

Armazenando dados

Baixando arquivos de imagens e arquivos diversos

Trabalhando com arquivos CSV

Armazenando dados

Arquivos de mídia

Após a realização de um web scraper provavelmente você vai precisar armazenar estas informações de alguma forma.

Em se tratando de arquivos de mídia, podemos realizar o download e armazenar os arquivos localmente ou podemos armazenar somente o caminho dos arquivos.

Armazenando dados

Arquivos de mídia

Gravando apenas os caminhos (URLs) você economiza espaço em disco, executa o scraper com uma velocidade maior e diminui a carga no servidor evitando download de grandes arquivos.

É claro que existe o risco do arquivo não estar mais disponível no futuro, quando precisarmos realizar o download do mesmo.

Armazenando dados

Arquivos de mídia

Podemos usar a função `urllib.request.urlretrieve` para baixar arquivos à partir de qualquer URL remoto.

Veja um exemplo:

Digitei no Google a palavra “receitas” e localizei o site “<http://www.destemperados.com.br>” nos resultados.

Olhei o código-fonte do site e identifiquei as tags com o logotipo do site. A seguir veja um exemplo do uso da função `urlretrieve` para baixar esta imagem.

Armazenando dados

Arquivos de mídia

```
from urllib.request import urretrieve
from urllib.request import urlopen
from bs4 import BeautifulSoup

site = "http://www.destemperados.com.br/"
html = urlopen(site + "receitas")
bsObj = BeautifulSoup(html)
imageLocation = bsObj.find("a", {"title": "Destemperados"}).find("img")["src"]
urretrieve(site+imageLocation, "teste.jpg")
```

```
<a href="https://destemperados.clicrbs.com.br/"
title="Destemperados" class="hidden-sm hidden-xs">

</a>
```

Veja a imagem teste.jpg a seguir:



Armazenando dados

Arquivos de mídia

Isso funciona quando queremos baixar um único arquivo, porém, na maioria dos casos não vamos baixar um único arquivo e renomear.

Veja a seguir um exemplo para realização de download de diversos arquivos de imagens do site.

O exemplo seguinte retorna imagens do site <http://pythonscraping.com> que foi criado pelo autor do livro “Web Scraping with Python” (Ryan Mitchell) para realização de testes.

Armazenando dados

Arquivos de mídia

```
import os
from urllib.request import urlretrieve
from urllib.request import urlopen
from bs4 import BeautifulSoup

downloadDirectory = "downloaded"
baseUrl = "http://pythonscraping.com"

def getAbsoluteURL(baseUrl, source):
    if source.startswith("http://www."):
        url = "http://" + source[11:]
    elif source.startswith("http://"):
        url = source
    elif source.startswith("www."):
        url = "http://" + source
    else:
        url = baseUrl + "/" + source
    if baseUrl not in url or ".js" in url:
        return None
    return url
```

getAbsoluteURL:

Faz um tratamento na URL informada para retornar apenas links do próprio site, descartando links externos.

Armazenando dados

Arquivos de mídia

```
def getDownloadPath(baseUrl, absoluteUrl, downloadDirectory):  
    path = absoluteUrl.replace("www.", "")  
    path = path.replace(baseUrl, "")  
    path = downloadDirectory+path  
    directory = os.path.dirname(path)
```

```
    if not os.path.exists(directory):  
        os.makedirs(directory)
```

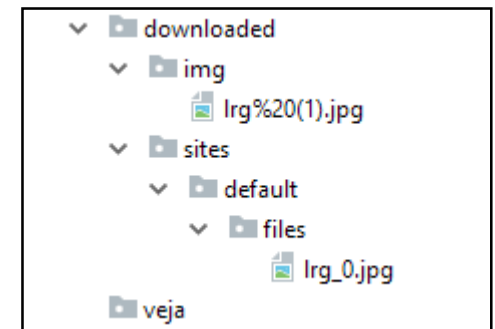
```
    return path
```

```
html = urlopen("http://www.pythonscraping.com")  
bsObj = BeautifulSoup(html, "html.parser")  
downloadList = bsObj.findAll(src=True)
```

```
for download in downloadList:  
    fileUrl = getAbsoluteURL(baseUrl, download["src"])  
    if fileUrl is not None:  
        urlretrieve(fileUrl, getDownloadPath(baseUrl, fileUrl, downloadDirectory))
```

getDownloadPath:

Cria pastas localmente seguindo o caminho dos arquivos que serão baixados.



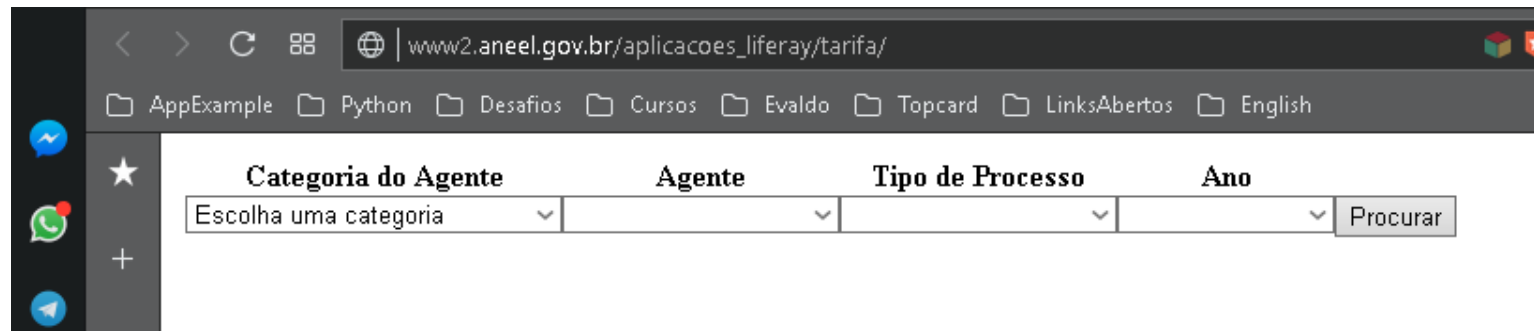
Armazenando dados

Baixando diversos arquivos

A seguir segue exemplo de uma necessidade levantada por um dos alunos do curso.

A necessidade era baixar todos arquivos do site:

http://www2.aneel.gov.br/aplicacoes_liferay/tarifa/



The screenshot shows a web browser window with the address bar displaying `www2.aneel.gov.br/aplicacoes_liferay/tarifa/`. Below the address bar, there are several browser tabs labeled 'AppExample', 'Python', 'Desafios', 'Cursos', 'Evaldo', 'Topcard', 'LinksAbertos', and 'English'. On the left side of the browser window, there is a sidebar with a star icon, a plus icon, and a minus icon. The main content area of the browser shows a search form with the following structure:

Categoria do Agente	Agente	Tipo de Processo	Ano	
Escolha uma categoria				Procurar

Armazenando dados

Baixando diversos arquivos

Em programação podemos ter várias soluções pra um mesmo problema, com web scraping não é diferente, depende muito de sua necessidade e da fonte de informações disponíveis (e do tempo que você tem disponível para analisar o problema).

No caso, como ele precisava de todos arquivos do site e o mesmo mostrava todos os arquivos ao selecionar o filtro “Todos” em todas as opções, podemos simplesmente salvar a página e trabalhar em cima do arquivo salvo.

Categoria do Agente		Agente	Tipo de Processo		Ano			
Todos		Todos	Todos		Todos	Procurar		
Resultado								
Agente	Categoria do Agente	Tipo de Processo	Data de Aniversário	Status Resultado	Nível TarifárioEstrutura TarifáriaAtos Regulatórios			
COPEL-DIS	Concessionária de Distribuição	Reajuste	24/06/2018	Definitivo				
CEB-DIS	Concessionária de Distribuição	Revisão Extraordinária	22/06/2018	Definitivo				

Armazenando dados

Baixando diversos arquivos

```
from urllib.request import urlopen
from urllib.request import urlretrieve
from bs4 import BeautifulSoup

html = urlopen('file:///P:/local/www2.aneel.gov.br.html')
bsObj = BeautifulSoup(html.read(), "html.parser")

for link in bsObj.find_all('a'):
    if "href" in link.attrs:
        try:
            link_completo = link.get('href')
            link_partes = link_completo.split("/")
            nome_arquivo = link_partes[-1]
            urlretrieve(link_completo, filename=nome_arquivo)
        except Exception as e:
            print(f"Erro arquivo: {link.get('href')}: ", e)
```

Armazenando dados

Armazenando dados em CSV

O CSV, ou comma-separated values (valores separados por vírgula), é um dos formatos de arquivo mais populares para o armazenamento de dados de planilha.

O Microsoft Excel, assim como o LibreOffice e muitos outros aplicativos dão suporte a este tipo de arquivo.

Veja um exemplo de conteúdo de arquivo CSV:

Nome,telefone,celular

Fulano,(11)1111-1111,(11)99999-9999

Cicrano,(27)2222-2222,(27)88888-8888

Beltrano,(31)3333-3333,(31)77777-7777

Um arquivo CSV também pode ter outros separadores, como tabulação, ponto e vírgula, etc.

Armazenando dados

Armazenando dados em CSV

O Python possui uma biblioteca nativa para trabalharmos com arquivos CSV.
Veja um exemplo:

```
import csv

arquivo_csv = open("exemplo.csv", "w", newline="")
try:
    writer = csv.writer(arquivo_csv)
    writer.writerow(['Nome', 'Telefone', 'Celular'])
    writer.writerow(['Fulano', ' (11) 1111-1111', ' (11) 99999-9999'])
    writer.writerow(['Cicrano', ' (22) 2222-2222', ' (27) 88888-8888'])
    writer.writerow(['Beltrano', ' (33) 3333-3333', ' (31) 77777-7777'])
finally:
    arquivo_csv.close()
```

1	Nome,Telefone,Celular
2	Fulano, (11) 1111-1111, (11) 99999-9999
3	Cicrano, (22) 2222-2222, (27) 88888-8888
4	Beltrano, (33) 3333-3333, (31) 77777-7777
5	

Armazenando dados

Armazenando dados em CSV

Como vimos, `csv.writer` é o módulo para escrever dados em arquivos CSV. Usamos `writerow` para escrever uma linha no arquivo. `Writerows` pode receber um objeto iterável como uma lista por exemplo e escrever várias linhas.

```
import csv

telefones = [['Nome', 'Telefone', 'Celular'],
              ['Fulano', '(11) 1111-1111', '(11) 99999-9999'],
              ['Cicrano', '(22) 2222-2222', '(27) 88888-8888'],
              ['Beltrano', '(33) 3333-3333', '(31) 77777-7777']]

with open('exemplo2.csv', 'w', newline='') as arquivo_csv:
    writer = csv.writer(arquivo_csv)
    writer.writerows(telefones)
```

```
1 Nome,Telefone,Celular
2 Fulano,(11) 1111-1111,(11) 99999-9999
3 Cicrano,(22) 2222-2222,(27) 88888-8888
4 Beltrano,(33) 3333-3333,(31) 77777-7777
5
```

Armazenando dados

Armazenando dados em CSV

Baixei um arquivo CSV aleatório na internet (estará disponível junto ao material da aula), para isso fui ao Google e pesquisei por “site:org.br filetype:csv”.

Dos resultados escolhi este arquivo:

http://www.cropr.org.br/uploads/transparencia/FC_2016_05.csv

Armazenando dados

Armazenando dados em CSV

Vou usar este arquivo para demonstrar a leitura de arquivos CSV.

```
import csv

with open('FC_2016_05.csv') as arquivo_csv:
    reader = csv.reader(arquivo_csv, delimiter=';')
    for linha in reader:
        print(linha)
```


Armazenando dados

Armazenando dados em CSV

Abrindo um arquivo informado por parâmetro:

```
import csv
import sys

if len(sys.argv)>1:
    with open(sys.argv[1], 'r') as arquivo_csv:
        reader = csv.reader(arquivo_csv, delimiter=';')
        for linha in reader:
            print(linha)
else:
    print("Informe o nome do arquivo.")
    print("Sintaxe:")
    print("$ python exemplo_csv4.py arquivo.csv")
```

Armazenando dados

Armazenando dados em CSV

Abrindo um arquivo através do link informado:

```
from urllib.request import urretrieve
import csv

link = input("Informe o link do arquivo csv: ")
delimitador = input("Informe o delimitador: ")
urretrieve(link, "arquivo_baixado.csv")
with open("arquivo_baixado.csv", 'r') as arquivo_csv:
    reader = csv.reader(arquivo_csv, delimiter=delimitador)
    for linha in reader:
        print(linha)
```

Fiz esta pesquisa no Google para localizar um arquivo csv:
site:org filetype:csv

Como exemplo informe o link a seguir que foi encontrado na busca:

<http://www.mafmc.org/s/CT1.csv>

Armazenando dados

Armazenando dados em CSV

Trabalhando com dicionários. Leitura com DictReader:

```
import csv

estado = []
with open("estados.csv", encoding="utf-8") as arq:
    reader = csv.DictReader(arq)
    for linha in reader:
        print(linha['Nome'] + " - " + linha['Capital'])
```

Armazenando dados

Armazenando dados em CSV

Trabalhando com dicionários. Escrita com DictWriter:

```
import csv

try:
    arquivo = open("pessoas.csv", "w", newline="")
    cabecalho = ["nome", "sobrenome"]
    writer = csv.DictWriter(arquivo, fieldnames=cabecalho)
    writer.writeheader()
    writer.writerow({"nome": "Evaldo", "sobrenome": "Wolkers"})
    writer.writerow({"nome": "Fulano", "sobrenome": "de Tal"})
    writer.writerow({"nome": "Cicrano", "sobrenome": "Souza"})
    writer.writerow({"nome": "Beltrano", "sobrenome": "Silva"})
finally:
    arquivo.close()
```

Armazenando dados

Armazenando dados em CSV

Realizando scraping em uma tabela HTML e salvando como CSV:

Carros de Need for Speed Payback		
Carros	Custo	Classes
Acura NSX 2017	156350	Arrancada, Corrida
Acura RSX-S	43450	Off-road, Corrida
Aston Martin DB11	154600	Arrancada, Drift, Corrida
Aston Martin Vulcan	705100	Arrancada, Drift, Corrida
Audi R8 V10 plus	178100	Arrancada, Drift, Corrida
Audi S5 Sportback	91000	Drift, Corrida
BMW M2	94350	Arrancada, Drift, Corrida
BMW M3 E46	93850	Arrancada, Drift, Off-road, Corrida, Fuga
BMW M3 E92	105150	Arrancada, Drift, Corrida
BMW M3 Evolution II E30	77950	Arrancada, Drift, Off-road, Corrida, Fuga
BMW M4 GTS	105750	Arrancada, Drift, Corrida, Fuga
BMW M5	125450	Arrancada, Drift, Corrida, Fuga

```
class="show-table content-media content-media--normal"> <p class="mc-column show-table__title
class="show-table__container"> <table class="show-table__content
show-table__content--highlight-top
"> <tr> <td>Carros</td> <td>Custo</td> <td>Classes
Corrida</td> </tr> <tr> <td>Acura RSX-S</td> <td>43450</td> <td>Off-road, Corrida</td> </tr>
Corrida</td> </tr> <tr> <td>Aston Martin Vulcan</td> <td>705100</td> <td>Arrancada, Drift, Co
Drift, Corrida</td> </tr> <tr> <td>Audi S5 Sportback</td> <td>91000</td> <td>Drift, Corrida</
</tr> <tr> <td>BMW M3 E46</td> <td>93850</td> <td>Arrancada, Drift, Off-road, Corrida, Fuga</
Corrida</td> </tr> <tr> <td>BMW M3 Evolution II E30</td> <td>77950</td> <td>Arrancada, Drift,
<td>Arrancada, Drift, Corrida, Fuga</td> </tr> <tr> <td>BMW M5</td> <td>125450</td> <td>Arran
road, Corrida, Fuga</td> </tr> <tr> <td>Buick GNX</td> <td>56500</td> <td>Arrancada, Drift, C
Drift, Off-road, Corrida, Fuga</td> </tr> <tr> <td>Chevrolet C10 Stepside Pickup</td> <td>435
<td>Chevrolet Camaro SS</td> <td>49850</td> <td>Arrancada, Drift, Corrida</td> </tr> <tr> <td>
<td>Chevrolet Corvette Grand Sport</td> <td>116600</td> <td>Arrancada, Drift, Corrida</td> </
Corrida</td> </tr> <tr> <td>Dodge Challenger SRT8</td> <td>92300</td> <td>Arrancada, Off-road
Drift, Off-road, Corrida, Fuga</td> </tr> <tr> <td>Ford F-150 Raptor</td> <td>72950</td> <td>
```

<https://www.techtudo.com.br/noticias/2017/10/confira-a-lista-completa-de-carros-para-need-for-speed-payback.ghtml>

Armazenando dados

Armazenando dados em CSV

Realizando scraping em uma tabela HTML e salvando como CSV:

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

html = urlopen("https://www.techtudo.com.br/noticias/2017/10/confira-a-lista-completa-de-carros-para-need-for-speed-payback.ghhtml")
bsObj = BeautifulSoup(html, "html.parser")
tabela = bsObj.find("table")
linhas = tabela.findAll("tr")
arquivo_csv = open("carros.csv", "w", newline="")

for linha in linhas:
    colunas = linha.findAll("td")
    lista = []
    for coluna in colunas:
        lista.append(coluna.text)

    writer = csv.writer(arquivo_csv, delimiter=";")
    writer.writerow(lista)

arquivo_csv.close()
```

FIM