

Lógica de programação e algoritmo

AULA 06

Nessa Aula

- Strings
 - Exemplos
 - Exercícios

Strings

- Variáveis do tipo string armazenam cadeias de caracteres como nomes e textos em geral.
- Strings são imutáveis em Python.
- O conteúdo de variáveis string podem ser concatenados com o operador de adição (+). Exemplo: "AB" + "C" é igual a "ABC".

Trabalhando com strings

- **Exemplo 01:** Alteração de uma string.

```
# -*- coding: UTF-8 -*-
```

```
S = "Alô Mundo"
```

```
print(S[0])
```

```
S[0] = "a"
```

TypeError: 'str' object does not support item assignment

Trabalhando com strings

- **Exemplo 02:** Convertendo uma string em lista.

```
# -*- coding: UTF-8 -*-  
L = list("Alô Mundo")  
print(L)  
L[0] = "a"  
print(L)  
s = "".join(L)  
print(s)
```

- **list:** função que transforma cada caractere da string em um elemento da lista.
- **join:** método que transforma os elementos da lista em string.

Verificação parcial de strings

- **Exemplo 03:** Verificação parcial de strings.

```
# -*- coding: UTF-8 -*-
```

```
nome = "João da Silva"  
print(nome.startswith("João"))  
print(nome.startswith("joão"))  
print(nome.endswith("Silva"))
```

- **startswith:** método que verifica os primeiros caracteres de uma string.
- **endswith:** método que verifica os últimos caracteres de uma string.

Verificação parcial de strings

- **Exemplo 04:** Conversão em maiúsculas e minúsculas.

```
# -*- coding: UTF-8 -*-
```

```
s = "O Rato roeu a roupa do Rei de Roma"  
print(s.lower())  
print(s.upper())  
print(s.lower().startswith("o rato"))  
print(s.upper().startswith("O RATO"))
```

- **lower:** método que retorna uma cópia da string com todos os caracteres minúsculos.
- **upper:** método que retorna uma cópia da string com todos os caracteres maiúsculos.

Verificação parcial de strings

- **Exemplo 05:** Pesquisa de palavras em uma string usando **in**.

```
# -*- coding: UTF-8 -*-
```

```
s = "Maria Amélia Souza"
print("Amélia" in s)
print("Maria" in s)
print("Souza" in s)
print("a A" in s)
print("amélia" in s)
```

- **in:** operador que verifica se uma string está contida em outra.

Verificação parcial de strings

- **Exemplo 06:** Pesquisa de palavras em uma string usando **not in**.

```
# -*- coding: UTF-8 -*-
```

```
s = "Todos os caminhos levam a Roma"  
print("levam" not in s)  
print("Caminhos" not in s)  
print("AS" not in s)
```

- **not in:** operador que verifica se uma string não está contida em outra.

Contagem

- **Exemplo 07:** Contagem de letras e palavras.

```
# -*- coding: UTF-8 -*-
```

```
t = "um tigre, dois tigres, três tigres"  
print(t.count("tigre"))  
print(t.count("tigres"))  
print(t.count("t"))  
print(t.count("z"))
```

- **count:** método que conta as ocorrências de uma letra ou palavra em uma string.

Pesquisa de strings

- **Exemplo 08:** Pesquisa de strings com **find**.

```
# -*- coding: UTF-8 -*-
```

```
s = "Alô mundo"
```

```
print(s.find("mun"))
```

```
print(s.find("ok"))
```

- **find:** método que pesquisa se uma string está dentro de outra e obtém a posição da primeira ocorrência.

Substituição de strings

- **Exemplo 09:** Substituição de strings.

```
# -*- coding: UTF-8 -*-
```

```
s = "um tigre, dois tigres, três tigres"  
print(s.replace("tigre", "gato"))
```

- **replace:** método que substitui trechos de uma string por outros.

Remoção de espaços em branco

- **Exemplo 10:** Remoção de espaços em branco.

```
# -*- coding: UTF-8 -*-
```

```
t = "    Olá    "
```

```
print(t.strip())
```

- **strip:** método que remove espaços em branco do início ou fim da string.

Exercícios

1) Escreva um programa que leia duas strings. Verifique se a segunda ocorre dentro da primeira e imprima a posição de início.

1ª string: AABBEFAATT

2ª string: BE

Resultado: BE encontrado na posição 3 de AABBEFAATT

Exercícios

2) Escreva um programa que leia duas strings e gere uma terceira com os caracteres comuns às duas strings lidas.

1ª string: AAACCTBF

2ª string: CBT

Resultado: CTB

A ordem dos caracteres da string gerada não é importante, mas deve conter todas as letras comuns a ambas.

Exercícios

3) Escreva um programa que leia duas strings e gere uma terceira apenas com os caracteres que aparecem em uma delas.

1ª string: CTA

2ª string: ABC

Resultado: TB

A ordem dos caracteres da terceira string não é importante.

Exercícios

4) Escreva um programa que leia uma string e imprima quantas vezes cada caractere aparece nessa string.

String: TTAAC

Resultado:

T: 2x

A: 2x

C: 1x

Exercícios

5) Escreva um programa que leia duas strings e gere uma terceira, na qual os caracteres da segunda foram retirados da primeira.

1ª string: AATTGGAA

2ª string: TG

3ª string: AAAA

Exercícios

6) Escreva um programa que leia três strings. Imprima o resultado da substituição na primeira, dos caracteres da segunda pelos da terceira.

1ª string: AATTGGAA

2ª string: TG

3ª string: AC

Resultado: AAAACCAA

Referências

- MENEZES, N. N. C. **Introdução à Programação com Python**: algoritmos e lógica de programação para iniciantes. 2ª ed. São Paulo: Novatec, 2014.
- PYTONSOFTWAREFOUNDATION. **Download the latest version for Windows**: Python 3.6.4. Disponível em: <https://www.python.org/downloads/>. Acesso em: 21 fev. 2018.
- Em homenagem à Janaine Arantes

Referências

Bibliografia Complementar

- MENEZES, N. N. C. **Introdução à Programação com Python**: algoritmos e lógica de programação para iniciantes. 2ª ed. São Paulo: Novatec, 2014.
- SWEIGART, Al. **Automatize Tarefas Maçantes com Python** - Programação Prática para Verdadeiros Iniciantes. São Paulo: Novatec, 2015.
- BORGES, L. E. **Python para Desenvolvedores**. 3ª ed. São Paulo: Novatec, 2014.