

Data Science Academy - Python Fundamentos - Capítulo 4

Download: <http://github.com/dsacademybr>

```
In [1]: # Versão da Linguagem Python
        from platform import python_version
        print('Versão da Linguagem Python Usada Neste Jupyter Notebook:', python_version())
```

Versão da Linguagem Python Usada Neste Jupyter Notebook: 3.7.6

**** ATENÇÃO ****

Caso você tenha problemas com acentos nos arquivos:

Primeiro, recomendamos a leitura do material sobre Formato Unicode, ao final do capítulo 4.

Uma forma de resolver esse problema, é abrir o arquivo em um editor de texto como o Sublime Text, clicar em File - Save with Encoding e então salvar com encoding UTF-8.

Outra opção é incluir o parâmetro encoding='utf8' ao abrir o arquivo para leitura ou escrita.

Manipulação de Arquivos

- Arquivos TXT
- Arquivos CSV
- Arquivos JSON

Manipulando Arquivos TXT

```
In [3]: texto = "Cientista de Dados .\n"
        texto = texto + "Esses profissionais precisam se especializar.\n"
        texto += "E claro, em Big Data."
```

```
In [4]: print(texto)
```

Cientista de Dados .

Esses profissionais precisam se especializar.
E claro, em Big Data.

```
In [5]: # Importando o módulo os
import os
```

```
In [6]: # Criando um arquivo
arquivo = open(os.path.join('cientista1.txt'), 'w')
```

```
In [7]: # Gravando os dados no arquivo
for palavra in texto.split():
    arquivo.write(palavra+' ')
```

```
In [8]: # Fechando o arquivo
arquivo.close()
```

```
In [9]: # Lendo o arquivo
arquivo = open('cientista1.txt', 'r')
conteudo = arquivo.read()
arquivo.close()

print(conteudo)
```

Cientista de Dados . Esses profissionais precisam se especializar. E claro, em Big Data.

Usando a expressão with

O método close() é executado automaticamente

```
In [10]: with open('arquivos/cientista.txt', 'r') as arquivo:
    conteudo = arquivo.read()
```

```
In [11]: print(len(conteudo))

60
```

```
In [12]: print(conteudo)
```

Cientista de Dados Ã©
Cientista de Dados Ã© a profissÃ£o

```
In [13]: with open('arquivos/cientista.txt', 'w') as arquivo:
```

```
arquivo.write(texto[:21])
arquivo.write('\n')
arquivo.write(texto[:33])
```

```
In [14]: # Lendo o arquivo
arquivo = open('cientista1.txt','r')
conteudo = arquivo.read()
arquivo.close()

print (conteudo)
```

Cientista de Dados . Esses profissionais precisam se especializar. E claro, em Big Data.

Manipulando Arquivos CSV (comma-separated values)

```
In [15]: # Importando o módulo csv
import csv
```

```
In [20]: with open('numeros1.csv','w') as arquivo:
writer = csv.writer(arquivo)
writer.writerow(('primeira','segunda','terceira'))
writer.writerow((55,93,76))
writer.writerow((62,14,86))
```

```
In [21]: # Leitura de arquivos csv
with open('numeros1.csv','r') as arquivo:
    leitor = csv.reader(arquivo)
    for x in leitor:
        print ('Número de colunas:', len(x))
        print(x)
```

```
Número de colunas: 3
['primeira', 'segunda', 'terceira']
Número de colunas: 0
[]
Número de colunas: 3
['55', '93', '76']
Número de colunas: 0
[]
Número de colunas: 3
['62', '14', '86']
Número de colunas: 0
[]
```

```
In [22]: # Código alternativo para eventuais problemas com linhas em branco no arquivo
```

```
with open('arquivos/numeros.csv','r', encoding='utf8', newline = '\r\n') as arquivo:
    leitor = csv.reader(arquivo)
    for x in leitor:
        print ('Número de colunas:', len(x))
        print(x)
```

```
Número de colunas: 3
['primeira', 'segunda', 'terceira']
Número de colunas: 3
['55', '93', '76']
Número de colunas: 3
['62', '14', '86']
```

```
In [18]: # Gerando uma lista com dados do arquivo csv
with open('arquivos/numeros.csv','r') as arquivo:
    leitor = csv.reader(arquivo)
    dados = list(leitor)
```

```
print (dados)
```

```
[['primeira', 'segunda', 'terceira'], ['55', '93', '76'], ['62', '14', '86']]
```

```
In [19]: # Imprimindo a partir da segunda linha
for linha in dados[1:]:
    print (linha)
```

```
['55', '93', '76']
['62', '14', '86']
```

Manipulando Arquivos JSON (Java Script Object Notation)

JSON (JavaScript Object Notation) é uma maneira de armazenar informações de forma organizada e de fácil acesso. Em poucas palavras, ele nos dá uma coleção legível de dados que podem ser acessados de forma muito lógica. Pode ser uma fonte de Big Data.

```
In [20]: # Criando um dicionário
dict = {'nome': 'Guido van Rossum',
        'linguagem': 'Python',
        'similar': ['c', 'Modula-3', 'lisp'],
        'users': 1000000}
```

```
In [21]: for k,v in dict.items():
        print (k,v)
```

```
nome Guido van Rossum
linguagem Python
```

```
similar ['c', 'Modula-3', 'lisp']  
users 1000000
```

```
In [22]: # Importando o módulo Json  
import json
```

```
In [23]: # Convertendo o dicionário para um objeto json  
json.dumps(dict)
```

```
Out[23]: '{"nome": "Guido van Rossum", "linguagem": "Python", "similar": ["c", "Modula-3", "lisp"], "users": 1000000}'
```

```
In [24]: # Criando um arquivo Json  
with open('arquivos/dados.json', 'w') as arquivo:  
    arquivo.write(json.dumps(dict))
```

```
In [25]: # Leitura de arquivos Json  
with open('arquivos/dados.json', 'r') as arquivo:  
    texto = arquivo.read()  
    data = json.loads(texto)
```

```
In [26]: print (data)  
  
{'nome': 'Guido van Rossum', 'linguagem': 'Python', 'similar': ['c', 'Modula-3', 'lisp'], 'users': 1000000}
```

```
In [27]: print (data['nome'])  
  
Guido van Rossum
```

```
In [28]: # Imprimindo um arquivo Json copiado da internet  
from urllib.request import urlopen  
  
response = urlopen("http://vimeo.com/api/v2/video/57733101.json").read().decode('utf8')  
data = json.loads(response)[0]
```

```
In [29]: print ('Título: ', data['title'])  
print ('URL: ', data['url'])  
print ('Duração: ', data['duration'])  
print ('Número de Visualizações: ', data['stats_number_of_plays'])
```

```
Título: The Good Man trailer  
URL: https://vimeo.com/57733101  
Duração: 143  
Número de Visualizações: 5881
```

```
In [30]: # Copiando o conteúdo de um arquivo para outro
```

```
import os

arquivo_fonte = 'arquivos/dados.json'
arquivo_destino = 'arquivos/json_data.txt'
```

```
In [31]: # Método 1
with open(arquivo_fonte, 'r') as infile:
    text = infile.read()
    with open(arquivo_destino, 'w') as outfile:
        outfile.write(text)
```

```
In [32]: # Método 2
open(arquivo_destino, 'w').write(open(arquivo_fonte, 'r').read())
```

Out[32]: 107

```
In [33]: # Leitura de arquivos Json
with open('arquivos/json_data.txt', 'r') as arquivo:
    texto = arquivo.read()
    data = json.loads(texto)
```

```
In [34]: print(data)

{'nome': 'Guido van Rossum', 'linguagem': 'Python', 'similar': ['c', 'Modula-3', 'lisp'], 'users': 1000000}
```

FIM

Obrigado - Data Science Academy - facebook.com/dsacademybr