

Lendo documentos

Arquivos CSV

Lendo documentos

Arquivos CSV

Ao realizar um scraping você pode ter a necessidade de tratar arquivos CSV. Arquivos CSV são arquivos separados por vírgula (Comma-separated values). Um programa que pode ser usado para ler e gravar arquivos CSV é o Microsoft Excel, por exemplo.

Lendo documentos

Arquivos CSV

Temos três alternativas de como tratar um arquivo CSV:

- 1) Baixar o arquivo manualmente e informar ao programa.
- 2) Baixar o arquivo diretamente no programa.
- 3) Recuperar o arquivo como uma string e inserir seu conteúdo em um objeto StringIO.

Lendo documentos

Arquivos CSV

A terceira opção é interessante por não envolver salvar arquivos em disco, assim podemos tratar o arquivo diretamente no sistema.

O módulo StringIO realiza leitura e escrita de strings em buffer (arquivos em memória).

Lendo documentos

Arquivos CSV

O Python possui uma biblioteca denominada csv que é muito eficiente para trabalhar com este tipo de arquivo.

Lendo documentos


Arquivos CSV

Exemplo de leitura e impressão do conteúdo de um arquivo csv em forma de lista.

```
from urllib.request import urlopen
from io import StringIO
import csv

url = input("Informe o caminho do arquivo CSV: ")
dados = urlopen(url).read().decode(encoding='utf-8', errors='ignore')
arqDados = StringIO(dados)
csvReader = csv.reader(arqDados)

for linha in csvReader:
    print(linha)
```

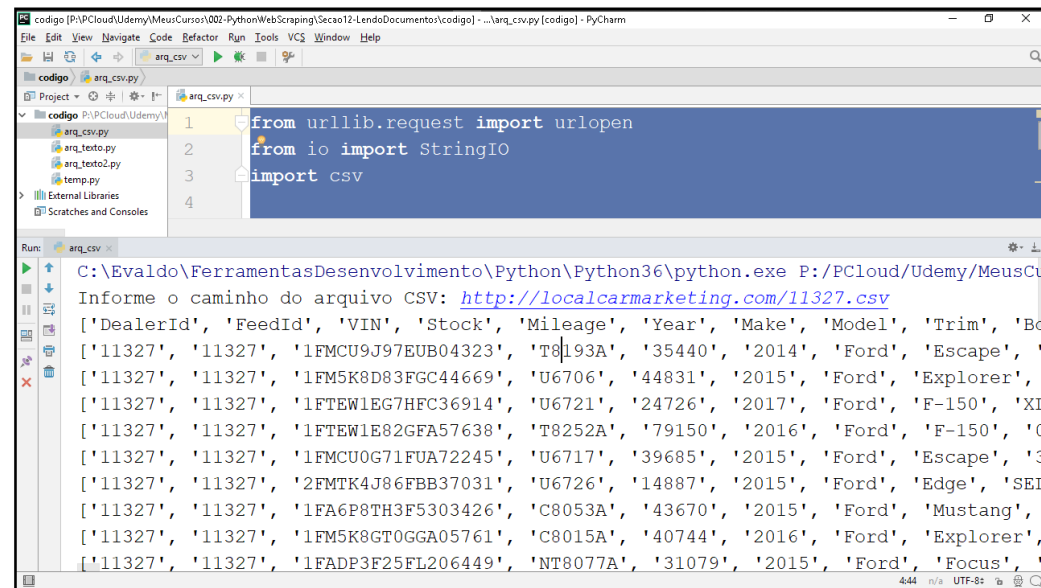


O reader retorna um objeto lista.

Lendo documentos

Arquivos CSV

Ao informar o endereço de um arquivo csv o sistema irá imprimir todas as linhas:



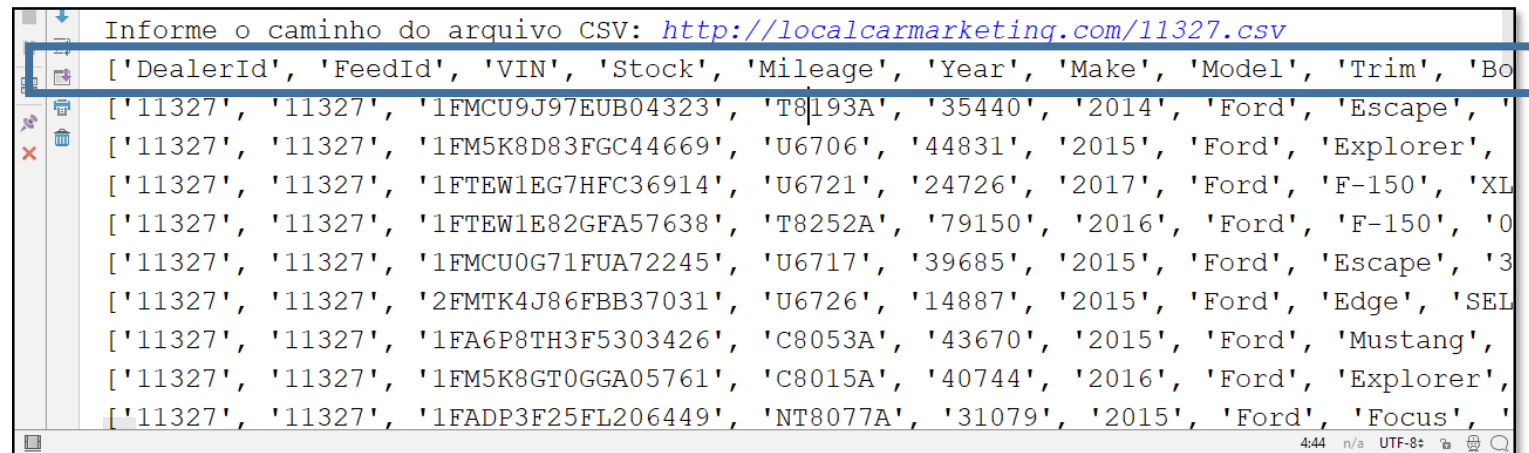
```
1 from urllib.request import urlopen
2 from io import StringIO
3 import csv
4
5 # Informe o caminho do arquivo CSV: http://localcarmarketing.com/11327.csv
6
7 ['DealerId', 'FeedId', 'VIN', 'Stock', 'Mileage', 'Year', 'Make', 'Model', 'Trim', 'Bo
8 ['11327', '11327', '1FMCU9J97EUB04323', 'T8193A', '35440', '2014', 'Ford', 'Escape', '
9 ['11327', '11327', '1FM5K8D83FGC44669', 'U6706', '44831', '2015', 'Ford', 'Explorer',
10 ['11327', '11327', '1FTEW1EG7HFC36914', 'U6721', '24726', '2017', 'Ford', 'F-150', 'XL
11 ['11327', '11327', '1FTEW1E82GFA57638', 'T8252A', '79150', '2016', 'Ford', 'F-150', '0
12 ['11327', '11327', '1FMCU0G71FUA72245', 'U6717', '39685', '2015', 'Ford', 'Escape', '3
13 ['11327', '11327', '2FMTK4J86FBB37031', 'U6726', '14887', '2015', 'Ford', 'Edge', 'SEL
14 ['11327', '11327', '1FA6P8TH3F5303426', 'C8053A', '43670', '2015', 'Ford', 'Mustang',
15 ['11327', '11327', '1FM5K8GT0GGA05761', 'C8015A', '40744', '2016', 'Ford', 'Explorer',
16 ['11327', '11327', '1FADP3F25FL206449', 'NT8077A', '31079', '2015', 'Ford', 'Focus', ']
```

O arquivo usado neste exemplo foi encontrado através de uma busca no Google por “filetype:csv” e seu endereço é:
<http://localcarmarketing.com/11327.csv>

Lendo documentos

Arquivos CSV

Observe no resultado que a primeira linha do arquivo é um cabeçalho (DealerId, FeedId, etc).



The screenshot shows a text editor window displaying a CSV file. The first line is the header: `['DealerId', 'FeedId', 'VIN', 'Stock', 'Mileage', 'Year', 'Make', 'Model', 'Trim', 'Bo`. This line is highlighted with a blue selection box. Below it are several rows of car data, each enclosed in single quotes and separated by commas. The status bar at the bottom indicates the file is 4:44, n/a, UTF-8, and has a search icon.

```
Informe o caminho do arquivo CSV: http://localcarmarketing.com/11327.csv
['DealerId', 'FeedId', 'VIN', 'Stock', 'Mileage', 'Year', 'Make', 'Model', 'Trim', 'Bo
['11327', '11327', '1FMCU9J97EUB04323', 'T8193A', '35440', '2014', 'Ford', 'Escape', '
['11327', '11327', '1FM5K8D83FGC44669', 'U6706', '44831', '2015', 'Ford', 'Explorer',
['11327', '11327', '1FTEW1EG7HFC36914', 'U6721', '24726', '2017', 'Ford', 'F-150', 'XL
['11327', '11327', '1FTEW1E82GFA57638', 'T8252A', '79150', '2016', 'Ford', 'F-150', '0
['11327', '11327', '1FMCU0G71FUA72245', 'U6717', '39685', '2015', 'Ford', 'Escape', '3
['11327', '11327', '2FMTK4J86FBB37031', 'U6726', '14887', '2015', 'Ford', 'Edge', 'SEL
['11327', '11327', '1FA6P8TH3F5303426', 'C8053A', '43670', '2015', 'Ford', 'Mustang',
['11327', '11327', '1FM5K8GT0GGA05761', 'C8015A', '40744', '2016', 'Ford', 'Explorer',
['11327', '11327', '1FADP3F25FL206449', 'NT8077A', '31079', '2015', 'Ford', 'Focus', '
4:44 n/a UTF-8
```

Se quisermos utilizar esta linha para retornar dicionários em vez de uma lista, podemos usar o método DictReader.

Lendo documentos

Arquivos CSV

Exemplo de leitura e impressão do conteúdo de um arquivo csv em forma de dicionário.

```
from urllib.request import urlopen
from io import StringIO
import csv

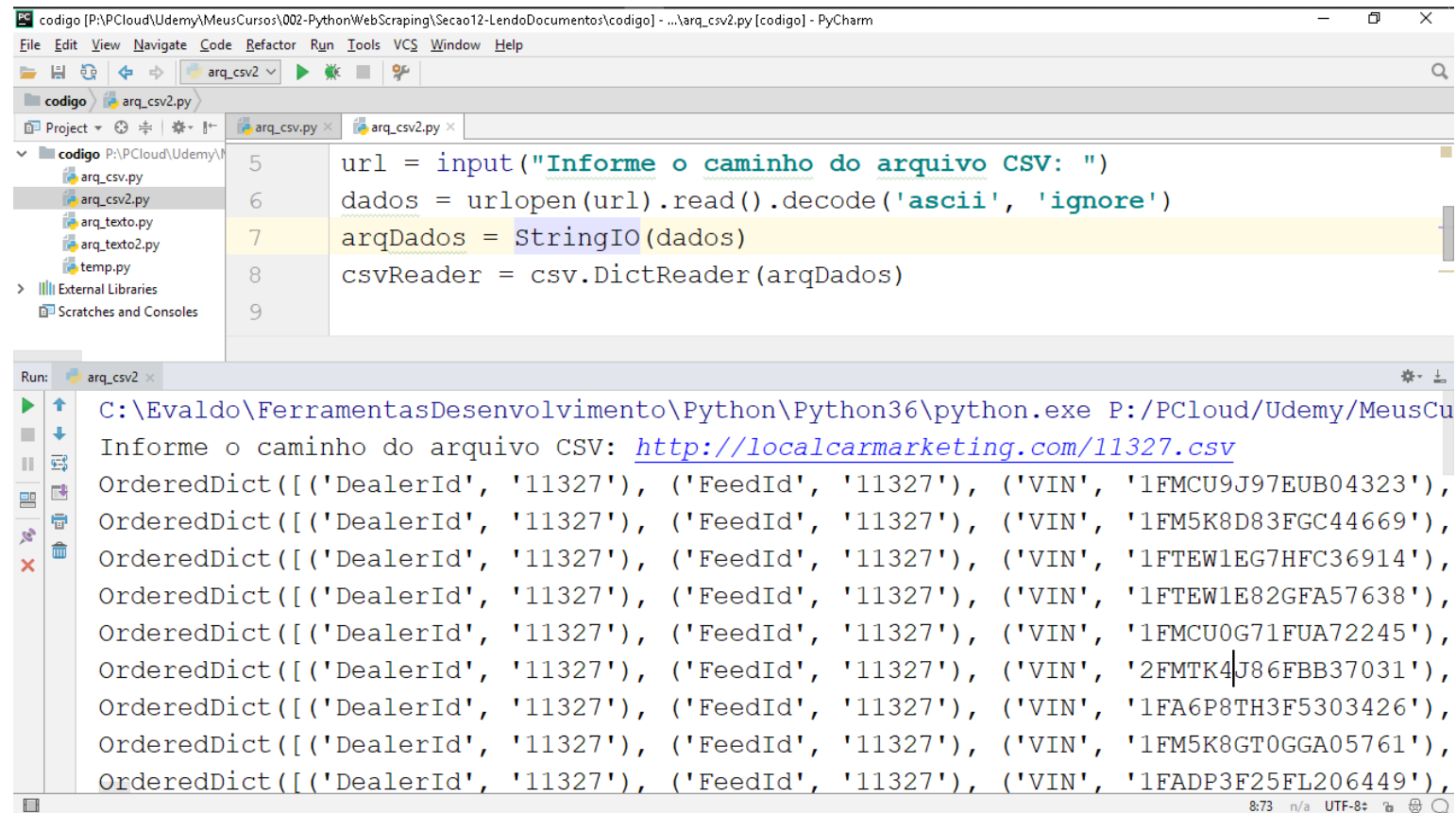
url = input("Informe o caminho do arquivo CSV: ")
dados = urlopen(url).read().decode('utf-8', 'ignore')
arqDados = StringIO(dados)
csvReader = csv.DictReader(arqDados)

for linha in csvReader:
    print(linha)
```

Lendo documentos

Arquivos CSV

Veja o resultado:



The screenshot displays the PyCharm IDE interface. The top pane shows the code editor for `arq_csv2.py` with the following Python code:

```
1 url = input("Informe o caminho do arquivo CSV: ")
2 dados = urlopen(url).read().decode('ascii', 'ignore')
3 arqDados = StringIO(dados)
4 csvReader = csv.DictReader(arqDados)
5
6
7
8
9
```

The bottom pane shows the Run console output for `arq_csv2`. The execution starts with the command prompt path and the input prompt. The user enters the URL `http://localcarmarketing.com/11327.csv`. The output then displays a list of 10 `OrderedDict` objects, each containing car details:

```
C:\Evaldo\FerramentasDesenvolvimento\Python\Python36\python.exe P:/PCloud/Udemy/MeusCu
Informe o caminho do arquivo CSV: http://localcarmarketing.com/11327.csv
OrderedDict([('DealerId', '11327'), ('FeedId', '11327'), ('VIN', '1FMCU9J97EUB04323'),
OrderedDict([('DealerId', '11327'), ('FeedId', '11327'), ('VIN', '1FM5K8D83FGC44669'),
OrderedDict([('DealerId', '11327'), ('FeedId', '11327'), ('VIN', '1FTEW1EG7HFC36914'),
OrderedDict([('DealerId', '11327'), ('FeedId', '11327'), ('VIN', '1FTEW1E82GFA57638'),
OrderedDict([('DealerId', '11327'), ('FeedId', '11327'), ('VIN', '1FMCU0G71FUA72245'),
OrderedDict([('DealerId', '11327'), ('FeedId', '11327'), ('VIN', '2FMTK4J86FBB37031'),
OrderedDict([('DealerId', '11327'), ('FeedId', '11327'), ('VIN', '1FA6P8TH3F5303426'),
OrderedDict([('DealerId', '11327'), ('FeedId', '11327'), ('VIN', '1FM5K8GT0GGA05761'),
OrderedDict([('DealerId', '11327'), ('FeedId', '11327'), ('VIN', '1FADP3F25FL206449'),
```

Lendo documentos

Arquivos CSV

Agora vamos escrever um programa que vai ler uma página da internet, pegar os links que apontam para arquivos csv e imprimir seu conteúdo.

PYTHON WEB SCRAPING – CSV
SAMPLE

<https://www.citibank.com.br/resources/datasources/atms.csv>

http://www.egr.rs.gov.br/upload/1438782049_RHPR_TRANS_9600_201506.csv

<http://www.granstoque.com.br/images/hidraulica.csv>

<https://camisetasdecorrida.com.br/usuarios2.csv>

http://www.cropr.org.br/uploads/transparencia/FC_2016_07.csv

<http://camaradealvorada.ro.gov.br/esic/arquivos/CatalogoDados.csv>

Lendo documentos

Arquivos CSV

```
from urllib.request import urlopen
from bs4 import BeautifulSoup
from io import StringIO
import csv

html = urlopen("https://evaldowolkers.wordpress.com/python-web-scraping-csv-sample/")
soup = BeautifulSoup(html, "html.parser")

for link in soup.findAll('a'):
    if link.get('href'):
        if ".csv" in link.get('href'):
            dados = urlopen(link.get('href')).read().decode(encoding='utf-8', errors='ignore')
            arqDados = StringIO(dados)
            csvReader = csv.reader(arqDados)

            for linha in csvReader:
                print(linha)
```

FIM