SIM Global Education

Topic 5
Retrieving data from the web

Learning Outcomes

After completing this topic and the recommended reading, you should be able to:

- Explain what HTTP is and how the client-server model makes it possible to access data on the internet.
- Implement requests and use them to retrieve data.
- Read data from RESTFul web APIs.

1. HTTP request for Python

- ***Requests*** is a HTTP library for the Python programming language.
- It makes HTTP requests simpler and more human-friendly, allows you to send HTTP requests using Python.

Installing/Importing NumPy Library

- *import requests*

Common Functions

- **get** request to a specified url
  - *page = requests.get("https://www.sim.edu.sg")*

- **status_code** returns a number that indicate the status
  - *page.status_code* *# 200: OK; 404: Not Found*

- **headers** returns the page header information
  - *page.hearders['content-type']* *# text/html; charset=utf-8*

- **encoding** returns the encoding setting of the page
  - *page.encoding* *# utf-8*

- **text** returns the entire hypertext document
  - *page.text*

2. Beautiful Soup

- ***BeautifulSoup*** is a Python package for parsing HTML and XML documents.
- It creates a parse tree for parsed pages that can be used to extract data from HTML and XML, which is useful for web scraping.
  - HTML/XML documents are composed of a tree of tags
- It provides ways of navigating, searching, and modifying parse trees.
- https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Installing/Importing BeautifulSoup Library

- *conda install -c anaconda beautifulsoup4*
- *pip install beautifulsoup4*
- *from bs4 import beautifulsoup*

BeautifulSoup Operations

- Parse the html page
  - *soup = BeautifulSoup(page.text, 'html.parser')*

- Print the prettify version of the html pages (with tabs & line breaks)
  - *print(soup.prettify())*

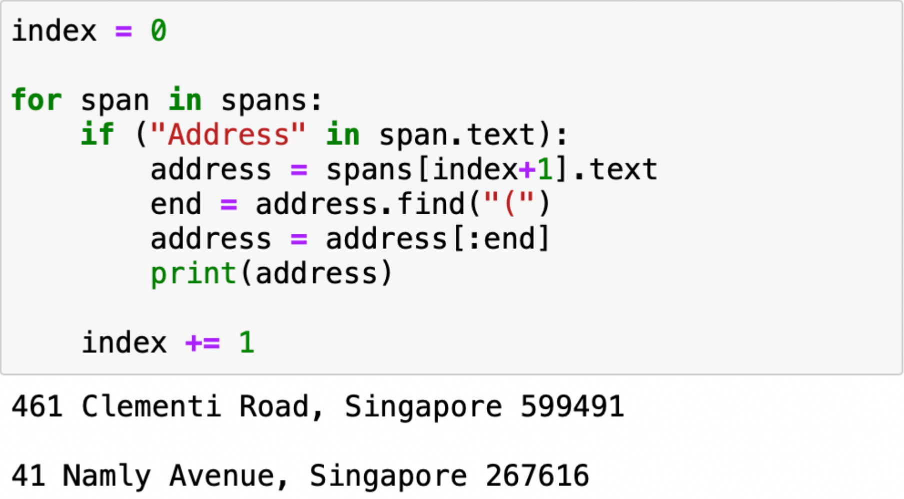
- Show the title of the page
  - *soup.title   # <title>Academic Programmes | Professional*
    # Courses | Enterprise Solutions | SIM</title>

- Locate the footer of the page

- *footer = soup.find('footer')*

- Extract all the specific tags
  - *spans = footer.find_all('span')* *# stored in a list*

- Extract the addresses
  - *Example:*



```
index = 0

for span in spans:
    if ("Address" in span.text):
        address = spans[index+1].text
        end = address.find("(")
        address = address[:end]
        print(address)

    index += 1
```

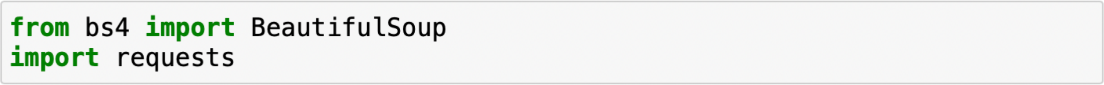
```
461 Clementi Road, Singapore 599491

41 Namly Avenue, Singapore 267616
```

3. Web Scraping Example

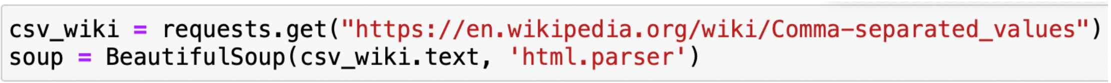
Scraping Text Data into File

- Load the libraries



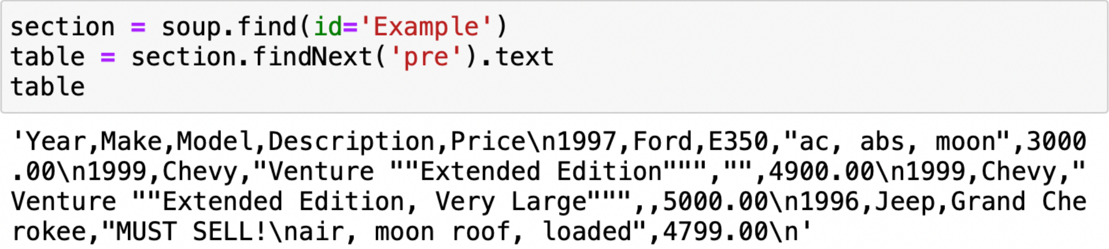
```
from bs4 import BeautifulSoup
import requests
```

- Load in the html



```
csv_wiki = requests.get("https://en.wikipedia.org/wiki/Comma-separated_values")
soup = BeautifulSoup(csv_wiki.text, 'html.parser')
```

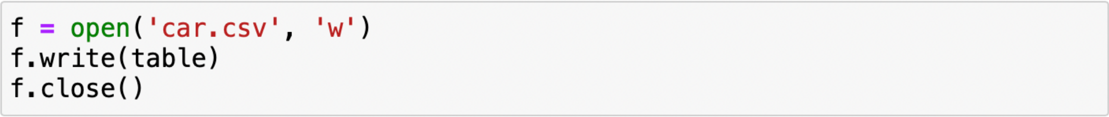
- Get the csv example under the header “Example”



```
section = soup.find(id='Example')
table = section.findNext('pre').text
table
```

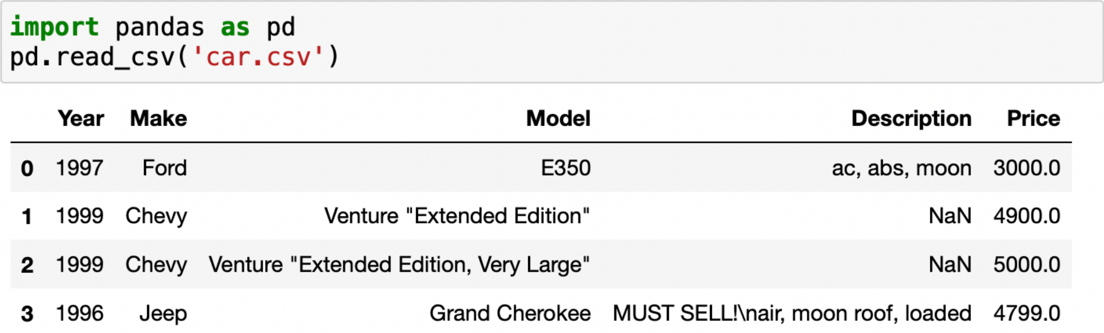
```
'Year,Make,Model,Description,Price\n1997,Ford,E350,"ac, abs, moon",3000
.00\n1999,Chevy,"Venture ""Extended Edition""","",4900.00\n1999,Chevy,"
Venture ""Extended Edition, Very Large""",,5000.00\n1996,Jeep,Grand Che
rokee,"MUST SELL!\nair, moon roof, loaded",4799.00\n'
```

- Save the csv example into a csv file



```
f = open('car.csv', 'w')
f.write(table)
f.close()
```

- Reload the csv data from the file to pandas data frame



```
import pandas as pd
pd.read_csv('car.csv')
```

	Year	Make	Model	Description	Price
0	1997	Ford	E350	ac, abs, moon	3000.0
1	1999	Chevy	Venture "Extended Edition"	NaN	4900.0
2	1999	Chevy	Venture "Extended Edition, Very Large"	NaN	5000.0
3	1996	Jeep	Grand Cherokee	MUST SELL!\nair, moon roof, loaded	4799.0

Scraping Tabular Data into File

- Exercise
- https://olympics.com/en/news/fifa-world-cup-2022-results-scores-football

4. Exercises

5.10 Web Scraping Basics

- Refers to “5.10 Web_Scraping_Basics.html”

5. Practice Quiz

- Work on *Practice Quiz 05* posted on Canvas.

Useful Resources

-
    - http://