

Manipulando Arquivos em Python

Banco de Dados

Charles Tim Batista Garrocho

Instituto Federal do Paraná – IFPR
Campus Goioerê

charles.garrocho.com/BD

charles.garrocho@ifpr.edu.br

Técnico em Informática



INSTITUTO FEDERAL

Em Python, devemos **abrir** (open) arquivos antes de usá-los e **fechar** (close) os arquivos depois de que tivermos terminado de utilizá-los.

Como você pode imaginar, depois de aberto um arquivo passa a ser um **objeto** Python de maneira semelhante que outros dados.

Tabela 1 mostra os métodos que podem ser usados para abrir e fechar arquivos.



Métodos Para Abrir e Fechar Arquivos

Nome do Método	Uso	Explicação
open	<code>open(n_arquivo, 'r')</code>	Abre um arquivo chamado <code>n_arquivo</code> e o usa para leitura. Retorna uma referência para um objeto file.
open	<code>open(n_arquivo, 'w')</code>	Abre um arquivo chamado <code>n_arquivo</code> e o usa para escrita. Retorna uma referência para um objeto file.
close	<code>ref_arquivo.close()</code>	Utilização do arquivo referenciado pela variável <code>ref_arquivo</code> terminou.



Encontrando um Arquivo em Disco

Abrir um arquivo requer que você, como programador, e o Python entre em acordo sobre onde está o arquivo no seu disco.

A maneira que os arquivos são localizados no disco e através do seu **caminho** (path). Você pode pensar em `n_arquivo` como o primeiro nome do arquivo e o caminho como o nome completo.

Por exemplo em um **Linux**, o caminho poderia ser `/home/fulano/dados.txt`

Já no **Windows** o caminho poderia ser `C:/Users/fulano/dados.txt`.



Escrevendo Texto em um Arquivo

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
arq = open('/tmp/lista.txt', 'w')
texto = """
Lista de Alunos
---
João da Silva
José Lima
Maria das Dores
"""
arq.write(texto)
arq.close()
```

Na linha 3 é aberto o arquivo `/tmp/lista.txt` em modo de gravação, caso o arquivo não exista ele será criado. Na quarta linha é definida a variável **texto** que, na linha 11, é incluída ao arquivo `/tmp/lista.txt`, que corresponde à variável **arq**. Na linha 12 o arquivo é fechado e então o programa termina.



Escrevendo Listas em um Arquivo

Também é possível escrever em arquivos usando listas no lugar de variáveis apenas em texto:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
arq = open('/tmp/lista.txt', 'w')
texto = []
texto.append('Lista de Alunos\n')
texto.append('---\n')
texto.append('João da Silva\n')
texto.append('José Lima\n')
texto.append('Maria das Dores')
arq.writelines(texto)
arq.close()
```

Isto vai gerar o mesmo arquivo que o exemplo anterior, a diferença é que deu mais trabalho, mas dependendo da situação este modo pode ser melhor.



Lendo Texto de um Arquivo

Lendo informações de um arquivo:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
arq = open('/tmp/lista.txt', 'r')
texto = arq.read()
print(texto)
arq.close()
```

read: Método que lê e retorna o conteúdo inteiro de um arquivo em um string. É frequentemente usado em um comando de atribuição de tal forma que a variável seja uma referência para um string com o conteúdo do arquivo:

```
string_arquivo = ref_arquivo.read()
```



Escrevendo Listas em um Arquivo

Lendo informações de um arquivo em forma de lista:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
arq = open('/tmp/lista.txt', 'r')
texto = arq.readlines()
for linha in texto :
    print(linha)
arq.close()
```

readlines: Método que lê e retorna o conteúdo de um arquivo como uma lista de strings. Cada linha do arquivo é representado por um string da lista:

```
lista_de_linhas = ref_arquivo.readlines()
```



Exercícios

Faça um programa gerenciador de coleção de livros. Ele deve salvar e recuperar as informações dos livros de um arquivo. Você deve salvar um título e um autor por livro.

O menu deve conter as seguintes operações:

- a) adicionar livro
- b) remover livro
- c) pesquisar livro por título
- d) pesquisar livros por autor
- e) listar todos livros

