



PYTHON
ACADEMY

COMO ABRIR ARQUIVOS UTILIZANDO PYTHON

Guia de arquivos: `open()`, `read()`, `write()`, `close()`, `with`, modos (`r/w/a/rb`), `pathlib`, context managers, casos práticos (ler TXT, escrever logs, copiar arquivos). Neste ebook, vamos explorar como realizar operações de abertura, leitura e escrita de forma eficiente e eficaz em Python.

PYTHONACADEMY.COM.BR

Este ebook foi gerado por



Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs

Deixe que nossa IA faça o trabalho pesado



Syntax Highlight



Adicione Banners Promocionais



Edite em Markdown em Tempo Real



Infográficos feitos por IA

TESTE AGORA

PRIMEIRO CAPÍTULO 100% GRÁTIS

Salve salve Pythonista 🙌

Um dos recursos mais importantes em qualquer linguagem de programação é a capacidade de trabalhar com arquivos.

Em Python, o processo de abrir, ler e escrever em arquivos é bastante simples e intuitivo.

Neste artigo, vamos explorar como realizar essas operações de forma eficiente e eficaz em Python.

Por que abrir arquivos em Python?

A capacidade de abrir e manipular arquivos é essencial para muitos projetos de programação.

Os arquivos podem conter uma variedade de informações, como texto, dados estruturados, imagens, entre outros.

Ao abrir e ler esses arquivos, podemos realizar diversas tarefas, como extrair informações, processar dados e até mesmo criar novos arquivos.

Além disso, ao escrever em arquivos, podemos armazenar dados de saída, resultados de cálculos e até mesmo criar registros para fins de auditoria.

Portanto, é fundamental saber como abrir, ler e escrever arquivos em Python para realizar projetos práticos e úteis.

Abrindo um arquivo em Python

O primeiro passo para trabalhar com um arquivo em Python é abrí-lo.

Para isso, utilizamos a função `open()`, que recebe dois parâmetros: o nome do arquivo a ser aberto e o modo de abertura do arquivo.

Vejamos um exemplo:

```
arquivo = open('texto.txt', 'r')
```

Neste exemplo, estamos abrindo o arquivo chamado “texto.txt” em modo de leitura (`'r'`). Existem diferentes modos de abertura de arquivos que podem ser usados:

- `'r'`: modo de leitura, o arquivo deve existir previamente
- `'w'`: modo de escrita, se o arquivo não existir, ele será criado
- `'a'`: modo de anexar, adiciona informações ao final do arquivo
- `'x'`: modo exclusivo, cria um novo arquivo somente se ele não existir
- `'b'`: modo binário, usado para arquivos binários, como imagens ou vídeos
- `'t'`: modo de texto, usado para arquivos de texto

Podemos usar combinações desses modos.

Por exemplo, para abrir um arquivo para leitura e escrita, podemos utilizar `'r+'`, e para abrir um arquivo em modo binário de leitura, podemos usar `'rb'`.

Lendo um arquivo em Python

Agora que temos o arquivo aberto, podemos lê-lo.

Para isso, utilizamos o método `read()`, que retorna todo o conteúdo do arquivo como uma string.

Veja o exemplo abaixo:

```
arquivo = open('texto.txt', 'r')

conteudo = arquivo.read()

print(conteudo)
```

Neste exemplo, estamos lendo o conteúdo do arquivo aberto na variável `arquivo` e armazenando-o na variável `conteudo`. Em seguida, exibimos o conteúdo na tela.

Também é possível ler o arquivo linha por linha utilizando o método `readline()`.

Este método retorna uma única linha do arquivo por vez. Veja um exemplo:

```
arquivo = open('texto.txt', 'r')

linha = arquivo.readline()

print(linha)
```

Neste exemplo, a cada chamada do método `readline()`, uma nova linha do arquivo é lida e armazenada na variável `linha`.

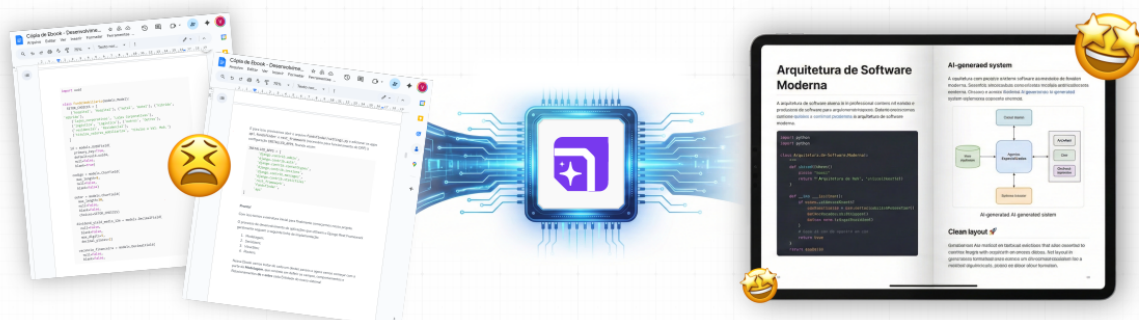
Em seguida, a linha é exibida na tela.



*Estou construindo o **DevBook**, uma plataforma que usa IA para criar ebooks técnicos — com código formatado e exportação em PDF. Te convido a conhecer!*

Crie Ebooks técnicos incríveis em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs

Deixe que nossa IA faça o trabalho pesado

Syntax Highlight

Adicione Banners Promocionais

Edite em Markdown em Tempo Real

Infográficos feitos por IA

TESTE AGORA! PRIMEIRO CAPÍTULO 100% GRÁTIS

Escrevendo em um arquivo em Python

Além de ler arquivos, também é possível escrever informações em um arquivo usando Python.

Para isso, utilizamos o método `write()`. Vejamos um exemplo:

```
arquivo = open('novo_texto.txt', 'w')
arquivo.write('Olá, Mundo!')
arquivo.close()
```

Neste exemplo, criamos um novo arquivo chamado “novo_texto.txt” e escrevemos a string “Olá, Mundo!” no arquivo.

Em seguida, fechamos o arquivo utilizando o método `close()`.

É importante fechar o arquivo depois de terminar de escrever para liberar os recursos do sistema operacional.

Manipulando arquivos com o statement `with`

Além de abrir e fechar arquivos manualmente, Python também oferece uma sintaxe mais conveniente para manipulação de arquivos.

O statement `with` é usado para abrir um arquivo e garantir que ele seja fechado corretamente, mesmo em caso de erro.

Vejamos um exemplo:

```
with open('texto.txt', 'r') as arquivo:
    conteudo = arquivo.read()
    print(conteudo)
```

Neste exemplo, o arquivo é aberto dentro do bloco `with` e o conteúdo é lido e armazenado na variável `conteudo`.

Ao final do bloco `with`, o arquivo é automaticamente fechado, mesmo que ocorra uma exceção durante a leitura.

O statement `with` é especialmente útil quando manipulamos arquivos grandes ou quando há várias operações a serem realizadas no arquivo.

Ele garante que os recursos do sistema operacional sejam liberados de forma eficiente.

Conclusão

Neste artigo, aprendemos como abrir, ler e escrever arquivos em Python.

Vimos que o processo é bastante simples e intuitivo, utilizando a função `open()` para abrir o arquivo, os métodos `read()` e `readline()` para ler o conteúdo e o método `write()` para escrever informações no arquivo.

Também vimos que podemos usar o statement `with` para garantir o fechamento adequado do arquivo.

A manipulação de arquivos em Python é uma habilidade fundamental para qualquer programador, e é especialmente importante em projetos que envolvam processamento de dados, gerenciamento de informações ou criação de registros.

Com o conhecimento adquirido neste artigo, você estará pronto para utilizar arquivos em seus projetos Python de forma eficiente e eficaz.

Nos vemos na próxima, Pythonista! 🙌

Não se esqueça de conferir!



DevBook

Crie Ebooks técnicos em minutos com IA

Conheça a 1ª IA Especializada na criação de Ebooks **com código!**



Chega de formatar código no Google Docs



 Syntax Highlight

 Infográficos feitos por IA

 Adicione Banners Promocionais

Deixe que nossa IA faça o trabalho pesado

 Edite em Markdown em Tempo Real

TESTE AGORA 

 PRIMEIRO CAPÍTULO 100% GRÁTIS