

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

AULA 11 – MANIPULANDO STRINGS

WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR

```
RegisterController.php
You, 7 months ago | 1 author (You)
import VueRouter from "vue-router";
import routes from "../routes/routes";
import store from "../store/index";
import vuexI18n from "vuex-i18n";
import enLangFile from "../lang/en";

// Set config file into the global variable
window.config = require("../vue.config");

// Import bootstrap file
require("../bootstrap");

// Set the globally
const Vue = require("vue");

// Import the router management
const Router = Router;

// Import the store management
const Store = Store;

// Import the i18n management
const I18n = I18n;

// Import the lang management
const Lang = Lang;

// Import the lang management
const Lang = Lang;

// Import the lang management
const Lang = Lang;
```

Calendário de Aulas

- ~~13/11/2025 – Matrizes e Funções~~
- ~~20/11/2025 – Feriado~~
- 27/11/2025 – Manipulando Strings e Arquivos TXT.
- 04/12/2025 – 3ª Avaliação
- **10/12/2025 – Apresentação do Fórum Integrador (Quarta)**
- 11/12/2025 – Avaliação Final

Aula de Hoje

- Manipular strings
- Verificação parcial de strings
- Contagem
- Pesquisa de strings
- Posicionamento de strings
- Substituição de strings
- Verificação de dados
- Definir Precisão de Valores Reais

Trabalhando com Strings em Python

Manipulação de Strings

- A manipulação de strings em Python envolve a utilização de várias técnicas para **manipular e modificar cadeias de caracteres**.
- Uma string é uma sequência de caracteres, como palavras, frases ou até mesmo textos completos.
- A manipulação de strings é uma tarefa comum na programação, pois permite a manipulação e transformação de dados textuais.

Em Python, as strings são tratadas como objetos

- Portanto, possuem vários métodos embutidos que facilitam sua manipulação.
- Alguns exemplos de operações comuns de manipulação de strings em Python incluem: concatenação (junção de duas ou mais strings), extração de partes de uma string, substituição de trechos de texto, conversão para letras maiúsculas ou minúsculas, entre outras.

Onde utilizamos manipulação de strings?

1. **Processamento de texto:** Em muitas aplicações, como análise de dados, web scraping, processamento de linguagem natural e mineração de texto, é necessário manipular strings para extrair informações relevantes ou transformar os dados em um formato adequado para análise.
2. **Desenvolvimento de software:** A manipulação de strings é uma habilidade fundamental para desenvolvedores de software. Ela é usada para criar interfaces de usuário, lidar com entradas de usuário, formatação de dados, geração de relatórios, criação de URLs, entre outras tarefas.
3. **Desenvolvimento web:** No desenvolvimento web, a manipulação de strings é usada para lidar com URLs, rotas, manipulação de dados de formulários, criação de consultas para bancos de dados, geração de templates dinâmicos, entre outros aspectos relacionados à manipulação de texto em um contexto web.
4. **Automação de tarefas:** Ao automatizar tarefas em Python, muitas vezes é necessário lidar com strings para analisar informações em arquivos de texto, renomear arquivos, realizar buscas e substituições em texto, entre outras operações.

Verificação Parcial de Strings

Verificação Parcial de Strings

Verificando se uma string **começa** com um conjunto de caracteres:

```
main.py
1 nome = 'Walter Travassos'
2
3 if nome.startswith('Walter'):
4     print('0 nome começa com Walter')
5
6 elif nome.startswith('Travassos'):
7     print('0 nome começa com Travassos')
```

```
>_ Console x Shell x
0 nome começa com Walter
>
```

Verificação Parcial de Strings

Verificando se uma string **termina** com um conjunto de caracteres:


```
main.py
1 nome = 'Walter Travassos'
2
3 if nome.endswith('Walter'):
4     print('0 nome termina com Walter')
5
6 elif nome.endswith('Travassos'):
7     print('0 nome termina com Travassos')
```

```
>_ Console v x Shell x +
0 nome termina com Travassos
>
```

Verificação Parcial de Strings

Formatando uma string para **caixa baixa** (minúscula):

```
main.py  
1 nome = 'WALTER TRAVASSOS'  
2 print(nome)  
3  
4 nome_minusculo = nome.lower()  
5 print(nome_minusculo)
```


```
>_ Console v x   
WALTER TRAVASSOS  
walter travassos  
?
```

Verificação Parcial de Strings

Formatando uma string para **caixa alta** (maiúscula):

 main.py

```
1 nome = 'walter travassos'  
2 print(nome)  
3  
4 nome_minusculo = nome.upper()  
5 print(nome_minusculo)
```

```
>_ Console v x   
walter travassos  
WALTER TRAVASSOS  
>
```

Verificação Parcial de Strings

Verificar se uma **palavra consta em uma string**:

```
main.py
1 nome = 'Walter Travassos Sarinho'
2
3 if 'Travassos' in nome:
4     print('Existe o sobrenome
    Travassos no nome em questão')
5
6 else:
7     print('Não existe o sobrenome
    Travassos no nome em questão')
```

```
>_ Console v x Shell x +
Existe o sobrenome Travassos no nome
em questão
>
```

Verificação Parcial de Strings

Verificar se uma **palavra não consta em uma string**:

```
main.py
1 nome = 'Walter Travassos Sarinho'
2
3 if 'Silva' not in nome:
4     print('Não existe o sobrenome Silva no nome em
   questão')
5
6 else:
7     print('Existe o sobrenome Silva no nome em questão')
8
```

```
>_ Console x Shell +
Não existe o sobrenome Silva no
nome em questão
>
```

Exercício – parte 1

- Receba um nome em Pascal Case numa variável, por exemplo: “Walter Travassos”.
- Em seguida transforme em caixa baixa e apresente o nome original e o nome em caixa baixa.

Contagem

Contagem

Contagem de letras e palavras:

```
main.py
1 frase = 'Assim pensando o tempo passa e a gente vai ficando
  pra trás. Esperando, esperando, esperando. Esperando o sol.
  Esperando o trem. Esperando o aumento. Desde o ano passado
  para o mês que vem. (Chico Buarque)'
2
3 conta_palavra = frase.count('Esperando')
4 print('Quantidade de esperando: ', conta_palavra)
5
6 conta_sinal = frase.count('.')
7 print('Quantidade de pontos: ', conta_sinal)
```

```
>_ Console v x Shell +
Quantidade de esperando: 4
Quantidade de pontos: 6
>
```

Contagem

Quantidade total de caracteres de uma string:

```
main.py  
1 palavra = 'Otorrinolaringologista'  
2 conta_letras = len(palavra)  
3 print('Quantidade de letras: ', conta_letras)
```

```
>_ Console v x Shell +  
Quantidade de letras: 22  
>
```

Exercício – parte 2

- Receba um nome em Pascal Case numa variável, por exemplo: “Walter Travassos”.
- Em seguida transforme em caixa baixa e apresente o nome original e o nome em caixa baixa.
- **Apresente a quantidade de caracteres do nome inserido.**

Pesquisa de Strings

Pesquisa de Strings

Saber se uma **string está dentro de outra** e obter o índice (posição) do primeiro caractere:

```
main.py
1 frase = 'café com leite'
2 posicao = frase.find('leite')
3
4 if posicao != -1:
5     print(posicao)
6 else:
7     print('Palavra não localizada!')
```

```
>_ Console v x
9
>
```

Exercício – parte 3

- Receba um nome em Pascal Case numa variável, por exemplo: “Walter Travassos”.
- Em seguida transforme em caixa baixa e apresente o nome original e o nome em caixa baixa (lower).
- Apresente a quantidade de caracteres do nome inserido (len).
- Apresente a posição do primeiro caractere de espaço (find).

Posicionamento de Strings

Posicionamento de Strings

Exemplo – Centralizar strings:

```
main.py  
1 palavra = 'PYTHON'  
2 print('|||', palavra.center(40), '|||')  
3  
4 frase = 'Changing the way you code'  
5 print('|||', frase.center(40, '.'), '|||')  
6
```

```
>_ Console ▾ × 🐚 Shell × +  
|||          PYTHON          |||  
||| .....Changing the way you code..... |||  
>
```

Posicionamento de Strings

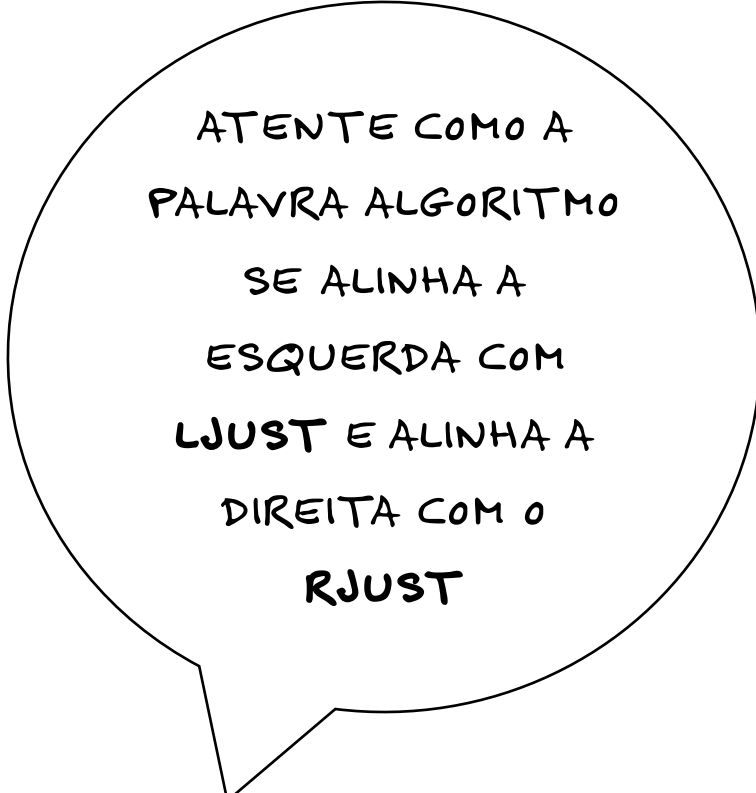
Preencher strings com espaços:

```
1 palavra = "Python"  
2 outrapalavra = "Algoritmo"  
3 print(palavra, outrapalavra, palavra)  
4 print(palavra, outrapalavra.ljust(20), palavra)  
5 print(palavra, outrapalavra.rjust(20), palavra)
```

```
Python Algoritmo Python  
Python Algoritmo Python  
Python Algoritmo Python
```

`ljust` -> Alinha o texto pra esquerda, deixando espaço à direita.

`rjust` -> Alinha o texto pra direita, deixando espaço à esquerda.



ATENTE COMO A
PALAVRA ALGORITMO
SE ALINHA A
ESQUERDA COM
LJUST E ALINHA A
DIREITA COM O
RJUST

Exercício – parte 4

- Receba um nome em Pascal Case numa variável, por exemplo: “Walter Travassos”.
- Em seguida transforme em caixa baixa e apresente o nome original e o nome em caixa baixa (lower).
- Apresente a quantidade de caracteres do nome inserido (len).
- Apresente a posição do primeiro caractere de espaço (find).
- Agora, centralize o nome com 40 caracteres em caixa alta. Coloque o caractere de * entre o nome (center).

Substituição de Strings

Substituição de Strings

Substituir um conjunto de caracteres de uma string por outro:

```
main.py
1 frase = 'Chá combina com tudo'
2 correto = frase.replace('Chá', 'Café')
3 print(correto)
4
```

```
>_ Console x Shell x
Café combina com tudo
>
```

Exercício – parte 5

- Receba um nome em Pascal Case numa variável, por exemplo: “Walter Travassos”.
- Em seguida transforme em caixa baixa e apresente o nome original e o nome em caixa baixa (lower).
- Apresente a quantidade de caracteres do nome inserido (len).
- Apresente a posição do primeiro caractere de espaço (find).
- Agora, centralize o nome com 40 caracteres em caixa alta. Coloque o caractere de * entre o nome (center).
- **Troque o caractere de espaço do nome completo por cerquilha (replace).**

Verificação de Dados

Verificação de Dados

Verificar o **tipo** de dado contido em um string:

```
main.py
1  dado = 'www123'
2
3  if dado.isdigit():
4      print('Existem apenas números')
5  elif dado.isalpha():
6      print('Existe apenas letras')
7  elif dado.isalnum():
8      print('Existe letras ou números')
```

```
>_ Console v x Shell x +
Existe letras ou números
>
```

Exercício – parte 6

- Receba um nome em Pascal Case numa variável, por exemplo: “Walter Travassos”.
- Em seguida transforme em caixa baixa e apresente o nome original e o nome em caixa baixa (lower).
- Apresente a quantidade de caracteres do nome inserido (len).
- Apresente a posição do primeiro caractere de espaço (find).
- Agora, centralize o nome com 40 caracteres em caixa alta. Coloque o caractere de * entre o nome (center).
- Troque o caractere de espaço do nome completo por cerquilha (replace).
- **Receba do usuário uma senha e, em seguida, verifique se ela possui letras ou números (isalnum).**

Definir Precisão de Valores Reais

Definir Precisão

Definir a **quantidade** de casas decimais após o ponto de um float:

```
main.py
1  numero = 43.13985765
2
3  print('Valor = %.2f' % numero)
4
5  print('Valor = %.3f' % numero)
6
7  print('Valor = %.4f' % numero)
```

```
>_ Console v x 9
Valor = 43.14
Valor = 43.140
Valor = 43.1399
?
```

Recortando uma string

- Em Python, você pode recortar ou extrair partes específicas de uma string usando a sintaxe de fatiamento (**slicing**).
- O **slicing** permite selecionar uma parte contígua da string com base em índices ou posições.
- A sintaxe básica do **slicing** em Python é:

`string [início : fim : passo]`

Por exemplo...

```
texto = "Olá, Mundo!"
```

```
# Recortar a palavra "Mundo"
```

```
recorte = texto[5:10]
```

```
print(recorte) # Saída: "Mundo"
```

Exercício – parte 7

- Receba um nome em Pascal Case numa variável, por exemplo: “Walter Travassos”.
- Em seguida transforme em caixa baixa e apresente o nome original e o nome em caixa baixa (lower).
- Apresente a quantidade de caracteres do nome inserido (len).
- Apresente a posição do primeiro caractere de espaço (find).
- Agora, centralize o nome com 40 caracteres em caixa alta. Coloque o caractere de * entre o nome (center).
- Troque o caractere de espaço do nome completo por cerquilha (replace).
- Receba do usuário uma senha e, em seguida, verifique se ela possui letras e números (isalnum).
- **Recorte o primeiro nome da string e apresente no console.**

Exercícios

Exercício 1

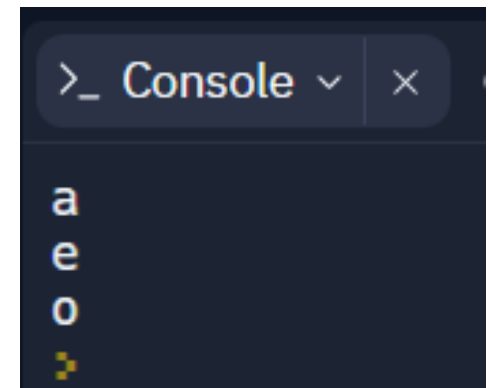
Faça um programa que receba uma palavra e verifica se a mesma tem vogais, imprimindo as vogais localizadas na palavra na tela.

O programa não conta as vogais, apenas verifica quais existem na palavra escolhida, e mostra essas vogais.

Dica: crie uma lista com as vogais.

Por exemplo: `vogais = ['a', 'e', 'i', 'o', 'u']`

Se a palavra for "Walter Travassos", o resultado vai ser:



```
>_ Console v x  
a  
e  
o  
>
```

Exercício 1 - Resposta

```
main.py
1 nome = 'Walter Travassos'
2 vogais = ['a', 'e', 'i', 'o', 'u']
3
4 for vogal in vogais:
5     if vogal in nome.lower():
6         print(vogal)
7
```

```
>_ Console v x
a
e
o
>
```

Exercício 2

Faça um programa que receba uma frase de um usuário e verifica se na mesma existe alguma palavra de baixo calão (escolha uma).

Só imprima a frase que o usuário digitou se a mesma não possuir a palavra de baixo calão escolhida.

Caso exista a palavra de baixo calão, exiba uma mensagem de erro.

Exercício 2 - Resposta

```
main.py × funcoes.py × +
main.py
1 frase = 'Sport campeão de 87'
2
3 if 'flamengo' in frase.lower():
4     print('Palavrões são proibidos')
5 else:
6     print(frase)
7
```

```
>_ Console × Shell
Sport campeão de 87
>
```

Exercício 3

Faça um programa que permita ao usuário verificar se uma senha a ser cadastrada em um sistema é válida ou não. O sistema em questão tem os seguintes critérios para validar uma senha:

- Deve ter 6 caracteres (nem mais nem menos);
- Deve ter letras e números;
- Não deve constar as palavras 'FLA', 'MENGO' ou 'MENGAO'
- Não deve começar com a letra 'A' nem terminar com a letra 'F'

Exercício 3 - Resposta

```
1  senha = "9spo12"
2  senha = senha.lower()
3  valida = False
4
5  if len(senha) == 6:
6      if not senha.isdigit() and not senha.isalpha():
7          if 'fla' not in senha and 'mengo' not in senha and 'mengao' not in senha:
8              if not senha.startswith('a') and not senha.endswith('f'):
9                  valida = True
10
11 if(valida):
12     print('Senha válida')
13 else:
14     print('Senha inválida')
```

```
>_ Console v x +
Senha válida
>
```

Explore o assunto!

Referência

Menezes, N. N. C. Introdução à Programação com Python. Editora Novatech (2ª Edição).

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

AULA 11 – MANIPULANDO STRINGS

WALTER TRAVASSOS SARINHO

@PROF.WALTERTRAVASSOS

WALTER.TRAVASSOS@SECTRAS.EDU.BR

```
RegisterController.php
You, 7 months ago | 1 author (You)
import VueRouter from "vue-router";
import routes from "../routes/routes";
import store from "../store/index";
import vuexI18n from "vuex-i18n";
import enLangFile from "../lang/en";

// Set config file into the global variable
window.config = require("../vue.config");

// Import bootstrap file
require("../bootstrap");

// Set the globally
const Vue = require("vue");

// Import the router management
const Router = Router;

// Application
const app = new Vue({
  el: "#app",
  router: Router,
  store: store,
  vuexI18n: vuexI18n({
    locales: ["pt", "en"],
    defaultLocale: "pt",
    fallbackLocale: "pt",
    messages: {}
  })
});
```