Manipulação de Strings

INSTITUTO FEDERAL
Paraíba
Campus João Pessoa

String - definição

- Nas linguagens de programação existem alguns tipos de dados denominados de tipos simples ou primitivos, por exemplo: inteiro, real, caracter, lógico.

- Por outro lado, existem tipos formados a partir de outros tipos primitivos denominados tipos de dados compostos ou estruturados. Strings e listas são tipos compostos.
  - Strings podem ser definidos como sequencias de caracteres.

Delimitação

- Strings podem ser delimitadas entre aspas simples ('...') ou aspas duplas ("...") com o mesmo resultado.

```
>>> 'manipulando strings em python'
'manipulando strings em python'
>>> "manipulando strings em python"
'manipulando strings em python'
```

- Use aspas simples se a string contem aspas duplas ou vice-versa.

```
>>> 'manipulando "strings" em python'
'manipulando "strings" em python'
>>> "marca d'água"
"marca d'água"
```

Quebra de linha (\n)

- Use "\n" para a quebra de linha em uma string.

```
>>> s = 'primeira linha\nsegunda linha'
>>> s
'primeira linha\nsegunda linha'
>>> print(s)
primeira linha
segunda linha
```

- Se for necessário que o "\" não seja interpretado como caracter de controle, use um "r" antes da string.

```
>>> print('C:\nome')
C:
ome
>>> print(r'C:\nome')
C:\nome
```

Múltiplas linhas

- Uma string pode abranger várias linhas. Uma maneira é usar aspas triplas: ''' ... ''' ou """..."""".

```
>>> print('''
primeira linha
segunda linha
terceira linha
''')

primeira linha
segunda linha
terceira linha
```

Múltiplas linhas

- Pode-se evitar a primeira linha em branco adicionando um "\".

```
>>> print('''\
primeira linha
segunda linha
terceira linha
''')
primeira linha
segunda linha
terceira linha
```

Indexação

- Cada caracter que compõe a string pode ser referenciado por um índice
- Se o índice for um número negativo a sequência é contada a partir da direita, começando em -1.

```
>>> s = 'python'
>>> s
'python'
>>> s[-1]
'n'
>>> s[-6]
'p'
```

Fatiamento

- O fatiamento (slice) é uma ferramenta usada para extrair apenas uma parte da string.
- Sintaxe:

Nome_String [Limite_inferior : Limite_Superior : Incremento]

```
>>> s = 'Linguagem Python'
>>> s
'Linguagem Python'
>>> s[1:4]
'ing'
>>> s[:4]
'Ling'
>>> s[4:]
'uagem Python'
```

```
>>> s[1:4:2]
'ig'
>>> s[1::2]
'igae yhn'
>>>
>>> s[::2]
'LnugmPto'
>>> s[::-2]
'nhy eagi'
```

Concatenação e repetição

- Strings podem ser concatenadas com o operador + e repetidas com *

```
>>> s1 = 'ab'
>>> s1
'ab'
>>> s2 = 'xy'
>>> s2
'xy'
>>> s3 = 3 * s1 + s2
>>> s3
'abababxy'
```

Contagem

- len() – retorna o tamanho da string

```
>>> s = 'Aula de Python'
>>> len(s)
14
```

- count() – retorna o número de ocorrências de uma sequência de caracteres dentro da string

```
>>> s = 'aula de python'
>>> s
'aula de python'
>>> s.count('a')
2
>>> s.count('py')
1
>>> s.count('abc')
0
```

Conversão de maiúsculas/minúsculas

- lower() – retorna a string toda em letras minúsculas
- upper() – retorna a string toda em letras maiúsculas
- swapcase() – retorna a string invertendo minúsculas/maiúsculas

```
>>> s = 'Aula de Python'
>>> s.lower()
'aula de python'
>>> s.upper()
'AULA DE PYTHON'
>>> s.swapcase()
'aULA DE pYTHON'
```

Iniciais maiúsculas

- capitalize() – retorna a string com a primeira letra em maiúsculo
- title() – retorna a string com a primeira letra de cada palavra em maiúsculo

```
>>> s = 'aula de python'
>>> s
'aula de python'
>>> s.capitalize()
'Aula de python'
>>> s.title()
'Aula De Python'
```

Busca

- Find() – retorna o índice da primeira ocorrência de uma determinada sequência de caracteres em uma string

```
>>> s = 'aula de python'
>>> s
'aula de python'
>>> s.find('a')
0
>>> s.find(' ')
4
>>> s.find('py')
8
>>> s.find('abc')
-1
```

Substituição

- Replace(s1,s2) – retorna uma cópia da string substituindo o trecho s1 pelo trecho s2

```
>>> s = 'aula de python'
>>> s1 = 'python'
>>> s2 = 'java'
>>> s.replace(s1,s2)
'aula de java'
```

Divisão

- split() – transforma a string em uma lista, utilizando os espaços em branco como referência.

```
>>> frase = 'Aula de Python'
>>> frase
'Aula de Python'
>>> divisao = frase.split()
>>> frase
'Aula de Python'
>>> divisao
['Aula', 'de', 'Python']
>>> divisao[1]
'de'
```

Remoção de brancos

- strip() – remove todos os espaços em branco da string
- lstrip() - remove todos os espaços em branco do lado esquerdo da string
- rstrip() - remove todos os espaços em branco do lado direito da string

```
>>> s = '    aula de python    '
>>> s
'    aula de python    '
>>> s.strip()
'aula de python'
>>> s.lstrip()
'aula de python    '
>>> s.rstrip()
'    aula de python'
```

Testando os valores de uma String

Métodos	Descrição
isalnum()	Verifica se a string contem apenas caracteres alfanuméricos
isalpha()	Verifica se a string contem apenas caracteres alfabéticos
islower()	Verifica se a string contem apenas caracteres minúsculos
isnumeric()	Verifica se a string contem apenas caracteres numéricos
isspace()	Verifica se a string contem espaços em branco
istitle	Verifica se a string está como título (letras iniciais maiúsculas)
isupper	Verifica se a string contem apenas caracteres maiúsculos

Percorrendo strings

```
frase='Aula de Python'
tam=len(frase)
for i in range(tam):
    print(frase[i],' ',end='')
```

OU

```
frase='Aula de Python'
for i in frase:
    print(i,' ',end='')
```

Saída:

```
A u l a   d e   P y t h o n
```