Ch10-2-Files-Advanced

October 30, 2020

1 Advanced Topics on Files

1.1 Working with HTML files

- fetch an HTML page from web
- parse the HTML file with BeautifulSoup library

[2]:
```
# fetch an html page
import urllib.request
url = 'https://rambasnet.github.io/teaching.html'
localfile = 'teaching.html'
urllib.request.urlretrieve(url, localfile)
```

[2]:
```
('teaching.html', <http.client.HTTPMessage at 0x7fa0c5e5cdc0>)
```

[3]:
```
with open(localfile) as f:
    data = f.read()
words = data.split(' ')
print('There are {0} words in the file.'.format(len(words)))
```

```
There are 9653 words in the file.
```

1.2 parsing HTML using BeautifulSoup library

- install Beautifulsoup library

  ```
  $ pip install bs4
  ```

- https://www.crummy.com/software/BeautifulSoup/bs4/doc/#
- Alternative is nltk (Natural Language Toolkit) library
- http://www.nltk.org/

1.3 Installing Parsers

- supports the HTML parser included in Python's standard library
- also supports a number of third-party Python parsers such as very fast `lxml` parser

[4]:
```
# can run terminal/bash commands from notebook using !
! pip install bs4
```

```
Collecting bs4
  Downloading bs4-0.0.1.tar.gz (1.1 kB)
Collecting beautifulsoup4
  Downloading beautifulsoup4-4.9.3-py3-none-any.whl (115 kB)
     |                | 115 kB 292 kB/s eta 0:00:01
Collecting soupsieve>1.2; python_version >= "3.0"
  Downloading soupsieve-2.0.1-py3-none-any.whl (32 kB)
Building wheels for collected packages: bs4
  Building wheel for bs4 (setup.py) … done
  Created wheel for bs4: filename=bs4-0.0.1-py3-none-any.whl size=1273
sha256=14cd4ba62626eff1dcc9a1ad692e42baa3a7b5c025cc58a0562bb32b998e4381
  Stored in directory: /Users/rbasnet/Library/Caches/pip/wheels/75/78/21/68b1245
49c9bdc94f822c02fb9aa3578a669843f9767776bca
Successfully built bs4
Installing collected packages: soupsieve, beautifulsoup4, bs4
Successfully installed beautifulsoup4-4.9.3 bs4-0.0.1 soupsieve-2.0.1
```

[10]:
```
# install lxml parser
! pip install lxml
```

```
Requirement already satisfied: lxml in
/Users/rbasnet/miniconda3/lib/python3.8/site-packages (4.6.1)
```

[13]:
```
from bs4 import BeautifulSoup
localfile = 'teaching.html'
with open(localfile) as f:
    #soup = BeautifulSoup(f, 'lxml') # used to but now not working!
    soup = BeautifulSoup(f, 'html.parser')
text = soup.get_text()
print(text)
```

```
Ram Basnet | Homepage
```

```
Dr. Ram Basnet
```

Associate Professor of Computer Science

Home

Teaching

Research

Resources

Contact

Teaching

Teaching Interests

Cybersecurity

Python, C++, SQL and NoSQL Databases, JavaScript, NodeJS

Data Science

Web Design and Secure Web App Development

Courses Taught at CMU

CSCI 106: Web1 - Web Page Design I
6

CSCI 110: Beg. Prog. Python & Lab
8

CSCI 111: CS1 - Foundation of CS
8

CSCI 112: CS2 - Data Structures
7

CSCI 206: Web2 - Web Page Design II
2

CSCI 250: CS3 - Intro to Algorithms
3

CSCI 310: Adv. Prog. Python
8

CSCI 310: Adv. Prog. JavaScript
1

CSCI 370: Computer Security
5

CSCI 420: Cybersecurity
6

CSCI 465: Net/App Security
6

FALL 2020 SCHEDULE

Mon
Tues
Wed
Thrs
Fri

10:00

Off. Hr.CH 32110-10:50

Adv. PythonCH 27610-10:50

Off. Hr.CH 32110-10:50

Adv. PythonCH 27610-10:50

Off. Hr.CH 32110-10:50

10:30

11:00

11:30

12:00

CS 1WS 12012-12:50

CS 1WS 12012-12:50

CS 1WS 12012-12:50

CS 1WS 12012-12:50

Off. Hr.CH 32112-12:50

12:30

1:00

1:30

2:00

Net/App SecWS 1202-2:50

Off. Hr.CH 3212-2:50

Net/App SecWS 1202-2:50

Net/App SecWS 1202-2:50

2:30

3:00

3:30

4:00

```
4:30




Home | Teaching |
        Research  |
        Resources |
         Contact       ©
        2019
```

[14]:
```
# break into lines and remove leading and trailing space on each line
lines = [line.strip() for line in text.splitlines()]
```

[15]:
```
print(lines[:20])
```

```
['', '', '', 'Ram Basnet | Homepage', '', '', '', '', '', '', '', '', '', '',
'', '', 'Dr. Ram Basnet', 'Associate Professor of Computer Science', '', '']
```

[16]:
```
# create list of words by spliting multi-word elements
words = [word.strip().lower() for line in lines for word in line.split()]
```

[17]:
```
print(words[:20])
```

```
['ram', 'basnet', '|', 'homepage', 'dr.', 'ram', 'basnet', 'associate',
'professor', 'of', 'computer', 'science', 'home', 'teaching', 'research',
'resources', 'contact', 'teaching', 'teaching', 'interests']
```

[18]:
```
print('There are {0} words in the file.'.format(len(words)))
```

```
There are 192 words in the file.
```

1.4 Find histogram of words

- use DefaultDict found in collections module
- https://docs.python.org/3/library/collections.html

[19]:
```
from collections import defaultdict
```

[20]:
```
hist = defaultdict(int)
for w in words:
    hist[w] += 1
```

[21]:
```
# print top 10 most common words
listHist = [(k, v) for k, v in hist.items()]
```

[22]:
```
print(listHist[:10])
```

```
[('ram', 2), ('basnet', 2), ('|', 5), ('homepage', 1), ('dr.', 1), ('associate',
1), ('professor', 1), ('of', 2), ('computer', 2), ('science', 2)]
```

[23]:
```
listHist.sort(key = lambda x: x[1], reverse=True)
```

[24]:
```
print(listHist[:10])
```

```
[('csci', 11), ('|', 5), ('-', 5), ('cs', 5), ('off.', 5), ('hr.ch', 5),
('teaching', 4), ('web', 4), ('adv.', 4), ('net/app', 4)]
```

1.4.1 Use Counter collection

- easier way!

[25]:
```
from collections import Counter
```

[26]:
```
hist = Counter(words)
```

[27]:
```
hist.most_common(10)
```

[27]:
```
[('csci', 11),
 ('|', 5),
 ('-', 5),
 ('cs', 5),
 ('off.', 5),
 ('hr.ch', 5),
 ('teaching', 4),
 ('web', 4),
 ('adv.', 4),
 ('net/app', 4)]
```

1.5 working with binary files

- the following example copies a binary file such as image

[11]:
```
fileSrc = './resources/brain.jpg'
fileDst = 'brain-copy.jpg'
with open(fileSrc, 'rb') as rbf:
    #rb - read binary mode
    data = rbf.read() # read the whole binary file
    with open(fileDst, 'wb') as wbf:
        wbf.write(data) # write the whole binary file
```

1.6 use checksum to compare if two files match exactly!

- checksum makes sure that not a single bit is different between the two files
- used in security
- import and use hashlib - https://docs.python.org/3/library/hashlib.html

[15]:
```
import hashlib
file1Contents = open(fileSrc, 'rb').read()
file2Contents = open(fileDst, 'rb').read()

file1ChkSum = hashlib.sha256(file1Contents).hexdigest()
file2ChkSum = hashlib.sha256(file2Contents).hexdigest()
if (file1ChkSum == file2ChkSum):
    print('two files\' checksums match!')
else:
    print('oops! two files\' checksums do NOT match!')
```

```
two files checksums match!
```

1.7 Python object serialization with pickle library

- https://docs.python.org/3/library/pickle.html
- pickle module implements binary protocols for serializing and de-serializing a Python object

- Pickling - serializing python object
- Unpickling - deserializing python object (inverse operation)
- Unpickling untrusted picked files could have security implications
  - e.g., executing system commands; installing and executing third-party malicious packages and modules; etc.
  - for more: https://owasp.org/www-project-top-ten/2017/A8_2017-Insecure_Deserialization

[29]:
```
import pickle
alist = list(range(2, 21, 2))
```

[30]:
```
print(alist)
```

```
[2, 4, 6, 8, 10, 12, 14, 16, 18, 20]
```

[31]:
```
# let's pickle alist; serialize a list
pickleFile = 'myPickle.pkl'
with open(pickleFile, 'wb') as p:
    pickle.dump(alist, p)
```

[32]:
```
# lets unpickle alist; deserialize a list
with open(pickleFile, 'rb') as p:
    blist = pickle.load(p)
```

[33]:
```
alist == blist
```

[33]:
```
True
```

[35]:
```
# dump Counter
with open('wordCounter.pkl', 'wb') as p:
    pickle.dump(hist, p)
```

[36]:
```
# load pickle
with open('wordCounter.pkl', 'rb') as p:
    newHist = pickle.load(p)
```

[37]:
```
hist == newHist
```

[37]:
```
True
```

[38]:
```
newHist.most_common(3)
```

[38]:
```
[('csci', 11), ('|', 5), ('-', 5)]
```

[ ]:
```

```