

ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

AULA 12 – MANIPULANDO ARQUIVOS TXT

WALTER TRAVASSOS SARINHO

@WALTEROPROFESSOR

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 vuex-i18n
const vuexI18n = vuexI18n;

// Import the enLangFile
const enLangFile = enLangFile;

// Create the Vue instance
const app = new Vue({
  el: "#app",
  router: Router,
  store: Store,
  vuexI18n: vuexI18n,
  enLangFile: enLangFile
}).$mount("#app");
```

Calendário de Aulas

- ~~20/05/2025 – Matrizes e Funções~~
- ~~27/05/2025 – Manipulando Strings~~
- 03/06/2025 – Manipulando Arquivos TXT e Revisão
- 10/06/2025 – 3ª Avaliação
- 16/06/2025 – Avaliação Final
- 17/06/2025 – Apresentação do Fórum Integrador

Aula de Hoje

- Leitura e escrita em arquivos txt

Leitura e Escrita de Arquivos em Python

Leitura e Escrita de Arquivos

- Variáveis são voláteis (os dados são perdidos após o encerramento da aplicação).
- Arquivos podem armazenar dados processados por aplicações, mantendo-os mesmo após o encerramento dos mesmos.
- Python fornece funções para a manipulação de arquivos, tanto para a leitura quanto para a escrita.

Acessando (e/ou criando) arquivos em Python

Função `open()`

Acessando arquivos para escrita

- Utilizamos a função `open()` para acessar um arquivo em Python.
- Sintaxe de acesso para **escrita**:

```
arquivo = open('nome.txt', 'w');
```

- Se o arquivo não existir, será criado um novo arquivo com o nome e extensão indicados na função.
- Se o arquivo já existir, **seu conteúdo será apagado e substituído pelo novo conteúdo a ser adicionado.**

Acessando arquivos para escrita

- Sintaxe de acesso para escrita (2ª forma):

```
arquivo = open('nome.txt', 'a');
```

- Se o arquivo não existir, será criado um novo arquivo com o nome e extensão indicados na função.
- Se o arquivo já existir, **os dados a serem adicionados serão acrescentados aos já existentes.**

Acessando arquivos

- Sintaxe de acesso para leitura:

```
arquivo = open('nome.txt', 'r');
```

- Se o arquivo não existir, retorna uma mensagem de erro.
- Se o arquivo já existir, fornecerá dados para leitura.

Escrevendo dados em arquivos

Função `write()`

E depois de escrever, função `close()` para gravar.

Escrevendo dados em arquivos

- Utilizamos a função `write()` para escrever dados em um arquivo. Exemplo:

```
arquivo = open('dados.txt', 'w')  
nome = 'Walter Travassos'  
arquivo.write(nome)  
arquivo.close()
```

- Após a inserção dos dados, a função `close()` deverá ser utilizada para que a gravação seja realizada com sucesso.

Lendo arquivos

Lendo arquivos

Python fornece 3 funções para a leitura de dados em arquivos:

- `read()`: Retorna **todo o conteúdo** de um arquivo.
- `readline()`: Retorna **uma linha do arquivo** por vez.
- `readlines()`: Retorna **uma lista**, onde cada elemento é uma linha do arquivo.

Lendo arquivos com a função `read()`

```
arquivo = open('dados.txt', 'r')
dados = arquivo.read()

print(dados)
arquivo.close()
```

Lendo arquivos com a função `readline()`

```
arquivo = open('dados.txt', 'r')  
linha1 = arquivo.readline()  
linha2 = arquivo.readline()  
  
print(linha1)  
print(linha2)  
arquivo.close()
```

Lendo arquivos com a função `readlines()`

```
arquivo = open("dados.txt", "r")  
linhas = arquivo.readlines()
```

```
for linha in linhas:  
    print(linha)
```

```
arquivo.close()
```

Exercícios

Exercício 1

Faça um programa para ler no console o nome de 3 pessoas e, em seguida, salvar esses nomes no arquivo **peessoas.txt**.

Em seguida, o programa deverá exibir o nome das 3 pessoas que foram salvas no arquivo **peessoas.txt**.

Exercício 1 - Resposta

```
1  #escrita
2  arquivo = open("pessoas.txt", "w")
3  for i in range(3):
4      pessoa = input("Digite o nome: \n")
5      pessoa = pessoa + "\n"
6      arquivo.write(pessoa)
7  arquivo.close()
8
9  #leitura
10 arquivo = open("pessoas.txt", "r")
11 linhas = arquivo.read()
12 print(linhas)
13 arquivo.close()
```

Exercício 2

Faça um programa que leia o nome de 3 pessoas, cadastrando-as em um arquivo.

Em seguida, o programa deverá exibir apenas o nome da segunda pessoa cadastrada.

Exercício 2 - Resposta

```
main.py
1  arquivo = open('dados.txt', 'a')
2
3  for n in range (3):
4      nome = input('Digite um nome: ')
5      nome+='\n'
6      arquivo.write(nome)
7
8  arquivo = open('dados.txt', 'r')
9  linhas = arquivo.readlines()
10 print(linhas[1])
11 arquivo.close()
```

Exercício 3

Agenda telefônica (Cadastro) - Faça um programa que:

1. Receba o nome e o telefone de 3 pessoas.
 2. Receba o nome do arquivo a ser criado.
 3. Crie um arquivo txt (com o nome do arquivo indicado pelo usuário) para cada pessoa e escreva o nome e o telefone no mesmo.
- O nome deverá ficar na primeira linha do arquivo.
 - O telefone deverá ficar na segunda linha do arquivo.

Exercício 3 - Resposta

```
1  for n in range (3):
2      nome = input('Digite o nome: ')
3      nome+='\n'
4      fone= input('Digite o telefone: ')
5      fone+='\n'
6      nomearquivo = input('Digite o nome do arquivo: ')
7      nomearquivo += '.txt'
8      arquivo = open(nomearquivo, 'w')
9      arquivo.write(nome)
10     arquivo.write(fone)
11     arquivo.close()
```

Exercício 4

Agenda telefônica (Listagem) - Faça um programa que:

- Receba o nome do arquivo a ser lido.
- Leia o arquivo indicado pelo usuário (a partir do nome do arquivo).
- Exiba o nome e o telefone cadastrados no mesmo.

Exercício 4 - Resposta

```
1 nomearquivo = input('Digite o nome do arquivo: ')
2 nomearquivo += '.txt'
3 arquivo = open(nomearquivo, 'r')
4 linhas = arquivo.read()
5 print(linhas)
6 arquivo.close()
```

Exercício 5

Sistema Acadêmico (Cadastro) - Faça um programa que:

1. Receba o nome e 3 notas de 4 alunos.
 2. Receba o nome do arquivo a ser criado.
 3. Crie um arquivo txt (com o nome do arquivo indicado pelo usuário) para cada aluno e escreva as informações digitadas pelo usuário no mesmo.
- O nome deverá ficar na primeira linha do arquivo.
 - A primeira nota deverá ficar na segunda linha do arquivo, a segunda na terceira linha e a terceira na quarta.

Exercício 5 - Resposta

```
1  for n in range (4):
2      nome = input('Digite o nome: ')
3      nome+='\n'
4      n1 = input('Digite a primeira nota: ')
5      n1 +='\n'
6      n2 = input('Digite a segunda nota: ')
7      n2 +='\n'
8      n3 = input('Digite a terceira nota: ')
9      n3 +='\n'
10     nomearquivo = input('Digite o nome do arquivo: ')
11     nomearquivo += '.txt'
12     arquivo = open(nomearquivo, 'w')
13     arquivo.write(nome)
14     arquivo.write(n1)
15     arquivo.write(n2)
16     arquivo.write(n3)
17     arquivo.close()
```

Exercício 6

Sistema acadêmico (Cálculo de Média) - Faça um programa que:

- Receba o nome do arquivo a ser lido.
- Leia o arquivo indicado pelo usuário (a partir do nome do arquivo).
- Exiba o nome do aluno, sua média no semestre e se o mesmo está aprovado ou reprovado (considere que a média da faculdade é 7,0).

Exercício 6 - Resposta

```
1 nomearquivo = input('Digite o nome do arquivo: ')
2 nomearquivo += '.txt'
3 arquivo = open(nomearquivo, 'r')
4 linhas = arquivo.readlines()
5 n1 = float(linhas[1])
6 n2 = float(linhas[2])
7 n3 = float(linhas[3])
8 media = (n1+n2+n3) / 3
9 print('nome: ', linhas[0])
10 print('média: ', media)
11 √ if media >= 7:
12     print("Aprovado")
13 √ else:
14     print("Reprovado")
15 arquivo.close()
```

Exercício 7

Faça um programa que receba dois arquivos do usuário, e crie um terceiro arquivo com o conteúdo dos dois primeiros juntos (o conteúdo do primeiro seguido do conteúdo do segundo).

Exercício 7 - Resposta

```
1  arquivo1 = open('aluno1.txt', 'r')
2  texto1 = arquivo1.read()
3
4  arquivo2 = open('aluno2.txt', 'r')
5  texto2 = arquivo2.read()
6
7  arquivo3 = open('alunos', 'w')
8  arquivo3.write(texto1)
9  arquivo3.write(texto2)
10 arquivo3.close()
```

Exercício 8

Faça um programa que leia um arquivo contendo o nome e o preço de diversos produtos (separados por linha), e calcule o total da compra.

Exemplo de arquivo

```
Café  
9  
Arroz  
6  
Feijão  
10  
Açúcar  
5
```

Exercício 8 - Resposta

```
1  arquivo = open('produtos.txt', 'r')
2  linhas = arquivo.readlines()
3  total = 0
4
5  for i in range(1, 9, 2):
6      total += int(linhas[i])
7  arquivo.close()
8
9  print('O total é ', total)
```

Quizlet de Revisão

<https://quizlet.com/br/810460722/python-flash-cards/?i=14uvey&x=1jqt>



ALGORITMOS E LÓGICA DE PROGRAMAÇÃO

AULA 12 – MANIPULANDO ARQUIVOS TXT

WALTER TRAVASSOS SARINHO

@WALTEROPROFESSOR

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 Vuex from "vuex";
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 VueRouter
const Router = require("vue-router");

// Import the Vuex
const Vuex = require("vuex");

// Import the enLangFile
const enLangFile = require("../lang/en");

// Create the Vue instance
const app = new Vue({
  el: "#app",
  router: Router,
  store: store,
  components: {
    HelloWorld,
  },
  data: {
    name: "Walter Travassos Sarinho"
  },
  methods: {
    // ...
  },
});
```