

# Python - Aula

Arquivos

# Python - Arquivos

## 1) OPEN

A função **OPEN** é utilizado para realizar operações com arquivos em Python.

# Python - Arquivos

## 1) OPEN

A função **OPEN** é utilizado para realizar operações com arquivos em Python.

Um arquivo pode ser aberto para leitura (**read**), escrita (**write**) ou acrescentar (**append**) conteúdo.

# Python - Arquivos

## 1) OPEN

A função **OPEN** é utilizado para realizar operações com arquivos em Python.

Um arquivo pode ser aberto para leitura (**read**), escrita (**write**) ou acrescentar (**append**) conteúdo.

Após utilizado, um arquivo **sempre** deve ser fechado.

# Python - Arquivos

## 1) OPEN - **READ**

**Abrir:**

```
1  
2  arquivo = open("meu_arquivo.txt", "r")  
3
```

# Python - Arquivos

## 1) OPEN - READ

**Abrir:**

```
1  
2 arquivo = open("meu_arquivo.txt", "r")  
3
```

**Caso o arquivo não exista:**

```
Traceback (most recent call last):  
  File "/home/alexandre/Documentos/Projetos/f/teste.py", line 2, in <module>  
    arquivo = open("meu_arquivo.txt", "r")  
FileNotFoundError: [Errno 2] No such file or directory: 'meu_arquivo.txt'  
>>> |
```

# Python - Arquivos

## 1) OPEN - READ

**Lendo como String:**

```
1  
2  arquivo = open("meu_arquivo.txt", "r")  
3  conteudo = arquivo.read()  
4
```

# Python - Arquivos

## 1) OPEN - READ

**Lendo como String:**

```
1
2  arquivo = open("meu_arquivo.txt", "r")
3  conteudo = arquivo.read()
4
```

**Lendo como Lista:**

```
1
2  arquivo = open("meu_arquivo.txt", "r")
3  conteudo = arquivo.readlines()
4
```



# Python - Arquivos

## 1) OPEN - READ

**Fechando o arquivo:**

```
1  
2  arquivo = open("meu_arquivo.txt", "r")  
3  arquivo.close()  
4
```

# Python - Arquivos

## 2) OPEN - **WRITE**

**Abrir:**

```
1  
2  arquivo = open("meu_arquivo.txt", "w")  
3
```

# Python - Arquivos

## 2) OPEN - **WRITE**

**Abrir:**

```
1  
2  arquivo = open("meu_arquivo.txt", "w")  
3
```

Se o arquivo não existir, cria.

Se o arquivo existir, substitui.

Um arquivo aberto nesse modo não pode ser lido.

# Python - Arquivos

## 2) OPEN - **WRITE**

**Escrever:**

```
1  
2  arquivo = open("meu_arquivo.txt", "w")  
3  arquivo.write("Um texto qualquer")  
4
```

# Python - Arquivos

## 2) OPEN - **WRITE**

**Escrever:**

```
1
2 arquivo = open("meu_arquivo.txt", "w")
3 arquivo.write("Um texto qualquer")
4
```

**Fechando o arquivo:**

```
1
2 arquivo = open("meu_arquivo.txt", "w")
3 arquivo.close()
4
```

# Python - Arquivos

## #EXERCÍCIOS

- 1) Dado um arquivo com uma lista de IPs, gere um segundo arquivo separando os IPs válidos dos inválidos.

### Entrada:

```
200.135.80.9  
192.168.1.1  
8.35.67.74  
257.32.4.5  
85.345.1.2  
1.2.3.4  
9.8.234.5  
192.168.0.256
```

### Saída:

```
[Endereços válidos:]  
200.135.80.9  
192.168.1.1  
8.35.67.74  
9.8.234.5  
1.2.3.4  
  
[Endereços inválidos:]  
257.32.4.5  
85.345.1.2  
192.168.0.256
```

# Python - Arquivos

## #EXERCÍCIOS

- 2) Uma empresa está tendo problemas de espaço em disco no seu servidor de arquivos. Para tentar resolver este problema, o Administrador de Rede precisa saber qual o espaço em disco ocupado pelas contas dos usuários, e identificar os usuários com maior espaço ocupado. Através de um aplicativo baixado da Internet, ele conseguiu gerar um arquivo chamado “usuarios.txt” que mostra o nome do usuário e o espaço ocupado em bytes. A partir deste arquivo, você deve criar um arquivo chamado relatorio.txt. **Use 1 KB = 1024bytes e 1MB = 1024KB.**

usuarios.txt

```
alexandre 456123789
anderson  1245698456
antonio   123456456
carlos    91257581
cesar     987458
rosemary  789456125
```

# Python - Arquivos

## #EXERCÍCIOS

relatorio.txt

Empresa Inc.		Uso do espaço em disco por usuário	
-----		-----	
Nr.	Usuário	Espaço utilizado	% do uso
1	alexandre	434,99 MB	16,85%
2	anderson	1187,99 MB	46,02%
3	antonio	117,73 MB	4,56%
4	carlos	87,03 MB	3,37%
5	cesar	0,94 MB	0,04%
6	rosemary	752,88 MB	29,16%
Espaço total ocupado:		2581,57 MB	
Espaço médio ocupado:		430,26 MB	