

Maheshvaran S

DS205229119

Lab11. Retrieving Data From Web and Parsing

Question1. Retrieve data from web page using URLLIB and print the frequency of words from that page.

In [8]:

```
import urllib.request
counts = dict()
fhand = urllib.request.urlopen('http://data.pr4e.org/romeo.txt')
for line in fhand:
    words = line.decode().split()
    for word in words:
        counts[word] = counts.get(word, 0) + 1
print(counts)
```

```
{'But': 1, 'soft': 1, 'what': 1, 'light': 1, 'through': 1, 'yonder': 1, 'window': 1,
'breaks': 1, 'It': 1, 'is': 3, 'the': 3, 'east': 1, 'and': 3, 'Juliet': 1, 'sun': 2,
'Arise': 1, 'fair': 1, 'kill': 1, 'envious': 1, 'moon': 1, 'Who': 1, 'already': 1,
'sick': 1, 'pale': 1, 'with': 1, 'grief': 1}
```

Question2. Retrieve and display all hyperlinks (ie., HREF attribute) from a webpage using BeautifulSoup

In [15]:

```
import urllib.request,urllib.parse,urllib.error
from bs4 import BeautifulSoup
import ssl
ctx=ssl.create_default_context()
ctx.check_hostname=False
ctx.verify_mode = ssl.CERT_NONE
url=input('enter url')
html=urllib.request.urlopen(url,context=ctx).read()
soup=BeautifulSoup(html,'html.parser')
tags=soup('a')
for tag in tags:
    print(tag.get('href',None))
```

```
enter urlhttps://www.crummy.com/software/BeautifulSoup/bs4/doc/
genindex.html
#
#beautiful-soup-documentation
http://www.crummy.com/software/BeautifulSoup/
http://www.crummy.com/software/BeautifulSoup/bs3/documentation.html
#porting-code-to-bs4
https://www.crummy.com/software/BeautifulSoup/bs4/doc.zh/
http://kondou.com/BS4/
https://www.crummy.com/software/BeautifulSoup/bs4/doc.ko/
https://www.crummy.com/software/BeautifulSoup/bs4/doc.ptbr
https://www.crummy.com/software/BeautifulSoup/bs4/doc.ru/
#getting-help
https://groups.google.com/forum/?fromgroups#!forum/beautifulsoup
#diagnose
#quick-start
#installing-beautiful-soup
http://www.crummy.com/software/BeautifulSoup/bs3/documentation.html
http://www.crummy.com/software/BeautifulSoup/download/4.x/
#problems-after-installation
#installing-a-parser
http://lxml.de/
http://code.google.com/p/html5lib/
```

```
#differences-between-parsers
#making-the-soup
#id17
#kinds-of-objects
#tag
#navigating-the-tree
#searching-the-tree
#name
#attributes
#multi-valued-attributes
#navigablestring
#navigating-the-tree
#searching-the-tree
#replace-with
#navigating-the-tree
#searching-the-tree
#beautifulsoup
#tag
#navigating-the-tree
#searching-the-tree
#modifying-the-tree
#tag
#comments-and-other-special-strings
#navigating-the-tree
#going-down
#navigating-using-tag-names
#searching-the-tree
#contents-and-children
#descendants
#string
#strings-and-stripped-strings
#going-up
#parent
#parents
#going-sideways
#next-sibling-and-previous-sibling
#next-siblings-and-previous-siblings
#going-back-and-forth
#next-element-and-previous-element
#next-elements-and-previous-elements
#searching-the-tree
#kinds-of-filters
#a-string
#a-regular-expression
#a-list
#true
#a-function
#find-all
#id12
#attrs
#recursive
#id13
#limit
#kwargs
#kinds-of-filters
#the-name-argument
#kinds-of-filters
#a-string
#a-regular-expression
#a-list
#a-function
#the-value-true
#the-keyword-arguments
#a-string
#a-regular-expression
#a-list
#a-function
#the-value-true
#searching-by-css-class
```

```
#multivalue
#the-string-argument
#a-string
#a-regular-expression
#a-list
#a-function
#the-value-true
#the-limit-argument
#the-recursive-argument
#calling-a-tag-is-like-calling-find-all
#find
#id12
#attrs
#recursive
#id13
#kwargs
#navigating-using-tag-names
#find-parents-and-find-parent
#id12
#attrs
#id13
#limit
#kwargs
#id12
#attrs
#id13
#parent
#parents
#find-next-siblings-and-find-next-sibling
#id12
#attrs
#id13
#limit
#kwargs
#id12
#attrs
#id13
#kwargs
#sibling-generators
#find-previous-siblings-and-find-previous-sibling
#id12
#attrs
#id13
#limit
#kwargs
#id12
#attrs
#id13
#kwargs
#sibling-generators
#find-all-next-and-find-next
#id12
#attrs
#id13
#limit
#kwargs
#id12
#attrs
#id13
#kwargs
#element-generators
#find-all-previous-and-find-previous
#id12
#attrs
#id13
#limit
#kwargs
#id12
```

```
#attrs
#id13
#kwargs
#element-generators
#css-selectors
https://facelessuser.github.io/soupsieve/
https://facelessuser.github.io/soupsieve/
#modifying-the-tree
#changing-tag-names-and-attributes
#attributes
#modifying-string
#append
#extend
#navigablestring-and-new-tag
#insert
#insert-before-and-insert-after
#clear
#extract
#decompose
#replace-with
#wrap
#unwrap
#smooth
#output
#pretty-printing
#non-pretty-printing
#encodings
#output-formatters
#get-text
#string-generators
#specifying-the-parser-to-use
#installing-a-parser
#differences-between-parsers
#encodings
#unicode-dammit
#output-encoding
#unicode-dammit
#smart-quotes
#inconsistent-encodings
#line-numbers
#comparing-objects-for-equality
#copying-beautiful-soup-objects
#advanced-parser-customization
#parsing-only-part-of-a-document
#soupstrainer
#searching-the-tree
#id12
#attrs
#id13
#kwargs
#searching-the-tree
#customizing-multi-valued-attributes
#handling-duplicate-attributes
#instantiating-custom-subclasses
#troubleshooting
#diagnose
#errors-when-parsing-a-document
#installing-a-parser
#parser-installation
#parser-installation
#version-mismatch-problems
#parsing-xml
#parser-installation
#other-parser-problems
#differences-between-parsers
http://www.w3.org/TR/html5/syntax.html#syntax
#parsing-xml
#miscellaneous
http://wiki.python.org/moin/PrintFails
```

```
#improving-performance
http://lxml.de/
#parser-installation
http://pypi.python.org/pypi/cchardet/
#parsing-only-part-of-a-document
#translating-this-documentation
#id18
http://www.crummy.com/software/BeautifulSoup/bs3/download/3.x/BeautifulSoup-3.2.0.tar.gz
http://www.crummy.com/software/BeautifulSoup/bs3/documentation.html
#porting-code-to-bs4
http://www.python.org/dev/peps/pep-0008/
#you-need-a-parser
#installing-a-parser
#method-names
#generators
#string-generators
#xml
#entities
#unicode-dammit
#output-formatters
#id19
#string
#multi-valued-attributes
#id13
#id12
#string
#id13
#
#
#getting-help
#quick-start
#installing-beautiful-soup
#problems-after-installation
#installing-a-parser
#making-the-soup
#kinds-of-objects
#tag
#name
#attributes
#multi-valued-attributes
#navigablestring
#beautifulsoup
#comments-and-other-special-strings
#navigating-the-tree
#going-down
#navigating-using-tag-names
#contents-and-children
#descendants
#string
#strings-and-stripped-strings
#going-up
#parent
#parents
#going-sideways
#next-sibling-and-previous-sibling
#next-siblings-and-previous-siblings
#going-back-and-forth
#next-element-and-previous-element
#next-elements-and-previous-elements
#searching-the-tree
#kinds-of-filters
#a-string
#a-regular-expression
#a-list
#true
#a-function
#find-all
#the-name-argument
```

```
#the-keyword-arguments
#searching-by-css-class
#the-string-argument
#the-limit-argument
#the-recursive-argument
#calling-a-tag-is-like-calling-find-all
#find
#find-parents-and-find-parent
#find-next-siblings-and-find-next-sibling
#find-previous-siblings-and-find-previous-sibling
#find-all-next-and-find-next
#find-all-previous-and-find-previous
#css-selectors
#modifying-the-tree
#changing-tag-names-and-attributes
#modifying-string
#append
#extend
#navigablestring-and-new-tag
#insert
#insert-before-and-insert-after
#clear
#extract
#decompose
#replace-with
#wrap
#unwrap
#smooth
#output
#pretty-printing
#non-pretty-printing
#output-formatters
#get-text
#specifying-the-parser-to-use
#differences-between-parsers
#encodings
#output-encoding
#unicode-dammit
#smart-quotes
#inconsistent-encodings
#line-numbers
#comparing-objects-for-equality
#copying-beautiful-soup-objects
#advanced-parser-customization
#parsing-only-part-of-a-document
#soupstrainer
#customizing-multi-valued-attributes
#handling-duplicate-attributes
#instantiating-custom-subclasses
#troubleshooting
#diagnose
#errors-when-parsing-a-document
#version-mismatch-problems
#parsing-xml
#other-parser-problems
#miscellaneous
#improving-performance
#translating-this-documentation
#id18
#porting-code-to-bs4
#you-need-a-parser
#method-names
#generators
#xml
#entities
#id19
_sources/index.rst.txt
genindex.html
#
```

<https://www.sphinx-doc.org/>

Question3. Create a HTML file for the following Student Marks and print the number of students and their names and marks.

In [2]:

```
from IPython.core.display import HTML

que3 = """


|      |       |       |       |       |
|------|-------|-------|-------|-------|
| ID   | NAME  | MARK1 | MARK2 | MARK3 |
| DS01 | REX   | 87    | 57    | 74    |
| DS02 | PETER | 68    | 98    | 55    |


"""

HTML(que3)
```

Out[2]:

ID	NAME	MARK1	MARK2	MARK3
DS01	REX	87	57	74
DS02	PETER	68	98	55

Question4. Create a JSON file for the following Students Marks and print the number of students and their names and marks.

ID: DS01: Name: rex

```
Semester1: 80, 55
Semester2: 50, 70, 82
```

ID: DS02: Name: peter

```
Semester1: 92, 75
```

In [23]:

```
import json
data = """
[
    {"id" : "ds01",
     "name" : "rex",
     "semester1": "80,55",
     "semester2": "50,70,82"
    },
    {"id" : "ds02",
     "name" : "peter",
     "semester1": "92,75",
     "semester2": "75,82,92"
    }
]
```

```

    "semester2": "78,98"
},
{"id" : "ds02",
"name" :"Brent",
"semester1": "35,67",
"semester2": "65,87"
}
]'''
info = json.loads(data)
print('User count:', len(info))
for item in info:
    print('Name', item['name'])
    print('Id', item['id'])
    print('semester1', item['semester1'])
    print('semester2', item['semester2'])

```

```

User count: 2
Name rex
Id ds01
semester1 80,55
semester2 78,98
Name Brent
Id ds02
semester1 35,67
semester2 65,87

```

Question5. Crawl Weather of a City and Display 7 Day Forecast

- Find weather data of Tiruchirappalli city from some website such as www.weather.com
- Exploring page structure with Chrome DevTools
- Extract information from web page and display the weather forecast for 10 days or 7 days
- Reference: <https://www.dataquest.io/blog/web-scraping-tutorial-python/>

In [1]:

```

import requests
from bs4 import BeautifulSoup

page=requests.get("https://weather.com/en-IN/weather/tenday/1/d251a574fa1fa7a38aef6a
page.content

bs4=BeautifulSoup(page.content,'html.parser')
bs4.find_all('p')
(bs4.find_all(class_="DetailsSummary--DetailsSummary--QpFD-"))
days=[bs4.find_all('h2')[day].get_text() for day in range(len(bs4.find_all('h2')))]
temp=[bs4.find_all(class_="DetailsSummary--temperature--3Fm1w")[temp].get_text() fo

days=days[1:12]
temp=temp[1:12]

from IPython.display import display
import pandas as pd
#creating a data frame to store the values
b={"Days": days,
   "Temperature": temp}
weather = pd.DataFrame(b)

weather

```

Out[1]:

Days	Temperature
0 Tonight	34°/22°
1 Sun 14	35°/22°

0	Tonight	34°/22°
1	Sun 14	35°/22°

Days	Temperature
2 Mon 15	35°/21°
3 Tue 16	35°/22°
4 Wed 17	34°/22°
5 Thu 18	33°/22°
6 Fri 19	33°/22°
7 Sat 20	32°/22°
8 Sun 21	34°/21°
9 Mon 22	35°/21°
10 Tue 23	36°/21°

Question6. Real Time Stock Prices Crawling and Display of a specified Company

- Crawl prices of a stock such as InfoSys (Stock Code INFY) for a period of month or year.
- Plot a line graph of monthly or yearly price movements
- Crawl prices of one more stock such as CTS
- Update your line graph with the prices movements of two stocks #### Reference:
- <https://ntguardian.wordpress.com/2018/07/17/stock-data-analysis-python-v2/>
- https://github.com/rajkumarbhc/datascienceutils/blob/master/data_science_utils/financial/WebTutorial.ipynb

In [8]:

```
!pip install yfinance --upgrade --no-cache-dir
import yfinance as yf
yf.pdr_override()
df_info = pdr.get_data_yahoo("INFY", start="2018-01-01").reset_index()
df_info.to_csv('INFY.csv', index=False)
df_info.head()
```

Requirement already satisfied: yfinance in c:\windows\system32\src\yfinance (0.1.55)
Requirement already satisfied: pandas>=0.24 in c:\users\mahesh\anaconda3\lib\site-packages (from yfinance) (1.2.1)
Requirement already satisfied: numpy>=1.15 in c:\users\mahesh\anaconda3\lib\site-packages (from yfinance) (1.19.2)
Requirement already satisfied: requests>=2.20 in c:\users\mahesh\anaconda3\lib\site-packages (from yfinance) (2.25.1)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\mahesh\anaconda3\lib\site-packages (from yfinance) (0.0.9)
Requirement already satisfied: lxml>=4.5.1 in c:\users\mahesh\anaconda3\lib\site-packages (from yfinance) (4.6.2)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\mahesh\anaconda3\lib\site-packages (from pandas>=0.24->yfinance) (2.8.1)
Requirement already satisfied: pytz>=2017.3 in c:\users\mahesh\anaconda3\lib\site-packages (from pandas>=0.24->yfinance) (2021.1)
Requirement already satisfied: six>=1.5 in c:\users\mahesh\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas>=0.24->yfinance) (1.15.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\mahesh\anaconda3\lib\site-packages (from requests>=2.20->yfinance) (1.26.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\mahesh\anaconda3\lib\site-packages (from requests>=2.20->yfinance) (2020.12.5)
Requirement already satisfied: idna<3,>=2.5 in c:\users\mahesh\anaconda3\lib\site-packages (from requests>=2.20->yfinance) (2.10)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\mahesh\anaconda3\lib\site-

```
te-packages (from requests>=2.20->yfinance) (4.0.0)
[*****100%*****] 1 of 1 completed
```

Out[8]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-01-02	8.135	8.195	8.115	8.145	7.429263	12298200
1	2018-01-03	8.120	8.135	8.050	8.075	7.365414	10250800
2	2018-01-04	8.100	8.100	8.010	8.025	7.319807	16272000
3	2018-01-05	8.085	8.190	8.075	8.175	7.456626	9813600
4	2018-01-08	8.190	8.260	8.170	8.240	7.515914	11198200

In [7]:

```
!pip install yfinance --upgrade --no-cache-dir
import yfinance as yf
yf.pdr_override()
df_cts = pdr.get_data_yahoo("CTS", start="2018-01-01").reset_index()
df_cts.to_csv('CTS.csv', index=False)
df_cts.head()
```

```
Requirement already satisfied: yfinance in c:\windows\system32\src\yfinance (0.1.55)
Requirement already satisfied: pandas>=0.24 in c:\users\mahesh\anaconda3\lib\site-packages (from yfinance) (1.2.1)
Requirement already satisfied: numpy>=1.15 in c:\users\mahesh\anaconda3\lib\site-packages (from yfinance) (1.19.2)
Requirement already satisfied: requests>=2.20 in c:\users\mahesh\anaconda3\lib\site-packages (from yfinance) (2.25.1)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\mahesh\anaconda3\lib\site-packages (from yfinance) (0.0.9)
Requirement already satisfied: lxml>=4.5.1 in c:\users\mahesh\anaconda3\lib\site-packages (from yfinance) (4.6.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\mahesh\anaconda3\lib\site-packages (from pandas>=0.24->yfinance) (2021.1)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\mahesh\anaconda3\lib\site-packages (from pandas>=0.24->yfinance) (2.8.1)
Requirement already satisfied: six>=1.5 in c:\users\mahesh\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas>=0.24->yfinance) (1.15.0)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\mahesh\anaconda3\lib\site-packages (from requests>=2.20->yfinance) (4.0.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\mahesh\anaconda3\lib\site-packages (from requests>=2.20->yfinance) (2020.12.5)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\mahesh\anaconda3\lib\site-packages (from requests>=2.20->yfinance) (1.26.3)
Requirement already satisfied: idna<3,>=2.5 in c:\users\mahesh\anaconda3\lib\site-packages (from requests>=2.20->yfinance) (2.10)
[*****100%*****] 1 of 1 completed
```

Out[7]:

	Date	Open	High	Low	Close	Adj Close	Volume
0	2018-01-02	25.900000	26.950001	25.850000	26.500000	26.042004	117900
1	2018-01-03	26.549999	26.700001	26.200001	26.250000	25.796324	62600
2	2018-01-04	26.500000	27.150000	26.450001	26.750000	26.287683	55800
3	2018-01-05	26.750000	27.049999	26.600000	26.850000	26.385950	39700
4	2018-01-08	26.750000	26.900000	26.549999	26.700001	26.238546	40100

In [9]:

```
import matplotlib.pyplot as plt
%matplotlib inline

df_info["Adj Close"].plot(grid = True)
```

Out[9]: <AxesSubplot:>

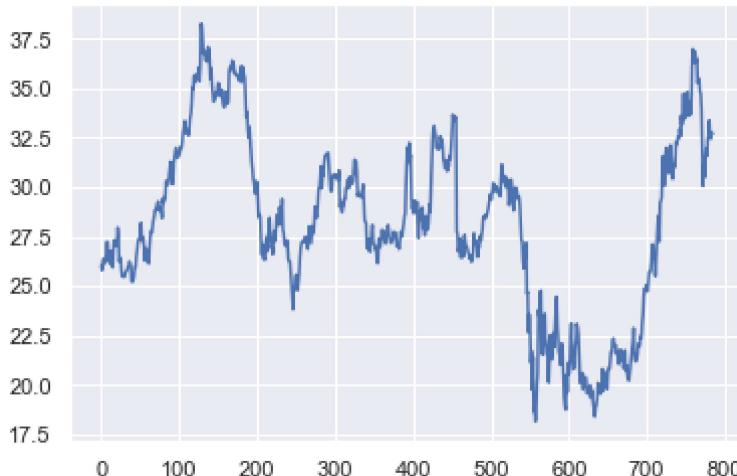


In [10]:

```
import matplotlib.pyplot as plt
%matplotlib inline

df_cts["Adj Close"].plot(grid = True)
```

Out[10]: <AxesSubplot:>



In [11]:

```
s="INFY"
infosys, cts = (pdr.get_data_yahoo(s, start="2018-01-01").reset_index()
                  for s in ["INFY", "CTS"])

stocks = pd.DataFrame({"INFY": df_info["Adj Close"],
                       "CTS": df_cts["Adj Close"]})

stocks.head()
```

[*****100%*****] 1 of 1 completed
[*****100%*****] 1 of 1 completed

Out[11]:

	INFY	CTS
0	7.429263	26.042004
1	7.365414	25.796324
2	7.319807	26.287683
3	7.456626	26.385950
4	7.515914	26.238546

```
In [12]: stocks.plot(grid = True)
```

```
Out[12]: <AxesSubplot:>
```

