Strings

Carlos Tavares

25 de Novembro de 2024

Strings

Strings i

Temos lidado com a noção de texto desde o início do curso:

```
» x = "algo"

» x = input ("Escreva algo")

» if x == "qualquer coisa":
        ....
```

Textos são formalmente conhecidos como strings, ou cadeia de caracteres, e há uma ampla gama de funções específicas para lidar com strings.

Strings ii

O que é uma **string**?

É uma coleção de caracteres.

O que é um **caractere**?

Uma "letra"que pode ser usada para compor texto em Python, por exemplo: a, A, z, 0, 9, +, -, %.

Uma lista mais extensa: todos os caracteres [a-z], [A-Z], $\{*, -, +, =, \&, ?, !, , (,), [,], ...\}$

Strings iii

Quantos caracteres existem? Depende do encoding escolhido!

Encoding	Number of Characters	Size (in Bits)
ASCII	128	7 bits
Extended ASCII	256	8 bits
UTF-8	Mais de 1.1 milhões	8 a 32 bits
UTF-16	Mais de 1.1 milhões	16 a 32 bits
UTF-32	Mais de 1.1 milhões	32 bits

Nestas tabelas estão contidos todos os caracteres existentes em todas as línguas do mundo, including emojis. O tipo de codificação pode variar consoante a eficiência necessária.

Strings iv

Tabela ascii

Dec	Char	Dec	Char	Dec	Char	Dec	Char
0	NUL	16	DLE	32	(space)	48	0
1	SOH	17	DC1	33	!	49	1
2	STX	18	DC2	34	"	50	2
3	ETX	19	DC3	35	#	51	3
4	EOT	20	DC4	36	$	52	4
5	ENQ	21	NAK	37	%	53	5
6	ACK	22	SYN	38	&	54	6
7	BEL	23	ETB	39	’	55	7
8	BS	24	CAN	40	(	56	8
9	TAB	25	EM	41	)	57	9
10	LF	26	SUB	42	*	58	:
11	VT	27	ESC	43	+	59	;
12	FF	28	FS	44	,	60	<
13	CR	29	GS	45	-	61	=
14	SO	30	RS	46	.	62	>
15	SI	31	US	47	/	63	?
64	@	80	P	96	‘	112	p
65	A	81	Q	97	a	113	q
66	B	82	R	98	b	114	r
67	C	83	S	99	c	115	s
68	D	84	T	100	d	116	t
69	E	85	U	101	e	117	u
70	F	86	V	102	f	118	v
71	G	87	W	103	g	119	w
72	H	88	X	104	h	120	x
73	I	89	Y	105	i	121	y
74	J	90	Z	106	j	122	z
75	K	91	[	107	k	123	{
76	L	92	\	108	l	124	|
77	M	93	]	109	m	125	}
78	N	94	^	110	n	126	
79	O	95	_	111	o	127	DEL

Strings v

Como mapear caracteres em numeros e caracteres?

Obter o numero na tabela de um dado caracter:
ord (caracter)

Obter o caracter a partir do seu numero na tabela:
chr (index)

Exemplos de emojis: chr (129376), chr (0x1F34E)

Strings vi

Uma string multilinha (codificada com três aspas ou apóstrofos)

» x = "Isto é uma string
multilinha"

Caracteres especiais

\n, \t

O python utiliza unicode por defeito, i.e. podem-se escrever palavras acentuadas e todo o tipo de caracteres:

s = "Esta é a \n aula de programação"

Pode-se também escrever um caracter em unicode directamente

print("Voce acabou de perder um milhao de euros \U0001F600")

Como lidar com strings em Python i

As Strings comportam-se como tuplos em Python: permitem *acesso aleatório* e são imutáveis

» s = "Aula de Aeroespacial"

É possível aceder às strings por índice: **s [índice]**

Exemplo:
» s [0]
'A'
» s [0] ='B'
...
TypeError: 'str' object does not support item assignment

Como lidar com strings em Python ii

É possível acessar strings usando intervalos **s [i:j]**

» s [4:9]
"espaço"

» s [1:]
?

» s [:3]
?

Como lidar com strings em Python iii

O operador in: substring faz parte da string.

» 'A' **in** s
True
» ' Aeroespacial' **in** s
True
» 'Água' **in** s
False

Strings são iteráveis:

```
for i in s:
    print (i)
```

Operações com strings i

É possível construir strings a partir de muitos outros objetos através do construtor **str ()**

Permite a construção de strings a partir de qualquer **tipo de número:**

» s = str (12.3)

» s

"12.3"

Permite a construção a partir de um **tuplo**:

» t = (12, "Exp", 14)

» s = str (t)

» s

"(12, "Exp", 14)"

Operações com strings ii

Operações com strings iii

Permite a construção a partir de uma **lista**:

```
» l = [1, 2, 3, 14]
» s = str (l)
» s
"[1, 2, 3, 14]"
```

A função print usa internamente este construtor para imprimir objetos. O objeto pode ser impresso desde que o Python saiba como construir uma string a partir dele.

Operações com strings iv

Concatenando strings: o operador "+"

» s =
» s = s + "Experiência 1"
» s = s + "Experiência 2"
» print (s)
Experiência 1;Experiência 2

Também é possível criar cópias de strings
» s *= 2
» s
’Experiência 1;Experiência 2Experiência 1;Experiência 2’

Exercícios

Exercício 1: Faça um programa para remover de uma string um conjunto específico de caracteres fornecidos por outra string.

Exercício 2: Faça um programa para duplicar os caracteres de uma string, especificados em outra string.

Exercício 3: Considere uma lista de tuplos com a seguinte forma: (número_de_cópias, extremidade_inferior_do_intervalo, extremidade_superior_do_intervalo). Construa um programa para criar cópias das substrings de uma string, onde o número de cópias e os intervalos são fornecidos pela lista de tuplos.

Funções importantes para strings i

Separar uma string em substrings, usando um separador (função **split**):

variável_string.split (separador)

Exemplo:

»s = "Esta não é uma frase que faça sentido, apenas um exemplo de sentença"
» resultado = s.split ()
» resultado

Funções importantes para strings ii

Criar uma string a partir de substrings, usando uma string como separador (função **join**):

string_variável.join (lista_de_strings)

Exemplo:

» j =
» rejoined = j.join (resultado)
» rejoined

Funções importantes para strings iii

Métodos de encontrar e substituir

O método **find**:

variável_string.find (substr, início, fim)

Exemplo:
»s = "Alguma string com uma palavra interessante no meio"
»índice = s.find ("palavra")
» índice
32

Funções importantes para strings iv

O método **replace**:

variável_string.replace (antigo, novo, [máx_substituições])

Exemplo:
» s = "Algum dia vai ser chuvoso"
» s.replace ("dia", "onde")
"Algum onde vai ser chuvoso"

Funções importantes para strings v

Existem várias funções importantes para manipulação direta de caracteres.

isalpha()	upper ()
isdigit()	swapcase ()
isspace()	lower ()
islower	

Referência:
https://python-reference.readthedocs.io/en/latest/docs/str/

Exercícios

Exercício 1: Faça um programa para contar:

- O número de espaços;
- O número de dígitos;
- O número de caracteres alfanuméricos.

Exercício 2: Faça um programa para verificar se uma palavra é um palíndromo.

Exercicio 3: Faça um programa para implementar uma cifra de substituição.

Perguntas?