5.2 – Working with Data in Files

## **5.2.3 – Importing Data in Python**

Chapter-1: Importing data from the Internet

|  |  |  |
| --- | --- | --- |
| Importing flat files from the web | | |
| automate file download | urlretrieve() | In [1]: from urllib.request import urlretrieve  In [2]: url = 'http://archive.ics.uci.edu/ml/machine-learningdatabases/  wine-quality/winequality-white.csv'  In [3]: urlretrieve(url, 'winequality-white.csv') |
| HTTP requests to import files from the web | | |
| GET requests using urllib | Request()  urlopen()  .read() | In [1]: from urllib.request import urlopen, Request  In [2]: url = "https://www.wikipedia.org/"  In [3]: request = Request(url)  In [4]: response = urlopen(request)  In [5]: html = response.read()  In [6]: response.close() |
| GET requests using requests | requests.get(url)  .text | In [1]: import requests  In [2]: url = "https://www.wikipedia.org/"  In [3]: r = requests.get(url)  In [4]: text = r.text |
| Scraping the web in Python | | |
| BeautifulSoup | BeautifulSoup() | In [1]: r  In [3]: url = 'https://www.crummy.com/software/BeautifulSoup/'  In [4]: r = requests.get(url)  In [5]: html\_doc = r.text  In [6]: soup = BeautifulSoup(html\_doc) |
| Prettified Soup | .prettify() | print(soup.prettify()) |
| Exploring BeautifulSoup | .title  .get\_text() | In [9]: print(soup.title)  In [8]: print(soup.get\_text()) |
| .find\_all()  .get() | for link in soup.find\_all('a'):  print(link.get('href')) |

Chapter-2: Interacting with APIs to import data from the web

|  |  |  |
| --- | --- | --- |
| Introduction to APIs and JSONs | | |
| API |  | Application Programming Interface  ● Protocols and routines |
| Loading JSON | json.load() | In [1]: import json  In [2]: with open('snakes.json', 'r') as json\_file:  json\_data = json.load(json\_file)  In [3]: type(json\_data)  Out[3]: dict  for key, value in json\_data.items():  print(key + ':', value) |
| APIs and interacting with the world wide web | | |
| Connecting to an API in Python | requests.get() | In [1]: import requests  In [2]: url = 'http://www.omdbapi.com/?t=hackers'  In [3]: r = requests.get(url)  In [4]: json\_data = r.json() |
| Query string | .json() | import requests  url='http://www.omdbapi.com/?apikey=ff21610b&t=the+social+network'  r = requests.get(url)  print(r.text) |

Chapter-3: Diving deep into the Twitter API

|  |  |  |
| --- | --- | --- |
| Using Tweepy: Authentication handler |  | import tweepy, json  access\_token = "..."  access\_token\_secret = "..."  consumer\_key = "..."  consumer\_secret = "..."  auth = tweepy.OAuthHandler(consumer\_key, consumer\_secret)  auth.set\_access\_token(access\_token, access\_token\_secret) |
| Tweepy: define stream listener class |  | class MyStreamListener(tweepy.StreamListener):  def \_\_init\_\_(self, api=None):  super(MyStreamListener, self).\_\_init\_\_()  self.num\_tweets = 0  self.file = open("tweets.txt", "w")  def on\_status(self, status):  tweet = status.\_json  self.file.write(json.dumps(tweet) + '\n')  tweet\_list.append(status)  self.num\_tweets += 1  if self.num\_tweets < 100:  return True  else:  return False  self.file.close() |
| Using Tweepy: stream tweets!! |  | # Create Streaming object and authenticate  l = MyStreamListener()  stream = tweepy.Stream(auth, l)  # This line filters Twitter Streams to capture data by keywords:  stream.filter(track=['apples', 'oranges']) |
|  |  |  |