

Cartilha Strings em Python

Hashtag Programação

```
l i r a @ g m a i l . c o m  
0 1 2 3 4 5 6 7 8 9 10 11 12 13
```

Métodos de String

Não decore os métodos, os que você for mais usando com o tempo você vai decorar o que precisar.

Mas a dica é: use essa lista para consulta e busque entender como os métodos funcionam e suas aplicações, para poder consultar e usar quando precisar.

Métodos do Python que funcionam em string:

- Sinal de Mais (+) -> Serve para concatenar strings

```
Uso:  
usuario = 'lira'  
servidor = 'gmail.com'  
print(usuario + '@' + servidor)  
Resultado:  
'lira@gmail.com'
```

- len() -> Calcula o tamanho do texto (quantidade de caracteres)

```
Uso:  
cpf = '15389731821'  
print('O cpf tem' + len(cpf) + 'digitos.')  
Resultado:  
'O cpf tem 11 digitos'
```

Métodos Específicos de String:

- capitalize() -> Coloca a 1ª letra Maiúscula

```
Uso:  
texto = 'lira'  
print(texto.capitalize())  
Resultado:  
'Lira'
```

- `casefold()` -> Transforma todas as letras em minúsculas (existe `lower()` mas o `casefold` é melhor normalmente)

Uso:

```
texto = 'Lira'
print(texto.casefold())
```

Resultado:

```
'lira'
```

- `count()` -> Quantidade de vezes que um valor aparece na string

Uso:

```
texto = 'lira@yahoo.com.br'
print(texto.count('.'))
```

Resultado:

```
2
```

- `endswith()` -> Verifica se o texto termina com um valor específico e dá como resposta `True` ou `False`

Uso:

```
texto = 'lira@gmail.com'
print(texto.endswith('gmail.com'))
```

Resultado:

```
True
```

- `find()` -> Procura um texto dentro de outro texto e dá como resposta a posição do texto encontrado

Uso:

```
texto = 'lira@gmail.com'
print(texto.find('@'))
```

Resultado:

```
4
```

Obs: lembrando como funciona a posição nas strings, então o @ está na posição 4

```
l i r a @ g m a i l . c o m
0 1 2 3 4 5 6 7 8 9 10 11 12 13
```

- `format()` -> Formata uma string de acordo com os valores passados. Já usamos bastante ao longo do programa.

Uso:

```
faturamento = 1000
print('O faturamento da loja foi de {} reais'.format(faturamento))
```

Resultado:

```
'O faturamento da loja foi de 1000 reais'
```

- `isalnum()` -> Verifica se um texto é todo feito com caracteres alfanuméricos (letras e números) -> letras com acento ou ç são considerados letras para essa função.

Uso:

```
texto = 'João123'  
print(texto.isalnum())
```

Resultado:

True

Obs: se o texto fosse 'Jo~ao' ou então 'Joao#' o resultado seria False

- `isalpha()` -> Verifica se um texto é todo feito de letras.

Uso:

```
texto = 'João'  
print(texto.isalpha())
```

Resultado:

True

Obs: nesse caso se o texto fosse 'Joao123' o resultado seria False, porque 123 não são letras.

- `isnumeric()` -> Verifica se um texto é todo feito por números.

Uso:

```
texto = '123'  
print(texto.isnumeric())
```

Resultado:

True

Obs: existem os métodos `isdigit()` e `isdecimal()` que tem variações pontuais em caracteres especiais tipo textos com frações e potências, mas para 99% dos casos eles não vão ser necessários.

- `replace()` -> Substitui um texto por um outro texto em uma string.

Uso:

```
texto = '1000.00'  
print(texto.replace('.', ','))
```

Resultado

'1000,00'

Obs: o `replace` precisa de 2 argumentos para funcionar. O 1º é o texto que você quer trocar. O 2º é o texto que você quer colocar no lugar daquele texto que você está tirando.

- `split()` -> Separa uma string de acordo com um delimitador em vários textos diferentes.

Uso:

```
texto = 'lira@gmail.com'
print(texto.split('@'))
Resultado:
['lira', 'gmail.com']
```

- `splitlines()` -> separa um texto em vários textos de acordo com os "enters" do texto

```
Uso:
texto = '''Olá, bom dia
Venho por meio desse e-mail lhe informar o faturamento da loja no dia de hoje.
Faturamento = R$2.500,00
'''
print(texto.splitlines())
Resultado:
['Olá, bom dia', 'Venho por meio desse e-mail lhe informar o faturamento da
loja no dia de hoje.', 'Faturamento = R$2.500,00']
```

- `startswith()` -> Verifica se a string começa com determinado texto

```
Uso:
texto = 'BEB123453'
print(texto.startswith('BEB'))
Resultado:
True
```

- `strip()` -> Retira caracteres indesejados dos textos. Por padrão, retira espaços "extras" no início e no final

```
Uso:
texto = ' BEB123453 '
print(texto.strip())
Resultado:
'BEB123453'
```

- `title()` -> Coloca a 1ª letra de cada palavra em maiúscula

```
Uso:
texto = 'joão paulo lira'
print(texto.title())
Resultado:
'João Paulo Lira'
```

- `upper()` -> Coloca o texto todo em letra maiúscula

Uso:

```
texto = 'beb12343'  
print(texto.upper())  
Resultado:  
'BEB12343'
```