



Manipulação de Strings

< thefutureisblue.me />



Strings

➤ Uma string é uma sequência de caracteres simples. Na linguagem Python, as strings são utilizadas com aspas simples ('...') ou aspas duplas ("..."). Para exibir uma string, utiliza-se o comando `print()`.

```
print('Hello, world!')
```



Concatenação de Strings

➤ Para concatenar strings, utiliza-se o operador +.

```
print('Programação' + 'Python')
```

OU

```
titulo = 'Programação'  
linguagem = 'Python'  
titulo_linguagem = titulo + linguagem  
print(titulo_linguagem)
```

Manipulação de Strings

➤ Em Python, existem várias maneiras de manipular strings, como mostra a tabela a seguir:

MÉTODO	DESCRIÇÃO	EXEMPLO
len()	Retorna o tamanho da string.	titulo = "Programação Python" len(titulo) 18
capitalize()	Retorna a string com a primeira letra maiúscula	linguagem = "python" linguagem.capitalize() 'Python'
count()	Informa quantas vezes um caractere (ou uma sequência de caracteres) aparece na string.	linguagem = "Linguagem Python" linguagem.count("y") 1
startswith()	Verifica se uma string inicia com uma determinada sequência.	linguagem = "Python" linguagem.startswith("Py") True



MÉTODO	DESCRIÇÃO	EXEMPLO
endswith()	Verifica se uma string termina com uma determinada sequência.	linguagem = "Python" linguagem.endswith("Py") False
isalnum()	Verifica se a string possui algum conteúdo alfanumérico (letra ou número).	simbolos = "!@#\$%" simbolos.isalnum() False
isalpha()	Verifica se a string possui apenas conteúdo alfabético.	linguagem = "Python" linguagem.isalpha() True
islower()	Verifica se todas as letras de uma string são minúsculas	linguagem = "pytHon" linguagem.islower() False
isupper()	Verifica se todas as letras de uma string são maiúsculas.	linguagem = "# PYTHON 12" linguagem.isupper() True
lower()	Retorna uma cópia da string trocando todas as letras para minúsculo.	linguagem = "#PYTHON 3" linguagem.lower() '#python 3'
upper()	Retorna uma cópia da string trocando todas as letras para maiúsculo.	linguagem = "Python" linguagem.upper() 'PYTHON'

MÉTODO	DESCRIÇÃO	EXEMPLO
swapcase()	Inverte o conteúdo da string (Minúsculo / Maiúsculo).	linguagem = "Python" linguagem.swapcase() 'pYTHON'
title()	Converte para maiúsculo todas as primeiras letras de cada palavra da string.	linguagem = "apostila de python" linguagem.title() 'Apostila De Python'
split()	Transforma a string em uma lista, utilizando os espaços como referência.	linguagem = "cana de açúcar" linguagem.split() ['cana', 'de', 'açúcar']
replace(S1, S2)	Substitui na string o trecho S1 pelo trecho S2.	linguagem = "Apostila Antiga" linguagem.replace("Antiga", "Python") 'Apostila Python'
find()	Retorna o índice da primeira ocorrência de um determinado caractere na string. Se o caractere não estiver na string retorna -1.	linguagem = "Python" linguagem.find("h") 3
ljust()	Ajusta a string para um tamanho mínimo, acrescentando espaços à direita se necessário.	linguagem = " Python" linguagem.ljust(15) ' Python '
center()	Ajusta a string para um tamanho mínimo, acrescentando espaços à esquerda e à direita, se necessário.	linguagem = "Python" linguagem.center(10) ' Python '
strip()	Remove todos os espaços em branco da string.	linguagem = " Python " linguagem.strip() 'Python'

Fatiamento de Strings

- O fatiamento é uma ferramenta usada para extrair apenas uma parte dos elementos de uma string.

Nome_String [Limite_Inferior : Limite_Superior]

- Retorna uma string com os elementos das posições do limite inferior até o limite superior - 1.

Exemplo:

`s = "Python"`

`s[1:4] ⇒` seleciona os elementos das posições 1,2,3
'yth'

`s[2:] ⇒` seleciona os elementos a partir da posição 2
'thon'

`s[:4] ⇒` seleciona os elementos até a posição 3
'Pyth'





**Botando para
rodar!**

Por hoje é só! Obrigado! =)

Até a próxima aula.