

## Mãos à obra!

Já sabemos como ler arquivos no Python, mas como podemos escrever em um arquivo?

Antes de tudo, para não modificar o arquivo original, vamos fazer uma cópia deste arquivo e chamá-la de `contatos-escrita.csv`. Para testar os comandos de escrita, vamos também criar um arquivo Python chamado de `testa_escrita.py`.

Da mesma forma que para ler um arquivo é necessário abri-lo antes, para escrever o arquivo também faremos uso da função `open()`, só que desta vez passaremos o modo de abertura do arquivo como escrita (`w`, *write*). Portanto, no arquivo `testa_escrita.py`:

```
arquivo_contatos = open('dados/contatos-escrita.csv', encoding='latin_1', mode='a+')
```

Vamos criar um contato e pedir para a variável `arquivo_contatos` escrever (`write`) este novo contato no arquivo:

```
arquivo_contatos = open('dados/contatos-escrita.csv', encoding='latin_1', mode='w')

contato = '11,Carol,carol@carol.com.br\n'

arquivo_contatos.write(contato)
```

Bacana! Quando pedimos para o Python executar esse arquivo, notamos que todo o conteúdo dele sumiu. Apenas o contato que pedimos para o Python inserir está lá. Isso acontece por causa da maneira que o modo `w` funciona.

Vamos copiar o conteúdo do arquivo `contatos.csv` para `contatos-escrita.csv` e trocar o modo de abertura para `a`:

```
arquivo_contatos = open('dados/contatos-escrita.csv', encoding='latin_1', mode='a')

contato = '11,Carol,carol@carol.com.br\n'

arquivo_contatos.write(contato)
```

Rodando o arquivo `testa_escrita.py` vemos que dessa vez o arquivo `contatos-escrita.csv` não foi apagado, e o contato que pedimos para ser inserido foi inserido ao final dele. Isso acontece por conta do comportamento do modo `a` que não apaga o arquivo e insere o que pedimos ao final do arquivo.

Vamos agora realizar outro teste. Vamos inserir diversos contatos no arquivo. Para isso, criamos uma lista com os contatos e a iteramos, percorrendo cada contato e escrevendo no arquivo. Como vamos realizar várias operações de escrita, alteramos o modo de abertura do arquivo para `w`:

```
arquivo_contatos = open('dados/contatos-escrita.csv', encoding='latin_1', mode='a')

contatos = ['11,Carol,carol@carol.com.br\n',
            '12,Ana,ana@ana.com.br\n',
            '13,Tais,tais@tais.com.br\n',
```

```
'14,Felipe,felipe@felipe.com.br\n']
```

```
for contato in contatos:
    arquivo_contatos.write(contato)
```

Vamos colocar uma função `input()` para parar a execução do programa e ver o que acontece:

```
arquivo_contatos = open('dados/contatos-escrita.csv', encoding='latin_1', mode='a')

contatos = ['11,Carol,carol@carol.com.br\n',
            '12,Ana,ana@ana.com.br\n',
            '13,Tais,tais@tais.com.br\n',
            '14,Felipe,felipe@felipe.com.br\n']

for contato in contatos:
    arquivo_contatos.write(contato)

input('Pressione <Enter> para encerrar o programa')
```

Se olharmos o arquivo `contatos-escrita.csv`, veremos que os dados ainda não foram inseridos. Isso acontece porque os arquivos, por padrão, só são escritos quando fazemos a liberação do seu recurso, ou seja, quando fechamos o arquivo. Se invocarmos o método `close()` do objeto `arquivo_contatos` antes da função `input()`, veremos que o arquivo foi escrito. Além disso, podemos forçar a inserção de um arquivo sem que ele precise ser fechado, utilizando o método `flush()`:

```
arquivo_contatos = open('dados/contatos-escrita.csv', encoding='latin_1', mode='a')

contatos = ['11,Carol,carol@carol.com.br\n',
            '12,Ana,ana@ana.com.br\n',
            '13,Tais,tais@tais.com.br\n',
            '14,Felipe,felipe@felipe.com.br\n']

for contato in contatos:
    arquivo_contatos.write(contato)

arquivo_contatos.flush()
```

Além disso, podemos ir percorrendo o arquivo através do método `seek()`. Com esse método, podemos passar uma posição para o arquivo e trocar o conteúdo de uma linha, por exemplo, passando os caracteres para maiúsculo:

```
arquivo_contatos = open('dados/contatos-escrita.csv', encoding='latin_1', mode='w')

contatos = ['11,Carol,carol@carol.com.br\n',
            '12,Ana,ana@ana.com.br\n',
            '13,Tais,tais@tais.com.br\n',
            '14,Felipe,felipe@felipe.com.br\n']

for contato in contatos:
    arquivo_contatos.write(contato)

arquivo_contatos.flush()
```

```
arquivo_contatos.seek(28)
arquivo_contatos.write('12,Ana,ana@ana.com.br\n'.upper())
arquivo_contatos.flush()
```

Veja que a linha do arquivo foi alterada. Podemos, então, percorrer todo o arquivo e visualizar essas alterações. Basta passar a posição 0 para o método `seek()` e iterar sobre o arquivo. Para isso, antes, temos que colocar o modo do arquivo como atualização ( + ):

```
arquivo_contatos = open('dados/contatos-escrita.csv', encoding='latin_1', mode='a+')

contatos = ['11,Carol,carol@carol.com.br\n',
            '12,Ana,ana@ana.com.br\n',
            '13,Tais,tais@tais.com.br\n',
            '14,Felipe,felipe@felipe.com.br\n']

for contato in contatos:
    arquivo_contatos.write(contato)

arquivo_contatos.flush()

arquivo_contatos.seek(28)
arquivo_contatos.write('12,Ana,ana@ana.com.br\n'.upper())
arquivo_contatos.flush()
arquivo_contatos.seek(0)

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