Web Scraping example

This is an example showing how tags from a marked up document are used to harvest data with a specialized package called Beautiful Soup.

In [1]:
```
from bs4 import BeautifulSoup
import pandas as pd
import re
from IPython.display import display
import requests
```

For this sample, I've used a previously saved copy of the HTML page.
The URL for the original content is: http://www.espn.com/nhl/statistics/player/_/stat/points/sort/points/year/2015/seasontype/2 (http://www.espn.com/nhl/statistics/player/_/stat/points/sort/points/year/2015/seasontype/2)

In [2]:
```
url = r'/home/bala/Documents/210x Python-20171127T120640Z-001/210x Python/Module2/2014-15 NHL Hockey Stats and League Leaders - Points - National Hockey League - ESPN.html'
#we'd use the declaration below instead to grab data from a URL
#url = urllib2.urlopen('http://www.espn.com/nhl/statistics/player/_/stat/points/sort/points/year/2015/seasontype/2')
soup = BeautifulSoup(open(url),"html.parser")
display(soup.findAll('tr', limit=2)[1].findAll('td')) #HTML tags: 'tr' = table row, 'td' = table data
```

```
[<td align="left" style="width:20px;">RK</td>,
 <td align="left">PLAYER</td>,
 <td align="left">TEAM</td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/games/year/2015" title="Games Played">GP</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/goals/year/2015" title="Goals">G</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/assists/year/2015" title="Assists">A</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/points/year/2015/order/false" title="Points">PTS</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/plusMinus/year/2015" title="Plus/Minus Rating">+/-</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/penaltyMinutes/year/2015" title="Penalty Minutes">PIM</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/avgPoints/year/2015" title="Points Per Game">PTS/G</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/shotsTotal/year/2015" title="Shots on Goal">SOG</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/shootingPct/year/2015" title="Shooting Percentage">PCT</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/gameWinningGoals/year/2015" title="Game-Winning Goals">GWG</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/powerPlayGoals/year/2015" title="Power-Play Goals">G</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/powerPlayAssists/year/2015" title="Power-Play Assists">A</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/shortHandedGoals/year/2015" title="Short-Handed Goals">G</a></td>,
 <td><a href="//www.espn.com/nhl/statistics/player/_/stat/points/sort/shortHandedAssists/year/2015" title="Short-Handed Assists">A</a></td>]
```

Such marked up content can be loaded into a dataframe as shown below.

In [3]:
```
column_headers = [td.getText() for td in
                  soup.findAll('tr', limit=2)[1].findAll('td')]
player_data_02 = []  # create an empty list to hold all the data
for row in soup('table')[0].findAll('tr')[0:]:
    player_row = []  # create an empty list for each pick/player
    for col in row.findAll('td'):
        thisCell = col.getText()
        player_row.append(thisCell)
    player_data_02.append(player_row)

df = pd.DataFrame(player_data_02, columns=column_headers)
df.drop(df.index[[0,1]], inplace=True)
display(df.head())
```

	RK	PLAYER	TEAM	GP	G	A	PTS	+/-	PIM	PTS/G	SOG	PCT	GWG	G	A	G	A
2	1	Jamie Benn, LW	DAL	82	35	52	87	1	64	1.06	253	13.8	6	10	13	2	3
3	2	John Tavares, C	NYI	82	38	48	86	5	46	1.05	278	13.7	8	13	18	0	1
4	3	Sidney Crosby, C	PIT	77	28	56	84	5	47	1.09	237	11.8	3	10	21	0	0
5	4	Alex Ovechkin, LW	WSH	81	53	28	81	10	58	1.00	395	13.4	11	25	9	0	0
6		Jakub Voracek, RW	PHI	82	22	59	81	1	78	0.99	221	10.0	3	11	22	0	0

Here are a couple of ways ways to search for known content directly. This doesn't go into HTML tags. Just the content between tags and visible on the HTML page.

In [4]:
```
display ('This is an exact match', soup.body.findAll(text='Daniel Sedin'))

display ('This is a contains-target-word match using a regular expression', soup.body.findAll(text=re.compile
('Daniel')))
```

```
'This is an exact match'

['Daniel Sedin']

'This is a contains-target-word match using a regular expression'

['Daniel Sedin']
```

This is one way to obtain meta data like # of ocurrences of a word of interest.

In [5]:
```
searched_word = 'Player'
results = soup.find_all(string=re.compile('.*{0}.*'.format(searched_word)), recursive=True)
display('Count of searched word "{0}" = {1}'.format(searched_word, len(results)))
```

```
'Count of searched word "Player" = 4'
```

Here is one way to read directly a website of publisher O'Reilly.

In [7]:
```
result = requests.get("https://www.oreilly.com/topics/ai")
c = result.content
soup = BeautifulSoup(c,"lxml")
```

Extracting specific tagged content, in this case authors.

In [8]:
```
samples = soup.find_all("span", class_="author")
#samples
for item in samples:
    print(item.a.get_text())
```

```
Andrew Ng
Kenneth O. Stanley
Steven Hewitt
Matt Coatney
Jacob Schreiber
Lukas Biewald
Mike Loukides
Shivon Zilis
James Cham
Justin Francis
Pete Warden
Lukas Biewald
Arthur Juliani
Matt Coatney
```

Extracting specific tagged content, in this case titles.

In [9]:
```
samples_title = soup.find_all("h2", class_="block-title")
for item in samples_title:
    print(item.a.get_text())
```

```
AI is the new electricity
Neuroevolution: A different kind of deep learning
Question answering with TensorFlow
How do I use IBM Watson to extract entity information from news articles?
Deep matrix factorization using Apache MXNet
Build a talking, face-recognizing doorbell for about $100
Planning for AI
The current state of machine intelligence 3.0
Get a hands-on look at how to build bots and conversational apps using NLP
Introduction to reinforcement learning and OpenAI Gym
How to build and run your first deep learning network
How to build a robot that “sees” with $100 and TensorFlow
Reinforcement learning for complex goals, using TensorFlow
How does Facebook recognize my face and the faces of friends and family?
```

In [ ]: