**Arquivos em Python**

O Python possui algumas funções prontas para manipular arquivos binários puros (onde, conhecendo a estrutura interna de qualquer formato, podemos salvar qualquer tipo de arquivo) e para manipular arquivos de texto (onde os binários são decodificados como *strings*).

Focaremos no básico de manipulação de arquivo de texto, pois, na prática, quando formos trabalhar com arquivos mais complexos, é provável que usaremos bibliotecas específicas para lidar com eles, e elas já terão funções próprias para ler e salvar esses arquivos da maneira correta.

**Abrindo e fechando arquivos**

Podemos criar arquivos novos ou abrir arquivos já existentes utilizando a função *open*. Ela possui 2 argumentos: o caminho do arquivo e o modo de operação.

| **Modo** | **Símbolo** | **Descrição** |
| --- | --- | --- |
| *read* | r | lê um arquivo existente |
| *write* | w | cria um novo arquivo |
| *append* | a | abre um arquivo existente para adicionar informações ao seu final |
| *update* | + | ao ser combinado com outros modos, permite alteração de arquivo já existente (ex: r+ abre um arquivo existente e permite modificá-lo) |

Após abrirmos (ou criarmos) um arquivo, podemos realizar diversas operações. Ao final de todas elas, devemos **fechar** o nosso arquivo usando a função *close*. Essa etapa é importante por 2 motivos:

1. Se alteramos o arquivo mas não o fechamos, as alterações não serão salvas;
2. Se esquecemos de fechar um arquivo, outros programas podem ter problemas ao acessá-lo.

**Roteiro básico**

Vamos seguir os seguintes passos para manipular nossos arquivos:

1. Abrir ou criar um arquivo:

arquivocriado = open("criado.txt", "w")

A linha de comando acima abre (ou cria se não existe) um arquivo chamado "criado.txt" para escrita ("w", de write) e guarda na variável "arquivocriado" as informações para manipulá-lo.

arquivolido = open("teste.txt", "r")

A linha acima lê ("r", de read) um arquivo já existente chamado "teste.txt" e guarda na variável "arquivolido" as informações para manipulá-lo.

1. Carregar os dados do arquivo (leitura)

dados = arquivolido.read()

print(dados)

A função read() retorna todo o conteúdo do arquivo como uma string.

Precisamos carregar o conteúdo do arquivo em algum formato que sabemos trabalhar. A read() carrega o conteúdo de um arquivo de texto em uma string.

1. Manipular os dados do arquivo (escrita)

arquivocriado.write("linha 1")

arquivocriado.write("linha 2")

arquivocriado.write("linha 3")

Em casos mais complexos, iremos manipular o conteudo LIDO no passo anterior para posteriormente reescrevê-lo. Em outros mais simples, podemos escrever diretamente no arquivo.

1. Fechar o arquivo

arquivocriado.close()

arquivolido.close()

Essa etapa é muito importante para garantir a integridade dos novos dados no arquivo. As modificações são salvas somente ao fechar o arquivo.

**Comando with**

Um jeito mais inteligente de se trabalhar com arquivos é utilizar a sintaxe do "with". Ele garante que após a finalização do bloco, o arquivo será fechado.

with open('teste.txt', 'r') as arquivolido:

dados = arquivolido.read()

print(dados)

É possível ler o arquivo linha a linha, como no exemplo:

with open('teste.txt', 'r') as arquivolido:

linha = arquivolido.readline()

while linha != '':

print(linha, end='')

linha = arquivolido.readline()

# OU

with open('teste.txt', 'r') as arquivolido:

for linha in arquivolido:

print(linha, end='')

O mesmo pode ser feito para escrever no arquivo:

with open('teste.txt', 'r') as arquivolido:

with open('copiateste.txt', 'w') as arquivocriado:

for linha in arquivolido:

arquivocriado.write(linha)

No comando acima, as linhas do arquivo "teste.txt" são copiadas e salvas no arquivo "copiateste.txt".