

Trabalhando com arquivos em Python

Fabio Lubacheski
fabio.lubacheski@mackenzie.br

Trabalhando com arquivos em Python

- Durante a execução de um programa, seus dados ficam na memória. Quando o programa termina, ou o computador é desligado, os dados na memória desaparecem.
- Para armazenar os dados permanentemente, você tem que colocá-los em um **arquivo**. Arquivos usualmente são guardados em um disco rígido (HD), num pendrive ou em um CD-ROM.
- Um arquivo é organizado pelo sistema operacional em pasta (=diretório=folder) e tem um nome único (nome+extensão) dentro de uma pasta.

Trabalhando com arquivos em Python

- Trabalhar com arquivos é muito parecido com trabalhar com livros. Para utilizar um livro, você tem que **abrí-lo**. Quando você termina, você tem que **fechá-lo**.
- Enquanto o livro estiver **aberto**, você pode tanto lê-lo quanto escrever nele. Na maioria das vezes, você lê o livro inteiro em sua ordem natural, uma página após a outra, como no arquivo.
- Dentro de um arquivo texto, agrupar as informações do arquivo em linhas.

Manipulação de arquivos em Python

- Para **abrir um arquivo** em Python utilizamos a função `open` que serve para obter uma referência a um objeto do tipo arquivo. O formato geral da função `open` é:
`open(nome_do_arquivo, modo)`
- Para modo temos:
 - '**r**' : abre para leitura, falha se o arquivo não existir.
 - '**w**' : abre para escrita, se o arquivo não existir é criado.
- **Importante:** A função `open()` abre o arquivo para o programa manipular o conteúdo do arquivo, além disso, o arquivo para ser aberto precisa estar na mesma pasta onde se encontra o arquivo fonte com o o seu programa.

Manipulação de arquivos em Python

- Para ler conteúdo do arquivo podemos usar as funções:
 - `read()` : Retorna uma string única com todo o conteúdo do arquivo.
 - `readline()` : Retorna a próxima linha do arquivo, e incrementa a posição atual.
 - `readlines()` : retorna todo o conteúdo do arquivo em uma lista, uma linha do arquivo por elemento da lista.
 - `write(data)` : escreve a string data para o arquivo, na posição atual ou ao final do arquivo, dependendo do modo de abertura. Esta função falha se o arquivo foi aberto com modo ``r'`.
 - `seek(n)` : muda a posição atual do arquivo para o valor indicado em n.
 - `close()` : fecha o arquivo.

Manipulação de arquivos em Python

- A função seguinte, copia um arquivo, o primeiro argumento é o nome do arquivo original (origem) e o segundo é o nome do novo arquivo (destino):

```
def copiaArquivo(origemArquivo, destinoArquivo):  
    f1 = open( origemArquivo, "r")  
    f2 = open( destinoArquivo, "w")  
    while True: #laço infinito  
        texto = f1.readline()  
        # se string vazia, fim de arquivo  
        # finaliza a leitura  
        if texto == "":  
            break  
        f2.write(texto)  
    f1.close()  
    f2.close()
```

exemplosArquivos.py

Fim