

MÓDULO VI - STRINGS

Prof. Marília Nestor: nestor.marilia@gmail.com

O que é uma string para Python?

- Lista de caracteres
- Lista imutável de pontos de código UNICODE
- Tupla com funcionalidades específicas
- Também podemos usar a função len()
- Quando se altera uma string, se cria uma nova string
- Não é possível se alterar como se altera a lista, em atribuição de valores

ASCII

0 NUL	32 espace	64 @	96 `
1 SOH	33 !	65 A	97 a
2 STX	34 "	66 B	98 b
3 ETX	35 #	67 C	99 c
4 EOT	36 \$	68 D	100 d
5 ENQ	37 %	69 E	101 e
6 ACK	38 &	70 F	102 f
7 BEL	39 '	71 G	103 g
8 BS	40 (72 H	104 h
9 HT	41)	73 I	105 i
10 LF	42 *	74 J	106 j
11 VT	43 +	75 K	107 k
12 FF	44 ,	76 L	108 l
13 CR	45 -	77 M	109 m
14 SO	46 .	78 N	110 n
15 SI	47 /	79 O	111 o
16 SLE	48 0	80 P	112 p
17 CS1	49 1	81 Q	113 q
18 DC2	50 2	82 R	114 r
19 DC3	51 3	83 S	115 s
20 DC4	52 4	84 T	116 t
21 NAK	53 5	85 U	117 u
22 SYN	54 6	86 V	118 v
23 ETB	55 7	87 W	119 w
24 CAN	56 8	88 X	120 x
25 EM	57 9	89 Y	121 y
26 SIB	58 :	90 Z	122 z
27 ESC	59 ;	91 [123 {
28 FS	60 <	92 \	124
29 GS	61 =	93]	125 }
30 RS	62 >	94 ^	126 ~
31 US	63 ?	95 _	127 ■

Unicode

	000	001	002	003	004	005	006	007
0	␣ [NUL]	␣ [DLE]	␣ [SP]	0	@	P	`	p
1	␣ [STX]	␣ [DC1]	!	1	A	Q	a	q
2	␣ [SOT]	␣ [DC2]	"	2	B	R	b	r
3	␣ [ETX]	␣ [DC3]	#	3	C	S	c	s
4	␣ [EOT]	␣ [DC4]	\$	4	D	T	d	t
5	␣ [ENQ]	␣ [NAK]	%	5	E	U	e	u
6	␣ [ACK]	␣ [SYN]	&	6	F	V	f	v
7	␣ [BEL]	␣ [ETB]	'	7	G	W	g	w
8	␣ [BS]	␣ [CAN]	(8	H	X	h	x
9	␣ [HT]	␣ [EM])	9	I	Y	i	y
A	␣ [LF]	␣ [SUB]	*	A	J	Z	j	z
B	␣ [VT]	␣ [ESC]	+	B	[k	{	
C	␣ [FF]	␣ [FS]	,	C	<	L	\	
D	␣ [CR]	␣ [GS]	-	D	=	M]	m
E	␣ [SO]	␣ [RS]	.	E	>	N	^	n
F	␣ [SI]	␣ [US]	/	F	?	O	_	o

	008	009	00A	00B	00C	00D	00E	00F
0	␣ [CTRL]	␣ [CTRL]	␣ [NB SP]	°	À	Ð	à	ð
1	␣ [CTRL]	␣ [CTRL]	¡	±	Á	Ñ	á	ñ
2	␣ [CTRL]	␣ [CTRL]	¢	2	Â	Ò	â	ò
3	␣ [CTRL]	␣ [CTRL]	£	3	Ã	Ó	ã	ó
4	␣ [CTRL]	␣ [CTRL]	¤	´	Ä	Ô	ä	ô
5	␣ [CTRL]	␣ [CTRL]	¥	µ	Å	Õ	å	õ
6	␣ [CTRL]	␣ [CTRL]	¦	¶	Æ	Ö	æ	ö
7	␣ [CTRL]	␣ [CTRL]	§	·	Ç	×	ç	÷
8	␣ [CTRL]	␣ [CTRL]	¨	¸	È	Ø	è	ø
9	␣ [CTRL]	␣ [CTRL]	©	¹	É	Ù	é	ù
A	␣ [CTRL]	␣ [CTRL]	ª	º	Ê	Ú	ê	ú
B	␣ [CTRL]	␣ [CTRL]	«	»	Ë	Û	ë	û
C	␣ [CTRL]	␣ [CTRL]	¼	½	Ì	Ü	ì	ü
D	␣ [CTRL]	␣ [CTRL]	½	¾	Í	Ý	í	ý
E	␣ [CTRL]	␣ [CTRL]	¾	¾	Î	Þ	î	þ
F	␣ [CTRL]	␣ [CTRL]	¿	¿	Ï	ß	ï	ÿ

	110	111	112	113	114	115	116	117
0	␣ [U+1100]	␣ [U+1101]	␣ [U+1102]	␣ [U+1103]	␣ [U+1104]	␣ [U+1105]	␣ [U+1106]	␣ [U+1107]
1	␣ [U+1108]	␣ [U+1109]	␣ [U+110A]	␣ [U+110B]	␣ [U+110C]	␣ [U+110D]	␣ [U+110E]	␣ [U+110F]
2	␣ [U+1110]	␣ [U+1111]	␣ [U+1112]	␣ [U+1113]	␣ [U+1114]	␣ [U+1115]	␣ [U+1116]	␣ [U+1117]
3	␣ [U+1118]	␣ [U+1119]	␣ [U+111A]	␣ [U+111B]	␣ [U+111C]	␣ [U+111D]	␣ [U+111E]	␣ [U+111F]
4	␣ [U+1120]	␣ [U+1121]	␣ [U+1122]	␣ [U+1123]	␣ [U+1124]	␣ [U+1125]	␣ [U+1126]	␣ [U+1127]
5	␣ [U+1128]	␣ [U+1129]	␣ [U+112A]	␣ [U+112B]	␣ [U+112C]	␣ [U+112D]	␣ [U+112E]	␣ [U+112F]
6	␣ [U+1130]	␣ [U+1131]	␣ [U+1132]	␣ [U+1133]	␣ [U+1134]	␣ [U+1135]	␣ [U+1136]	␣ [U+1137]
7	␣ [U+1138]	␣ [U+1139]	␣ [U+113A]	␣ [U+113B]	␣ [U+113C]	␣ [U+113D]	␣ [U+113E]	␣ [U+113F]
8	␣ [U+1140]	␣ [U+1141]	␣ [U+1142]	␣ [U+1143]	␣ [U+1144]	␣ [U+1145]	␣ [U+1146]	␣ [U+1147]
9	␣ [U+1148]	␣ [U+1149]	␣ [U+114A]	␣ [U+114B]	␣ [U+114C]	␣ [U+114D]	␣ [U+114E]	␣ [U+114F]
A	␣ [U+1150]	␣ [U+1151]	␣ [U+1152]	␣ [U+1153]	␣ [U+1154]	␣ [U+1155]	␣ [U+1156]	␣ [U+1157]
B	␣ [U+1158]	␣ [U+1159]	␣ [U+115A]	␣ [U+115B]	␣ [U+115C]	␣ [U+115D]	␣ [U+115E]	␣ [U+115F]
C	␣ [U+1160]	␣ [U+1161]	␣ [U+1162]	␣ [U+1163]	␣ [U+1164]	␣ [U+1165]	␣ [U+1166]	␣ [U+1167]
D	␣ [U+1168]	␣ [U+1169]	␣ [U+116A]	␣ [U+116B]	␣ [U+116C]	␣ [U+116D]	␣ [U+116E]	␣ [U+116F]
E	␣ [U+1170]	␣ [U+1171]	␣ [U+1172]	␣ [U+1173]	␣ [U+1174]	␣ [U+1175]	␣ [U+1176]	␣ [U+1177]
F	␣ [U+1178]	␣ [U+1179]	␣ [U+117A]	␣ [U+117B]	␣ [U+117C]	␣ [U+117D]	␣ [U+117E]	␣ [U+117F]

UFT-8

- Melhor dos dois mundos
- Digite no computador:

```
for x in range(32, 127):  
    print (str(x), ' é ',chr(x))
```

Concatenação de strings

- Adição de strings, de modo a formar uma nova string
- É da esquerda para a direita, por motivos óbvios
- Estrutura: <str> + <str>

Digite no seu computador:

```
A="123"
```

```
B='456'
```

```
C= A + B
```

```
print (C)
```

Qual a diferença para **print (A+B)** ? E se não fossem strings? Como transformar de volta em números?

Interpolação de strings

- Inserção de uma string em outra
- Estruturas:

```
'<String> %s, %(<variável>)'
```

```
'<String>{<indicador>}'.format(<indicador>=<variável>)
```

Digite no computador:

```
nome = 'Marília'
```

```
idade = '23'
```

```
pergunta = 'E vc?'
```

```
saudacao = "Olá! Meu nome é {A} e eu tenho {B} anos.
```

```
%s".format(A=nome,B=idade) %(pergunta)
```

```
print (saudacao)
```

Exercícios

1- Coloque a ? que falta neste enunciado

2- Atribua variáveis diferentes a cada um dos seus nomes e sobrenome. Junte-os utilizando concatenação de strings

Fatiamento de strings

- Igual ao fatiamento de listas
- Primeiro caracter: 0, último: -1

Digite no computador:

```
A='lista'  
print (A[-1])  
print (A[1:3])  
print (A[:3])  
print (A[2:])
```

Funções para strings

- Função `split()` divide o texto
- Estrutura: `<variável>.split(" ")`

```
A = "tuplas sao muito legais!"
```

```
print (A.split(" "))
```

- Função `replace()` substitui o texto desejado
- Estrutura: `<variável>.replace("<string>","<string>")`

```
print (A.replace("tuplas","aulas"))
```

Comparação de strings

Digite um por um no computador:

`ord('a')`

`chr(97)`

`str(97)`

`"a" > "X"`

```
A= "abcd"
```

```
B= "abcd"
```

```
if A!=B:
```

```
    print ("São diferentes")
```

```
else:
```

```
    print ("São iguais")
```

Exercícios

- Escreva um programa que solicite ao usuário a digitação do número 20 até 99 e imprima-o na tela por extenso.
- Faça um programa que solicite a data de nascimento (dd/mm/aaaa) do usuário e imprima a data com o nome do mês por extenso.

Ex: Entrada: Data de Nascimento: 29/10/1973

Saída: Você nasceu em 29 de Outubro de 1973.

- Faça um programa que leia 2 strings e informe o conteúdo delas seguido do seu comprimento. Informe também se as duas strings possuem o mesmo comprimento e são iguais ou diferentes no conteúdo.

Fonte: PythonBrasil

Iteração com strings

- while

```
A = 'Eu amo strings'
```

```
i=0
```

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

```
    print(i, A[i])
```

```
    i+=1
```

- for

```
for i in A:
```

```
    print(i)
```

- Função enumerate: for <variável>, <variável> in enumerate("<string>"):

```
for i, j in enumerate(A):
```

```
    print( i, j)
```

Exercícios

- Faça um programa que leia um número de telefone, e corrija o número no caso deste conter somente 7 dígitos, acrescentando o '3' na frente. O usuário pode informar o número com ou sem o traço separador.
- Dado uma string com uma frase informada pelo usuário (incluindo espaços em branco), conte:
 1. quantos espaços em branco existem na frase.
 2. quantas vezes aparecem as vogais a, e, i, o, u.

Fonte: PythonBrasil

Dicionários