

Python - Manipulação de arquivos

Quando utilizamos qualquer aplicativo de computador, alguns dados precisam ser fornecidos. Os dados são armazenados na memória principal do computador (RAM) até que o aplicativo seja executado. Depois disso, o conteúdo da memória RAM é apagado.

Gostaríamos de armazená-lo de forma que possa ser recuperado sempre que necessário em um meio persistente, como um arquivo em disco.

Python usa funções `input()` e `print()` integradas para realizar operações padrão de entrada/saída. O programa Python interage com esses dispositivos IO por meio de objetos de fluxo padrão `stdin` e `stdout` definidos no módulo `sys`.

A função `input()` lê bytes de um dispositivo de fluxo de entrada padrão, ou seja, teclado. Portanto, ambas as instruções a seguir leem a entrada do usuário.

```
name = input()
#is equivalent to
import sys
name = sys.stdin.readline()
```

A função `print()`, por outro lado, envia os dados para o dispositivo de fluxo de saída padrão, ou seja, o monitor. É uma função de conveniência que emula o método `write()` do objeto `stdout`.

```
print (name)

#is equivalent to
import sys
sys.stdout.write(name)
```

Qualquer objeto que interage com o Stream de entrada e saída é chamado de objeto Arquivo. A função interna do Python `open()` retorna um objeto de arquivo.

A função `open()`

Esta função cria um objeto de arquivo, que seria utilizado para chamar outros métodos de suporte associados a ela.

Sintaxe

```
file object = open(file_name [, access_mode][, buffering])
```



Aqui estão os detalhes dos parâmetros -

- **file_name** - O argumento file_name é um valor de string que contém o nome do arquivo que você deseja acessar.
- **access_mode** - O access_mode determina o modo em que o arquivo deve ser aberto, ou seja, leitura, gravação, acréscimo, etc. Uma lista completa de valores possíveis é fornecida abaixo na tabela. Este é um parâmetro opcional e o modo padrão de acesso ao arquivo é leitura (r).
- **buffering** - Se o valor do buffer for definido como 0, nenhum buffer ocorrerá. Se o valor do buffer for 1, o buffer de linha será executado durante o acesso a um arquivo. Se você especificar o valor do buffer como um número inteiro maior que 1, a ação de buffer será executada com o tamanho do buffer indicado. Se for negativo, o tamanho do buffer será o padrão do sistema (comportamento padrão).

Modos de abertura de arquivo

A seguir estão os modos de abertura de arquivo -

Sr. Não.	Modos e descrição
1	R Abre um arquivo somente para leitura. O ponteiro do arquivo é colocado no início do arquivo. Este é o modo padrão.
2	rb Abre um arquivo somente para leitura em formato binário. O ponteiro do arquivo é colocado no início do arquivo. Este é o modo padrão.
3	r+ Abre um arquivo para leitura e gravação. O ponteiro do arquivo colocado no início do arquivo.
4	rb+ Abre um arquivo para leitura e gravação em formato binário. O ponteiro do arquivo colocado no início do arquivo.
5	c Abre um arquivo apenas para gravação. Substitui o arquivo se ele existir. Caso o arquivo não exista, cria um novo arquivo para gravação.
6	b Abre o arquivo em modo binário

7	t Abre o arquivo em modo texto (padrão)
8	+ abrir arquivo para atualização (leitura e escrita)
9	wb Abre um arquivo para gravação apenas em formato binário. Substitui o arquivo se ele existir. Caso o arquivo não exista, cria um novo arquivo para escrita.
10	c+ Abre um arquivo para escrita e leitura. Substitui o arquivo existente se o arquivo existir. Caso o arquivo não exista, cria um novo arquivo para leitura e gravação.
11	wb+ Abre um arquivo para gravação e leitura em formato binário. Substitui o arquivo existente se o arquivo existir. Caso o arquivo não exista, cria um novo arquivo para leitura e gravação.
12	a Abre um arquivo para anexar. O ponteiro do arquivo estará no final do arquivo, se o arquivo existir. Ou seja, o arquivo está no modo de acréscimo. Se o arquivo não existir, cria um novo arquivo para escrita.
13	ab Abre um arquivo para anexar em formato binário. O ponteiro do arquivo estará no final do arquivo, se o arquivo existir. Ou seja, o arquivo está no modo de acréscimo. Se o arquivo não existir, cria um novo arquivo para escrita.
14	um+ Abre um arquivo para anexar e ler. O ponteiro do arquivo estará no final do arquivo, se o arquivo existir. O arquivo é aberto no modo de acréscimo. Caso o arquivo não exista, cria um novo arquivo para leitura e gravação.
15	ab+ Abre um arquivo para anexar e ler em formato binário. O ponteiro do arquivo estará no final do arquivo, se o arquivo existir. O arquivo é aberto no modo de acréscimo. Caso o arquivo não exista, cria um novo arquivo para leitura e gravação.
16	x aberto para criação exclusiva, falhando se o arquivo já existir

Depois que um arquivo for aberto e você tiver um objeto de arquivo, poderá obter várias informações relacionadas a esse arquivo.

Exemplo

```
# Open a file
fo = open("foo.txt", "wb")
print ("Name of the file: ", fo.name)
print ("Closed or not: ", fo.closed)
print ("Opening mode: ", fo.mode)
fo.close()
```

Ele produzirá a seguinte **saída** -

```
Name of the file: foo.txt
Closed or not: False
Opening mode: wb
```