

## Aula 22 – Arquivos de texto

Dados são normalmente gravados num dispositivo de armazenamento externo para utilização posterior. Disco rígido, disco removível, CD, DVD, pen drive, memory stick, dispositivos SSD, etc.) são exemplos de dispositivos de armazenamento. Outros dispositivos mais antigos, como fitas magnéticas também podem ser utilizadas.

Em um disco, ou mesmo em alguma memória de massa (pen drive, memory stick, SSD, etc.) podemos armazenar vários arquivos cujo conteúdo está distribuído pelo dispositivo não necessariamente de forma contígua. Um diretório desse dispositivo descreve quais os arquivos presentes e sua localização.

Arquivos podem conter qualquer tipo de dados (textos, dados binários, imagens, vídeos, etc.).

Um arquivo de texto pode ser criado através de qualquer editor de texto (bloco de notas, word, etc.) e normalmente possuem a extensão .txt. Um programa em Python, ou em qualquer outra linguagem também é um texto e nestes casos possuem extensão específica (.py, .c, .php, etc.). Podemos usar o próprio editor de programas do IDE Python para criar arquivos de texto.

Embora ocupe mais espaço, os arquivos em texto ou de texto são a forma mais portátil para os dados. Podem facilmente serem usados por diferentes programas feitos em diferentes linguagens.

A seguir, veremos a utilização de arquivos de texto em Python.

### Métodos para abrir e fechar arquivos

Antes de usar o conteúdo de um arquivo é preciso que o sistema o localize dentro do dispositivo e o associe ao seu programa. Isso é feito através do método open. Da mesma forma para desassociá-lo do seu programa usamos o método close.

Método	Uso	Explicação
open	<code>open(nome_arquivo, 'r')</code>	Abre um arquivo chamado nome_arquivo para leitura. Posiciona a leitura no início do arquivo. Retorna uma referência a um objeto <i>file</i> .
open	<code>open(nome_arquivo, 'w')</code>	Abre um arquivo chamado nome_arquivo para escrita. Se o arquivo já existe, o seu conteúdo anterior é perdido. Retorna uma referência a um objeto <i>file</i> .
close	<code>nome_arquivo.close()</code>	Finaliza o uso do arquivo referenciado pela variável nome_arquivo.

## As linhas de um arquivo de texto

Um arquivo de texto é constituído por linhas. No final de cada linha há um caractere especial (newline, ou \n). Esse caractere tem o efeito de mudar para a próxima linha quando o enviamos para o vídeo ou impressora.

Como exemplo, vamos construir um arquivo, onde cada linha tenha os seguintes campos:

Nome, Endereço, Documento de Identificação, Data de Nascimento

Os elementos da linha são strings e o separador de campos é o caractere espaço (branco). Temos que ser rigorosos com relação aos campos. Um caractere espaço separa os campos. Assim, não podemos ter espaços dentro dos campos, mas podemos ter mais de um espaço entre os campos:

Nome – duas strings - nome e sobrenome

Endereço – três strings - tipo (rua, avenida, etc.), um nome e um número

Documento de Identificação – uma string

Data de Nascimento – uma string

Vamos digitá-lo com o seguinte conteúdo e dar o nome de “pessoas.txt”:

```
Jose Santos Rua Alvarenga 311 232345-8 12/04/1973
Carlos Dias Avenida Ipiranga 712-apto34 213453456-9 14/08/2000
Ricardo Amaral Rua Andradas 356-casa7 894748-32 15.08.1934
Maria Silva Av. Paulista 123-cj54 777888999-00 12/12/1912
Casemiro Abreu Alameda Nothman 657 01203404-7 08/11/1984
Antonio Alvarenga Rua Servilio 65 43434343-43 09.08.2016
```

## Acessando cada linha do arquivo

Aberto o arquivo, ele se comporta para o programa como uma tabela, onde cada elemento é uma linha. Podemos então varrê-lo com um comando for.

O programa abaixo lê o arquivo e mostra o seu conteúdo no vídeo. Usamos o método split de strings para separar cada um dos campos:

```
ref_arq = open("pessoas.txt", "r")

for linha in ref_arq:
    itens = linha.split()
    print("Nome:", itens[0], itens[1])
    print("Endereço: ", itens[2], itens[3], itens[4])
    print("Documento de Identidade: ", itens[5])
    print("Data de Nascimento: ", itens[6])
    print("\n\n")

ref_arq.close()
```

Saida:

Nome: Jose Santos  
Endereço: Rua Alvarenga 311  
Documento de Identidade: 232345-8  
Data de Nascimento: 12/04/1973

Nome: Carlos Dias  
Endereço: Avenida Ipiranga 712-apto34  
Documento de Identidade: 213453456-9  
Data de Nascimento: 14/08/2000

Nome: Ricardo Amaral  
Endereço: Rua Andradas 356-casa7  
Documento de Identidade: 894748-32  
Data de Nascimento: 15.08.1934

Nome: Maria Silva  
Endereço: Av. Paulista 123-cj54  
Documento de Identidade: 777888999-00  
Data de Nascimento: 12/12/1912

Nome: Casemiro Abreu  
Endereço: Alameda Nothman 657  
Documento de Identidade: 01203404-7  
Data de Nascimento: 08/11/1984

Nome: Antonio Alvarenga  
Endereço: Rua Servilio 65  
Documento de Identidade: 43434343-43  
Data de Nascimento: 09.08.2016

### **Outra forma de acessar linha a linha**

O método `readline()` nos permite obter linha a linha do arquivo. Quando chega ao fim do arquivo, esse método retorna uma string vazia. Veja o programa abaixo com a mesma saída do anterior:

```

ref_arq = open("pessoas.txt", "r")
# ler a primeira linha
linha = ref_arq.readline()

# quando chegar ao fim do arquivo linha == "" (string vazia)
while linha:
    itens = linha.split()
    print("Nome:", itens[0], itens[1])
    print("Endereço: ", itens[2], itens[3], itens[4])
    print("Documento de Identidade: ", itens[5])
    print("Data de Nascimento: ", itens[6])
    print("\n\n")
    # ler a linha seguinte
    linha = ref_arq.readline()

ref_arq.close()

```

### Criando um novo arquivo

Voltando ao exemplo inicial, suponha agora que desejamos criar um novo arquivo em vez de imprimir no vídeo as mensagens com nome, endereço, documento e data do nascimento de cada elemento. Necessário então gravar o novo arquivo com o método write.

```

# Abrir arquivo de entrada
arq1 = open("pessoas.txt", "r")
# Abrir arquivo de saída
arq2 = open("novo_pessoas.txt", "w")

for linha in arq1:
    itens = linha.split()
    arq2.write('Nome: ' + itens[0] + ' ' + itens[1] + ' ' + '\n')
    arq2.write('Endereço: ' + itens[2] + ' ' + itens[3] + ' ' + itens[4] + '\n')
    arq2.write('Documento de Identidade: ' + itens[5] + ' ' + '\n')
    arq2.write('Data de Nascimento: ' + itens[6] + '\n')
# Fechar os arquivos
arq1.close()
arq2.close()

```

O arquivo “novo\_pessoas.txt” criado, tem o seguinte conteúdo:

```

Nome: Jose Santos
Endereço: Rua Alvarenga 311
Documento de Identidade: 232345-8
Data de Nascimento: 12/04/1973
Nome: Carlos Dias
Endereço: Avenida Ipiranga 712-apto34
Documento de Identidade: 213453456-9
Data de Nascimento: 14/08/2000
Nome: Ricardo Amaral
Endereço: Rua Andradas 356-casa7

```

Aula 22 – Arquivos de texto  
MAC110/115 – Python - Marcilio  
Documento de Identidade: 894748-32  
Data de Nascimento: 15.08.1934  
Nome: Maria Silva  
Endereço: Av. Paulista 123-cj54  
Documento de Identidade: 777888999-00  
Data de Nascimento: 12/12/1912  
Nome: Casemiro Abreu  
Endereço: Alameda Nothman 657  
Documento de Identidade: 01203404-7  
Data de Nascimento: 08/11/1984  
Nome: Antonio Alvarenga  
Endereço: Rua Servilio 65  
Documento de Identidade: 43434343-43  
Data de Nascimento: 09.08.2016

### Outros métodos para acessar arquivos

Método	Uso	Efeito
write	<code>ref_arquivo.write(s)</code>	Adiciona o string <code>s</code> no final do arquivo. <code>ref_arquivo</code> deve ser uma referência a um arquivo que foi aberto para escrita ("w").
read(n)	<code>ref_arquivo.read()</code>	Lê e retorna um string de <code>n</code> caracteres ou o arquivo inteiro como um string se <code>n</code> não é fornecido.
readline(n)	<code>ref_arquivo.readline()</code>	Retorna como uma string a próxima linha do arquivo com todo o texto e incluindo o caractere de nova linha. Se <code>n</code> é fornecido como argumento então somente <code>n</code> caracteres são retornados se a linha tem mais do que <code>n</code> caracteres.
readlines(n)	<code>ref_arquivo.readlines()</code>	Retorna uma lista de strings, cada elemento da lista representando o conteúdo de uma linha do arquivo. Se <code>n</code> é fornecido então <code>n</code> caracteres serão lidos.

Para ler um arquivo a partir da primeira linha, é necessário abri-lo novamente. Podemos abri-lo mais de uma vez se necessário. A cada leitura, o marcador de próximo a ser lido se move para o caractere seguinte ao último lido. No caso de `readline` o marcador se move para o primeiro caractere da próxima linha. Nos casos de `read` ou `readlines`, sem o parâmetro `n`, a marca se move para o final do arquivo.

O exemplo abaixo mostra algumas variações do uso dos métodos:

```
# exemplos de read
ref_arq = open("pessoas.txt", "r")
print("* * * Mostrar todo o arquivo:")
linha = ref_arq.read()
print(linha)

ref_arq = open("pessoas.txt", "r")
print("* * * Mostrar os primeiros 10 caracteres do arquivo:")
```

```

linha = ref_arq.read(10)
print(linha)
print()

```

```

# exemplos de readline

```

```

ref_arq = open("pessoas.txt", "r")
print("* * * Mostrar a primeira linha do arquivo:")
linha = ref_arq.readline()
print(linha)

```

```

ref_arq = open("pessoas.txt", "r")
print("* * * Mostrar os primeiros 20 caracteres do arquivo:")
linha = ref_arq.readline(20)
print(linha)
print()

```

```

# exemplos de readlines

```

```

ref_arq = open("pessoas.txt", "r")
print("* * * Ler todas as linhas do arquivo e mostrar linha a linha:")
todas_as_linhas = ref_arq.readlines()
for i in range(len(todas_as_linhas)):
    print(todas_as_linhas[i])

```

```

print("\n* * * Mostrar todas as linhas como uma lista:")
print(todas_as_linhas)
print()

```

```

ref_arq = open("pessoas.txt", "r")
print("* * * Ler e mostrar partes de uma linha:")
parte_de_linha1 = ref_arq.readline(20)
print(parte_de_linha1)
parte_de_linha2 = ref_arq.readline(15)
print(parte_de_linha2)

```

```

ref_arq.close()

```

Saida:

```

* * * Mostrar todo o arquivo:

```

```

Jose Santos Rua Alvarenga 311 232345-8 12/04/1973
Carlos Dias Avenida Ipiranga 712-apto34 213453456-9 14/08/2000
Ricardo Amaral Rua Andradas 356-casa7 894748-32 15.08.1934
Maria Silva Av. Paulista 123-cj54 777888999-00 12/12/1912
Casemiro Abreu Alameda Nothman 657 01203404-7 08/11/1984
Antonio Alvarenga Rua Servilio 65 43434343-43 09.08.2016

```

```

* * * Mostrar os primeiros 10 caracteres do arquivo:

```

```

Jose Santo

```

**\* \* \* Mostrar a primeira linha do arquivo:**

Jose Santos Rua Alvarenga 311 232345-8 12/04/1973

**\* \* \* Mostrar os primeiros 20 caracteres do arquivo:**

Jose Santos Rua Alva

**\* \* \* Ler todas as linhas do arquivo e mostrar linha a linha:**

Jose Santos Rua Alvarenga 311 232345-8 12/04/1973

Carlos Dias Avenida Ipiranga 712-apto34 213453456-9 14/08/2000

Ricardo Amaral Rua Andradas 356-casa7 894748-32 15.08.1934

Maria Silva Av. Paulista 123-cj54 777888999-00 12/12/1912

Casemiro Abreu Alameda Nothman 657 01203404-7 08/11/1984

Antonio Alvarenga Rua Servilio 65 43434343-43 09.08.2016

**\* \* \* Mostrar todas as linhas como uma lista:**

```
['Jose Santos Rua Alvarenga 311 232345-8 12/04/1973\n', 'Carlos  
Dias Avenida Ipiranga 712-apto34 213453456-9 14/08/2000\n', 'Ricardo  
Amaral Rua Andradas 356-casa7 894748-32 15.08.1934\n', 'Maria Silva  
Av. Paulista 123-cj54 777888999-00 12/12/1912\n', 'Casemiro Abreu  
Alameda Nothman 657 01203404-7 08/11/1984\n', 'Antonio Alvarenga  
Rua Servilio 65 43434343-43 09.08.2016']
```

**\* \* \* Ler e mostrar partes de uma linha:**

Jose Santos Rua Alva

renga311 23234

Observe agora o que foi impresso no caso:

**\* \* \* Mostrar todas as linhas como uma lista:**

Nesse caso, são mostrados os vários elementos da lista com o seu real conteúdo. Veja que no final da linha também aparece o \n.

Exercícios:

P22.1) Supondo que o arquivo pessoas tivesse agora um formato mais livre com os campos separados por vírgulas:

```
Jose Carlos dos Santos,Rua do Alvarenga 311, 232345-8, 12/04/1973  
Carlos Dias Antunes, Avenida Grito do Ipiranga 712 apto 34, 213453456-9, 14/08/2000  
Ricardo do Amaral Peixoto, Rua Andradas e Silva 356 casa 7, 894748-32, 15.08.1934  
Maria Aparecida da Silva, Av. Paulista 123 - cj 54, 777888999-00, 12/12/1912  
Casemiro dos Anjos Abreu, Alameda Nothman e Gothman 657, 01203404-7, 08/11/1984  
Antonio Amador Alvarenga Alves, Rua Servilio 65 Fundos, 43434343-43, 09.08.2016
```

Modifique o primeiro programa acima para que produza a mesma saída com esse arquivo.

A função `split()` pode ser usada também com o separador vírgula ou com qualquer outra substring como separador. Quando não especificado, o separador é qualquer sequência de um ou mais espaços. Possui dois outros parâmetros. O primeiro é o string separador e o segundo a quantidade de vezes a separar. Exemplos abaixo:

```
st = 'o rato      roeu a      roupa do rei de roma'
print(st.split())
print(st.split('r', 2))
print(st.split('r'))
print(st.split('ro'))
```

Saída:

```
['o', 'rato', 'roeu', 'a', 'roupa', 'do', 'rei', 'de', 'roma']
['o ', 'ato      ', 'oeu a      roupa do rei de roma']
['o ', 'ato      ', 'oeu a      ', 'oupa do ', 'ei de ', 'oma']
['o rato      ', 'eu a      ', 'upa do rei de ', 'ma']
```

P22.2) Dado um arquivo contendo:

Nome, NUSP, P1, P2 – campos separados por vírgulas.

Faça um programa que imprima o conteúdo desse arquivo juntamente com a média das provas.

Exemplo do arquivo dado:

```
Jose Carlos da Silva, 98987678, 7.0, 8.3
Antonio Silva e Santos, 4748392, 4.9, 8.2
```

P22.3) Idem, construindo também um arquivo de saída agregando a média com os campos separados por vírgula.

P22.4) Idem, supondo que o arquivo contenha os campos:

Nome, NUSP, P1, P2, EP1, EP2, EP3

Imprima a média das provas  $(P1 + 2P2) / 3$ , e a média dos EPs  $(EP1+2EP2+3EP3) / 6$

P22.5) Idem, construindo também um arquivo de saída agregando as médias com os campos separados por vírgula.

P22.6) Dado o arquivo contendo Nome, Endereço, Documento, Data do Nascimento. Cada campo está numa nova linha. Portanto 4 linhas por pessoa. Exemplo:

```
Jose dos Santos
Rua Alvarenga da Silva 311 - fundos
232345-8
12/04/1973
```



Aula 22 – Arquivos de texto  
MAC110/115 – Python - Marcilio  
Carlos de Santana Dias Leite  
Avenida do Ipiranga 712 - Apto34 – Bom Retiro – CEP 05454 050  
213453456-9  
14/08/2000

...

Faça um programa que leia e imprima os dados de cada uma das pessoas como o exemplo inicial.