code>x.find(y)</code> 返回的值为4 |
| <code>x.split(str = "" , num = string.count(str))</code> | 以str分隔符切片x，若有num的指定值，则表示仅分隔num + 1个子串，否则分隔所有子串 | <code>x = "1 , 2 , 3 , 4"</code> <code>y = x.split(" , ")</code> y列表的内容为： <code>['1' , '2' , '3' , '4']</code> |

| | | |
|--|---|--|
| <p><code>x.replace</code> (<code>old</code> , <code>new</code>[, <code>max</code>])</p> | <p>返回字符串中的<code>old</code>替换成<code>new</code>后的新字符串，若指定<code>max</code>，则表示替换不超过<code>max</code>次</p> | <pre>x = "This data is new data." x.replace("data" , "bag")</pre> <p>执行后x字符串为：'This bag is new bag.'</p> |
| <p><code>x.upper()</code></p> | <p>把字符串x中的小写字母转换成大写字母</p> | <pre>x = "Python" x.upper()</pre> <p>执行后x字符串为： 'PYTHON'</p> |
| <p><code>x.lower()</code></p> | <p>把字符串x中的大写字母转换成小写字母</p> | <pre>x = "Python" x.lower()</pre> <p>执行后x字符串为： 'python'</p> |

5.正则表达式

(1)正则表达式是用一些特定的字符组成的一个“规则字符串”。

(2)作用：它可以实现检查一个字符串中是否含有某种子串、将匹配的子串替换或从字符串中取出符合某个条件的子串等操作。

2

例题精析

例1 下列有关字符串描述正确的是(**B**)

- A.空串指由一个空格字符组成的字符串
- B.字符串中的字符可以按ASCII码值进行比较
- C.表达式s[3] = "a"的功能是将字符串s第4个位置的字符串修改为"a"
- D.字符串a各个字符出现在字符串b中，则表达式a in b的值为True

解析 本题考查字符串的特性。A选项空串是指不包含任何字符的串，其长度为零。B选项字符串具有可以比较性，按ASCII码值进行比较。C选项字符串的内容不能修改，只能重新连接。D选项字符串中字符还包含位置，如"ac"不是"abc"的子串。

变式训练 下列有关字符串的描述正确的是(**B**)

A.字符串是一种非线性表结构

B.字符串是由 n ($n \geq 0$ 且为整数)个字符组成的有限序列

C.字符串“我的Python”的长度为10

D.已知字符串 $x = \text{"Python"}$ ，则 $x[1]$ 表示字符“P”

解析 本题主要考查的是字符串的特性。字符串是一种线性表结构，因此，选项A错误；字符串“我的 Python”的长度为8，一个中文或英文字符的长度均为1，因此，C选项错误；已知字符串 $x = \text{"Python"}$ ，则 $x[1]$ 表示字符“y”，因为字符串的第一个位置的索引号为0，因此D选项错误；字符串是由零个或多个字符组成的有限序列，即字符串是由 n ($n \geq 0$ 且为整数)个字符组成的有限序列，因此，答案为B。

例2 十八位身份证号由六位数字地址码、八位数字出生日期码、三位数字顺序码和一位校验码组成，其中倒数第二位是性别代码，男单女双。以下 Python 程序的功能为识别身份证中的出生日期和性别信息。则加框①②处的代码依次为(**B**)

```
s = input("输入身份证号")
```

```
xb = ['女', '男']
```

```
print('出生日期为：', s[6 : 10] + '年' + ① + '月' + s[12 : 14] + '日')
```

```
print('性别为：', ②)
```

- A. ①s[10 : 11] ②xb[int(s[- 2])%2]
- B. ①s[10 : 12] ②xb[int(s[16])%2]
- C. ①s[- 12 : 10] ②xb[int(s[16])%2]
- D. ①s[- 12 : - 6] ②xb[(int(s[- 2]) + 1)%2]

解析 本题考查字符串的切片。十八位身份证号由六位数字地址码、八位数字出生日期码，出生日期的开始位置索引号为6，月的开始位置索引号为10，取两位，因此为s[10: 12]。倒数第二位是性别代码，因此索引号为16。

变式训练 有如下Python程序段：

```
x = "Python"
```

```
y = ""
```

```
for i in range(len(x)):
```

```
    y = x[i] + y
```

程序段执行后，变量y的值为(**C**)

A.Python

B."Python"

C."nohtyP"

D.nohtyP

解析 本题主要考查的是字符串组操作。本程序的功能是将字符串x进行反转操作，故答案为C。

例 3 若字符串 `s` 的值为 "abcde" 执行如下程序段后 变量 `res` 的值不可能是(**B**)

```
from random import randint
```

```
res = ""
```

```
i, j = 0, len(s) - 1
```

```
while i < len(s) and j >= i :
```

```
    if randint(0, 1) == 0: #randint(0, 1)随机生成 0 或 1
```

```
        res += s[i]
```

```
        i += 1
```

else :

res + = s[j]

j - = 1

A."abcd"

B."aecbd"

C."aedbc"

D."edcba"

解析 变量i和j从两端向中间遍历，当i大于j时结束循环。若产生随机数为0，将s[i]连接到res中，向后移动i，否则将s[j]连接到res中，向前移动j。A选项产生的随机数为0，0，0，1，0。B选项产生"ae"，i和j分别指向b和d，因此不可能连接到c。C选项产生的随机数为0，1，1，0，0或0，1，1，0，1。D选项产生的随机数均为1或1，1，1，1，0。

变式训练 有如下Python程序段：

```
s = "abnokycmhq"  
m = int(input("输入一个数："))  
j = 0 ; sm = ""  
for i in range(len(s)) :  
    if i - j == m :  
        sm = sm + s[j : i][ : : - 1]  
        j = i  
    if i - j + 1 == m :  
        sm = sm + s[j : i + 1][ : : - 1]
```

else :

```
    sm = s[j : i + 1] + sm
```

```
print(sm)
```

若输入m的值为3，则以上程序运行后，输出结果为(**D**)

A.nbaykqohmc

B.qhmcykonba

C.nbaykohmcq

D.qnbaykohmc

解析 条件 $i-j==m$ 表示m个长度的切片，表达式 $s[j:i][::-1]$ 表示将该切片进行反向连接。选择结构表示最后一段的切片长度，若条件 $i-j+1==m$ 成立表示长度为m-1个，还是要反向，否则把剩余一段取出，放在sm的前面。而s按3个一段分，最后剩下字母q。

例4 有如下Python程序段：

```
code = "0123456789ABCDEF"  
RGB = "255 , 96 , 8".split(" , ")  
ans = "#"  
for c in RGB :  
    s1 = "" ; t = int(c)  
    for i in range(2) :  
        s1 = code[t%16] + s1  
        t// = 16  
    ans + = s1  
print(ans)
```

执行程序后，输出的结果为(**B**)

A.#FF0608

B.#FF6008

C.#8060FF

D.#6E60FF

解析 本题考查字符串的基本操作。RGB是将字符串按逗号分隔，形成一个列表['255', '96', '8']。程序的功能是将列表中三个字符串转换成数值类型，并用十六进制表示。for i in range(2)表示循环2次，若转换后十六进制只有一位数字，也要在前面加0。语句s1=code[t%16]+s1表示将取出的余数反向连接。

变式训练 有如下 Python 程序段：

```
qiu = ""  
x = "394 - 5 - 222 - 60 - 809".split(" - ")  
for ch in x :  
    if qiu<ch :  
        qiu = ch + qiu  
print(qiu)
```

执行该程序段后，输出结果是(**C**)

A.1203

B.605394

C.809605394

D.809394

解析 本题考查字符串基本操作。x是将字符串按“-”分隔，形成一个列表 ['394', '5', '222', '60', '809']。遍历列表，查找一个连续递增的字符串，并将该字符串连接在qiu的前面。

3

随堂检测

1.已知s = "Python is so easy ! " , 下列表达式的值是"is"的是(C)

A.s[8 : 9]

B.s[7 : 8]

C.s[7 : 9]

D.s[8 : 10]

解析 本题主要考查的是字符串的切片操作。字符串中第一个字符的索引号为0，切片操作时，只能取到下标end前面位置上的字符，s[7: 9]的值为"is"，因此，答案为C。

2. 有如下 Python 程序段：

```
s = "CixiStudent"
```

```
f = [0]*26 ; t = s[4 : ]
```

```
i = 0
```

```
while i < len(t) :
```

```
    if "A" <= t[i] <= "Z" :
```

```
        i + = 1
```

```
        continue
```

```
    elif "a" <= t[i] <= "z" and f[ord(t[i]) - ord("a")] == 0 :
```

```
        f[ord(t[i]) - ord("a")] + = 1
```

```
i += 1
```

```
for i in range(26):
```

```
    if f[i] == 1:
```

```
        print(chr(i + ord("a")), end = "")
```

执行以上程序段后，输出的内容为(**A**)

A.dentu

B.Student

C.tuden

D.deintux

解析 f数组分别存储小写字母a—z的个数。如果是大写字母，则跳过，如果是小写字母，且没有出现过，列表对应位置加1，再从小到大输出对应字母。

3.英语单词“illustriousness”意思是优秀，有如下Python程序段：

```
import random  
s = "illustriousness"  
for i in range(4):  
    k = random.randint(0, 4)*random.randint(0, 3)  
    print(s[k], end = "")
```

执行该程序后，输出的结果不可能是(**C**)

A.uiso B.iiii C.tree D.siur

解析 k 的值为0至4之间的数乘以0至3之间的数， $[0, 4]*0$ 值均为0， $[0, 4]*1$ 值依次为0、1、2、3、4， $[0, 4]*2$ 值依次为0、2、4、6、8， $[0, 4]*3$ 值依次为0、3、6、9、12，因此不可能产生5，7，10，11，即字符中不可能包含t。

4. 现有一个只包含字符“A”“B”的字符串s，并且两类字符的总数量刚好相等。将其尽可能地分割成多段，使得每一段中包含相同数量的“A”、“B”，输出每一段的范围。例如，当s = "ABAABABBAB"时，根据要求最多可以分割成3段，分别为[0, 1], [2, 7], [8, 9]，如下Python程序段实现该功能。

```
s = "ABAABABBAB"
```

```
c = 0 ; pos = 0
```

```
for i in range(len(s)) :
```

```
    if s[i] == "A" :
```

```
        c = c + 1
```

```
if (1) _____ :  
    print("[", pos, ", ", " , i, "]")
```

(2) _____

上述程序段中划线处可选语句为：① $i - pos = c$

② $i - c = c$ ③ $i - c + 1 = c$ ④ $pos = i$ ⑤ $pos = i + 1$

则(1)(2)处的语句依次可为(**D**)

A. ①⑤

B. ②④

C. ③④

D. ③⑤

解析 当 $s[i]$ 为 "A" 时, c 每次加 1, 即 c 是所有 "A" 的数量。每一段中包含相同数量的 "A"、"B", 当前共 $i+1$ 个字母中, "B" 的数量也应该是 c 个, 即 $i+1 = 2*c$ 。 pos 是每段开始位置, 当前位置 i 是每段结束位置, 因此 pos 的值为 $i+1$ 。

5.有如下程序段：

```
s = "abAB ! AB" ; k = x = ""
```

```
for i in s :
```

```
    if "a"<= i<= "z" :
```

```
        x = chr((ord(i) - 98)%26 + 97)
```

```
    elif "A"<= i<= "Z" :
```

```
        x = chr((ord(i) - 40)%26 + 65)
```

```
        k + = x
```

```
    print(k)
```

执行程序后，输出值为(**D**)

A.bcBC ! BC

B.bcBCCBC

C.zaZA ! ZA

D.zaZAAZA

解析 本题考查字母的加密。a的内码为97， $\text{ord}(i)-98$ 等效于 $\text{ord}(i)-97-1$ ，表达式 $(\text{ord}(i)-98)\%26$ 表示字母中位置循环向前移动一个位置。a的内码为65， $\text{ord}(i)-40$ ，等效于 $\text{ord}(i)-65+25$ ，等效于 $\text{ord}(i)-65+26-1$ 。程序的功能是将字母循环向前移一位，非字母不连接，用该字母前一个字母代替。

6.编写一个Python程序，实现查找替换功能，下列程序的功能是：在text1字符串中，将所有的good替换为ok。

实现上述功能的程序如下，请在程序划线处填入合适的代码。

```
text1 = "it's good , good , very very good."
```

```
fword = "good"
```

```
cword = "ok"
```

```
text2 = ""
```

```
n1 = len(text1)
```

```
① _____
```

```
i = 0
```

```
while i <= n1 - len(fword) :  
    if ② _____ :  
        text2 = text2 + cword  
        i = i + n2  
    else :  
        ③ _____  
        i + = 1  
if i < n1 :  
    text2 = text2 + text1[i : n1 + 1]  
print(text2)
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：
<https://d.book118.com/418055107112007003>