

Carregando e armazenando dados em vários formatos de arquivos

Trabalhando com JSON

Carregando e armazenando dados

Trabalhando com JSON

Para trabalhar com JSON temos uma biblioteca padrão chamada json. Vamos iniciar importando pandas, numpy e json.

```
import pandas as pd  
import numpy as np  
import json
```

Carregando e armazenando dados

Trabalhando com JSON

Vamos agora criar uma string contendo um conteúdo em formato json.

```
string_json = """
{
    "marca": "Renault",
    "veiculos":
    [
        {"modelo": "Sandero", "lançamento": "2007"},
        {"modelo": "Kwid", "lançamento": "2017"},
        {"modelo": "Duster", "lançamento": "2011"},
        {"modelo": "Captur", "lançamento": "2017"},
        {"modelo": "Logan", "lançamento": "2007"}
    ]
}
"""
```

Carregando e armazenando dados

Trabalhando com JSON

Nós podemos criar um objeto Python (dicionário) através do método `loads` da biblioteca `json` informando a string JSON.

```
In [3]: obj_python = json.loads(string_json)
        obj_python

Out[3]: {'marca': 'Renault',
        'veiculos': [{'modelo': 'Sandero', 'lancamento': '2007'},
                      {'modelo': 'Kwid', 'lancamento': '2017'},
                      {'modelo': 'Duster', 'lancamento': '2011'},
                      {'modelo': 'Captur', 'lancamento': '2017'},
                      {'modelo': 'Logan', 'lancamento': '2007'}]}
```

```
In [4]: type(obj_python)

Out[4]: dict
```

Carregando e armazenando dados

Trabalhando com JSON

Podemos também converter um dicionário Python em JSON usando o método `dumps` da biblioteca `json`.

```
In [5]: meu_json = json.dumps(obj_python)
        meu_json
```

```
Out[5]: '{"marca": "Renault", "veiculos": [{"modelo": "Sandero", "lancamento": "2007"}, {"modelo": "Kwid", "lancamento": "2017"}, {"modelo": "Duster", "lancamento": "2011"}, {"modelo": "Captur", "lancamento": "2017"}, {"modelo": "Logan", "lancamento": "2007"}]}'
```

```
In [6]: type(meu_json)
```

```
Out[6]: str
```

Carregando e armazenando dados

Trabalhando com JSON

Após ter carregado o JSON para um dicionário, podemos criar um DataFrame à partir deste dicionário.

```
In [7]: modelos = pd.DataFrame(obj_python['veiculos'])  
modelos
```

```
Out[7]:
```

| | modelo | lançamento |
|---|---------|------------|
| 0 | Sandero | 2007 |
| 1 | Kwid | 2017 |
| 2 | Duster | 2011 |
| 3 | Captur | 2017 |
| 4 | Logan | 2007 |

Carregando e armazenando dados

Trabalhando com JSON

```
In [8]: ano_lancamento = pd.DataFrame(obj_python['veiculos'], columns=['lancamento'])  
ano_lancamento
```

```
Out[8]:
```

| | lancamento |
|---|------------|
| 0 | 2007 |
| 1 | 2017 |
| 2 | 2011 |
| 3 | 2017 |
| 4 | 2007 |

Carregando e armazenando dados

Trabalhando com JSON

Podemos carregar um arquivo JSON diretamente para um DataFrame ou Series usando o método `read_json`.

```
In [9]: dados = pd.read_json('exemplo.json')
        dados
```

Out[9]:

| | marca | veiculos |
|---|---------|---|
| 0 | Renault | {'modelo': 'Sandero', 'lançamento': '2007'} |
| 1 | Renault | {'modelo': 'Kwid', 'lançamento': '2017'} |
| 2 | Renault | {'modelo': 'Duster', 'lançamento': '2011'} |
| 3 | Renault | {'modelo': 'Captur', 'lançamento': '2017'} |
| 4 | Renault | {'modelo': 'Logan', 'lançamento': '2007'} |

Carregando e armazenando dados

Trabalhando com JSON

Se não for informado o tipo, por padrão o objeto será um DataFrame.

```
In [10]: type(dados)
```

```
Out[10]: pandas.core.frame.DataFrame
```

Carregando e armazenando dados

Trabalhando com JSON

Se quisermos carregar para um objeto Series, teremos que especificar.

```
In [11]: dados = pd.read_json('exemplo.json', typ='series')
         dados
```

```
Out[11]: marca                                Renault
veiculos    [{'modelo': 'Sander', 'lançamento': '2007'}, ...]
dtype: object
```

```
In [12]: type(dados)
```

```
Out[12]: pandas.core.series.Series
```

Carregando e armazenando dados

Trabalhando com JSON

Se cada linha do arquivo for um JSON diferente, temos que informar o parâmetro `lines` como verdadeiro.

Veja o conteúdo do arquivo `exemplo2.json`:

```
{"modelo": "Sander", "lancamento": "2007"}  
{"modelo": "Kwid", "lancamento": "2017"}  
{"modelo": "Duster", "lancamento": "2011"}  
{"modelo": "Captur", "lancamento": "2017"}  
{"modelo": "Logan", "lancamento": "2007"}
```

Carregando e armazenando dados

Trabalhando com JSON

```
In [13]: dados = pd.read_json('exemplo2.json', lines=True)
```

```
In [14]: dados
```

```
Out[14]:
```

| | modelo | lançamento |
|---|---------|------------|
| 0 | Sandero | 2007 |
| 1 | Kwid | 2017 |
| 2 | Duster | 2011 |
| 3 | Captur | 2017 |
| 4 | Logan | 2007 |

Carregando e armazenando dados

Trabalhando com JSON

Para exportar os dados de um objeto Series ou DataFrame, podemos usar o método `to_json()` destes objetos.

Primeiro vamos criar um dicionário com a população dos estados brasileiros.

Carregando e armazenando dados

Trabalhando com JSON

```
populacao_estados = {'Sao Paulo':45538936,  
'Minas Gerais':21040662, 'Rio de Janeiro':17159960, 'Bahia':14812617,  
'Parana':11348937,  
'Rio Grande do Sul':11329605, 'Pernambuco':9496294,  
'Ceara':9075649, 'Para':8513497, 'Santa Catarina':7075494,  
'Maranhao':7035055, 'Goias':6921161, 'Amazonas':4080611,  
'Paraiba':3996496, 'Espirito Santo':3972388,  
'Rio Grande do Norte':3479010, 'Mato Grosso':3441998,  
'Alagoas':3322820, 'Piaui':3264531, 'Distrito Federal':2974703,  
'Mato Grosso do Sul':2748023, 'Sergipe':2278308, 'Rondonia':1757589,  
'Tocantins':1555229, 'Acre':869265, 'Amapa':829494,  
'Roraima':576568}
```

Carregando e armazenando dados

Trabalhando com JSON

Agora vamos criar uma série com este dicionário.

```
In [28]: populacao = pd.Series(populacao_estados)
          populacao
```

```
Out[28]: Sao Paulo      45538936
          Minas Gerais  21040662
          Rio de Janeiro 17159960
          Bahia         14812617
          Parana         11348937
          Rio Grande do Sul 11329605
          Pernambuco     9496294
          Ceara          9075649
          Para           8513497
          Santa Catarina 7075494
```

Carregando e armazenando dados

Trabalhando com JSON

Agora vamos exportar a série para JSON.

```
In [29]: meu_json = populacao.to_json()
```

```
In [30]: meu_json
```

```
Out[30]: '{"Sao Paulo":45538936,"Minas Gerais":21040662,"Rio de Janeiro":17159960,"Bahia":14812617,"Parana":11348937,"Rio Grande do Sul":11329605,"Pernambuco":9496294,"Ceara":9075649,"Para":8513497,"Santa Catarina":7075494,"Maranhao":7035055,"Goias":6921161,"Amazonas":4080611,"Paraiba":3996496,"Espirito Santo":3972388,"Rio Grande do Norte":3479010,"Mato Grosso":3441998,"Alagoas":3322820,"Piaui":3264531,"Distrito Federal":2974703,"Mato Grosso do Sul":2748023,"Sergipe":2278308,"Rondonia":1757589,"Tocantins":1555229,"Acre":869265,"Amapa":829494,"Roraima":576568}'
```


Carregando e armazenando dados

Trabalhando com JSON

Vamos agora criar um DataFrame com o dicionário e exportar os dados para JSON com o método `to_json()`.

```
populacao_estados = {'estado': ['Minas Gerais', 'Espirito Santo',  
'Rio de Janeiro', 'Sao Paulo'], 'populacao_2000': [17891494, 3097232,  
14391282, 37032403], 'populacao_2010': [19595309, 3512672,  
15993583, 41252160]}
```

Carregando e armazenando dados

Trabalhando com JSON

```
In [38]: populacao = pd.DataFrame(populacao_estados)
populacao
```

```
Out[38]:
```

| | estado | populacao_2000 | populacao_2010 |
|---|----------------|----------------|----------------|
| 0 | Minas Gerais | 17891494 | 19595309 |
| 1 | Espirito Santo | 3097232 | 3512672 |
| 2 | Rio de Janeiro | 14391282 | 15993583 |
| 3 | Sao Paulo | 37032403 | 41252160 |

```
In [39]: meu_json = populacao.to_json()
meu_json
```

```
Out[39]: '{"estado":{"0":"Minas Gerais","1":"Espirito Santo","2":"Rio de Janeiro","3":"Sao Paulo"},"populacao_2000":{"0":17891494,"1":3097232,"2":14391282,"3":37032403},"populacao_2010":{"0":19595309,"1":3512672,"2":15993583,"3":41252160}}'
```

FIM