Web Scraping

> Web scraping is a computer software technique of extracting information from websites. Web scraping focuses more on the transformation of unstructured data on the web, typically in HTML format, into structured data that can be stored and analyzed in a central local database or spreadsheet.

HTML (Hyper Text Markup Language)

> Webpages are rendered by the brower from HTML and CSS code, but much of the information included in the HTML underlying any website is not interesting to us.
>
> We begin by reading in the source code for a given web page and creating a Beautiful Soup object with the BeautifulSoup function.

(1) Install BeautifulSoup4

Beautiful Soup is a Python library for pulling data out of HTML and XML files. It works with your favorite parser to provide idiomatic ways of navigating, searching, and modifying the parse tree. It commonly saves programmers hours or days of work.

Detailed documentation on BeautifulSoup4:

https://www.crummy.com/software/BeautifulSoup/bs4/doc/

In [ ]:

```
!pip install BeautifulSoup4
```

In case of ImportError: No module named bs4

For Mac, Open a Terminal, Enter: conda install BeautifulSoup4.

For Win, Open a Command Line Interface (cmd), Enter: conda install BeautifulSoup4.

In [ ]:

```
from bs4 import BeautifulSoup

infile = open('00_00_pokemon_android.html')
contents = infile.read()
infile.close()
soup = BeautifulSoup(contents, 'html.parser')
```

A parser is a compiler or interpreter component that breaks data into smaller elements for easy translation into another language. A parser takes input in the form of a sequence of tokens or program instructions and usually builds a data structure in the form of a parse tree or an abstract syntax tree.

In [ ]:

```
contents
```

In [ ]:

```
print(type(contents))
```

In [ ]:

```
len(contents)
```

In [ ]:

```
print(contents[0:1000])
```

In [ ]:

```
soup
```

In [ ]:

```
print(type(soup))
```

In [ ]:

```
len(soup)
```

(2) Our Sample Target Page

https://www.clemson.edu/science/departments/biosci/directory/index.html

In [ ]:

```
from bs4 import BeautifulSoup
from urllib.request import urlopen
```

In [ ]:

```
infile = urlopen('https://www.clemson.edu/science/departments/biosci/directory/index.htm
contents2 = infile.read()
infile.close()
```

In [ ]:

```
len(contents2)
```

In [ ]:

```
print(contents2[0:1000])
```

In [ ]:

```
print(type(contents2))
```

In [ ]:

```
help(bytes)
```

String vs. Bytes

https://stackoverflow.com/questions/6224052/what-is-the-difference-between-a-string-and-a-byte-string

(3) Convert Bytes Into a BeatifulSoup Object

In [ ]:
```
soup2 = BeautifulSoup(contents2, 'html.parser')
```

The simplest way to navigate the parse tree is to say ***the name of the tag*** you want. If you want the ***<head>*** tag, just say ***soup2.head***:

In [ ]:
```
soup2.title
```

In [ ]:
```
soup2.title.string
```

In [ ]:
```
soup2
```

In [ ]:
```
print(soup2.prettify()[0:1000])

# The prettify() method will turn a Beautiful Soup parse tree into a nicely formatted U
# with each HTML/XML tag on its own line:

# print(soup.prettify()[0:500])
```

Practice #1: Extract the Names and Emails For All Faculty Members

https://www.clemson.edu/science/departments/biosci/directory/index.html

(1) In the Web Browser, Select the Area You Want. Then Inspect to See the Internal HTML Structure

We found out that each professor's information is contained within tr tags. The tr tag defines a row in an HTML table.

In [ ]:
```
soup2.find('tr')
```

The td tag defines a standard cell in an HTML table.

In [ ]:
```
rows = soup2.find_all('tr')
```

In [ ]:
```
print(type(rows))
```

In [ ]:
```
print(len(rows))
```

In [ ]:

```
print(rows[0])
```

In [ ]:

```
print(rows[1])
```

(2) Using find / find_all, We Can Collect the Names and Emails of All Faculty Members

The first argument of find / find_all is the name of the tag.

The second argument of find / find_all is the conditions of tag attribute or string itself.

To call the string, you can add string method after find / find_all.

In [ ]:

```
rows[1].find('td')
```

In [ ]:

```
rows[1].find_all('td')
```

In [ ]:

```
rows[1].find_all('td')[0]
```

In [ ]:

```
rows[1].find('a')
```

In [ ]:

```
rows[1].find_all('a')
```

In [ ]:

```
rows[1].find_all('a')[0]
```

In [ ]:

```
rows[1].find('a', {'href':'https://www.clemson.edu/science/departments/biosci/directory/
```

In [ ]:

```
rows[1].find('a', {'href':'mailto:ja@clemson.edu'})
```

In [ ]:

```
rows[1].find('a', {'href':'mailto:ja@clemson.edu'}).get_text()
```

In [ ]:

```
# import re

name = rows[1].find_all('a')[0].get_text()
print(name)

# import re
# name = rows[1].find('a', {'href': re.compile('profiles')}).get_text(strip = True)
# print(name)
```

In [ ]:

```
# import re

email = rows[1].find_all('a')[1].get_text()
print(email)

# email = rows[1].find('a', {'href': re.compile('mailto:')}).get_text(strip = True)
# print(email)
```

In [ ]:

```
phone = rows[1].find_all('td')[-1].get_text()
print(phone)
```

In [ ]:

```
from pprint import pprint

rows = soup2.find_all('tr')
rows_n = rows[1: ]
name_email = {}
for f in rows_n:
    # pprint(item)
    # print()
    name = f.find_all('a')[0].get_text()
    email = f.find_all('a')[1].get_text()
    name_email[name] = email
```

In [ ]:

```
print(name_email)
```

In [ ]:

```
pprint(name_email)
```

(3) Store the Information Into CSV and JSON

In [ ]:

```
import json

outfile = open('prof_email.json', 'w')
json.dump(name_email, outfile)
outfile.close()
```

In [ ]:

```
outfile = open('prof_email_pretty.json', 'w')
json.dump(name_email, outfile, indent = 4)
outfile.close()
```