Reference

- This is example is from the following video
- https://www.youtube.com/watch?v=zD0FDYI5_rs (https://www.youtube.com/watch?v=zD0FDYI5_rs)
- added some more notes

Python BeautifulSoup Web Scraping

- Learn to scrape data from the web using the Python BeautifulSoup bs4 library.
- BeautifulSoup makes it easy to parse useful data out of an HTML page.
- First install the bs4 library on your system by running at the command line,
- pip install beautifulsoup4 or easy_install beautifulsoup4 (or bs4)
- See BeautifulSoup official documentation for the complete set of functions.
- https://www.crummy.com/software/BeautifulSoup/bs4/doc/ (https://www.crummy.com/software/BeautifulSoup/bs4/doc/)

- Import requests so we can fetch the html content of the webpage
- You can see our example page has about 28k characters.

In [1]:

```
import requests
r = requests.get('https://www.usclimatedata.com/climate/united-states/us
print(len(r.text))
```

```
30580
```

Import BeautifulSoup, and convert your HTML into a bs4 object

- Now we can access specific HTML tags on the page using dot, just like a JSON object.

In [2]:

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(r.text)
print(soup.title)
print(soup.title.string)
```

```
<title>Climate United States - Normals and averages</title>
Climate United States - Normals and averages
```

Drill into the bs4 object to access page contents

- soup.p will give you the contents of the first paragraph tag on the page.
- soup.a gives you anchors / links on the page.
- Get contents of an attribute inside an HTML tag using square brackets and perentheses.
- Use .parent to get the parent object, and .next_sibling to get the next peer object.
- Use your browser's inspect element feature to find the tag for the data you want.

In [3]:

```
print(soup.p)
print(soup.p.text)
print(soup.a)
print(soup.a['title'])
print()
print(soup.p.parent)
```

```
<p class="selection_title">Select a state by name</p>
Select a state by name
<a class="navbar-brand" href="/" title="Temperature - Precipitation - Sunsh
ine - Snowfall"><img alt="Temperature - Precipitation - Sunshine - Snowfal
l" height="34" src="/assets/images/us-climate-data.png" srcset="/assets/ima
ges/us-climate-data.png 1x, /assets/images/us-climate-data-2.png 2x" width
="31"/><span class="white ml-2">U.S. Climate Data</span></a>
Temperature - Precipitation - Sunshine - Snowfall

<div class="float-left mb-4 mt-2"><p class="selection_title">Select a state
by name</p></div>
```

Prettify() is handy for formatted printing

- but note this works only on bs4 objects, not on strings, dicts or lists. For those you need to import pprint.

In [4]:

```
print(soup.p.parent.prettify())
```

```
<div class="float-left mb-4 mt-2">
 <p class="selection_title">
  Select a state by name
 </p>
</div>
```

```
**We need all the state links on this page**

- First we find_all anchor tags, and print out the href attribute, which is the actual link url.
- But we see the result includes some links we don't want, so we need to filter those out.
```

In [5]:

```python
for link in soup.find_all('a'):
    print(link.get('href'))
```

```
/
#
/
/climate/united-states/us
/
/climate/united-states/us
/climate/alabama/united-states/3170
/climate/alaska/united-states/3171
/climate/arizona/united-states/3172
/climate/arkansas/united-states/3173
/climate/california/united-states/3174
/climate/colorado/united-states/3175
/climate/connecticut/united-states/3176
/climate/delaware/united-states/3177
/climate/district-of-columbia/united-states/3178
/climate/florida/united-states/3179
/climate/georgia/united-states/3180
/climate/hawaii/united-states/3181
/climate/idaho/united-states/3182
/climate/illinois/united-states/3183
/climate/indiana/united-states/3184
/climate/iowa/united-states/3185
/climate/kansas/united-states/3186
/climate/kentucky/united-states/3187
/climate/louisiana/united-states/3188
/climate/maine/united-states/3189
/climate/maryland/united-states/1872
/climate/massachusetts/united-states/3191
/climate/michigan/united-states/3192
/climate/minnesota/united-states/3193
/climate/mississippi/united-states/3194
/climate/missouri/united-states/3195
/climate/montana/united-states/919
/climate/nebraska/united-states/3197
/climate/nevada/united-states/3198
/climate/new-hampshire/united-states/3199
/climate/new-jersey/united-states/3200
/climate/new-mexico/united-states/3201
/climate/new-york/united-states/3202
/climate/north-carolina/united-states/3203
/climate/north-dakota/united-states/3204
/climate/ohio/united-states/3205
/climate/oklahoma/united-states/3206
/climate/oregon/united-states/3207
/climate/pennsylvania/united-states/3208
/climate/puerto-rico/united-states/7335
/climate/rhode-island/united-states/3209
/climate/south-carolina/united-states/3210
/climate/south-dakota/united-states/3211
/climate/tennessee/united-states/3212
/climate/texas/united-states/3213
/climate/utah/united-states/3214
/climate/vermont/united-states/3215
```

```
/climate/virginia/united-states/3216
/climate/washington/united-states/3217
/climate/west-virginia/united-states/3218
/climate/wisconsin/united-states/3219
/climate/wyoming/united-states/3220
/climate/washington/district-of-columbia/united-states/usdc0001
https://www.facebook.com/yourweatherservice (https://www.facebook.com/you
rweatherservice)
https://twitter.com/usclimatedata (https://twitter.com/usclimatedata)
/website-info
```

Filter urls using string functions

- We just add an if to check conditions, then add the good ones to a list.
- In the end we get 51 state links, including Washington DC.

In [6]:

```python
base_url = 'https://www.usclimatedata.com'
state_links = []
for link in soup.find_all('a'):
    url = link.get('href')
    if url and '/climate/' in url and '/climate/united-states/us' not in
        state_links.append(url)
print(len(state_links))
```

```
53
```

Test getting the data for one state

- then print the title for that page.

In [12]:

```python
r = requests.get(base_url + state_links[5])
soup = BeautifulSoup(r.text)
print(soup.title.string)
```

```
Climate Colorado - Temperature, Rainfall and Averages
```

The data we need is in tr tags

In [10]:

```python
rows = soup.find_all('tr')
print(len(rows))
```

```
12
```

Filter rows, and add temp data to a list

- We use a list comprehension to filter the rows.
- Then we have only 2 rows left.
- We iterate through those 2 rows, and add all the temps from data cells (td) into a list.

In [14]:

```
rows = [row for row in rows if 'Average high' in str(row)]
print(len(rows))

high_temps = []
for row in rows:
    tds = row.find_all('td')
    for i in range(1,6):
        high_temps.append(tds[i].text)
print(high_temps)
```

```
2
['46', '54', '61', '72', '82', '88', '79', '66', '52', '45']
```

Get the name of the State

- First attempt we just split the title string into a list, and grab the second word.
- But that doesn't work for 2-word states like New York and North Carolina.
- So instead we slice the string from first blank to the hyphen.

In [15]:

```
state = soup.title.string.split()[1]
print(state)
s = soup.title.string
state = s[s.find(' '):s.find('-')].strip()
print(state)
```

```
Colorado
Colorado
```

Add state name and temp list to the data dictionary

- For a single state, this is what our scraped data looks like.
- In this example we only got monthly highs by state, but you could drill into cities, and could get lows and precipitation.

In [19]:

```
data = {}
data[state] = high_temps
print(data)
```

```
{'Washington': ['44', '53', '64', '75', '83', '84', '78', '67', '55', '4
5']}
```

Put it all together and iterate 51 states

- We loop through our 51-state list, and get high temp data for each state, and add it to the data dict.
- This combines all our work above into a single for loop.
- The result is a dict with 51 states and a list of monthly highs for each.

In [20]:

```
data = {}
for state_link in state_links:
    url = base_url + state_link
    r = requests.get(base_url + state_link)
    soup = BeautifulSoup(r.text)
    rows = soup.find_all('tr')
    rows = [row for row in rows if 'Average high' in str(row)]
    high_temps = []
    for row in rows:
        tds = row.find_all('td')
        for i in range(1,6):
            high_temps.append(tds[i].text)
    s = soup.title.string
    state = s[s.find(' '):s.find('-')].strip()
    data[state] = high_temps
print(data)
```

```
{'Alabama': ['58', '67', '74', '82', '88', '91', '85', '75', '65', '56'],
'Alaska': ['27', '34', '44', '56', '63', '64', '55', '40', '28', '25'], 'Ar
izona': ['71', '77', '85', '95', '104', '104', '100', '89', '76', '66'], 'A
rkansas': ['55', '64', '73', '81', '89', '93', '86', '75', '63', '52'], 'Ca
lifornia': ['60', '65', '71', '80', '87', '91', '87', '78', '64', '54'], 'C
olorado': ['46', '54', '61', '72', '82', '88', '79', '66', '52', '45'], 'Co
nnecticut': ['40', '47', '58', '68', '77', '81', '74', '63', '53', '42'],
'Delaware': ['47', '55', '66', '75', '83', '85', '79', '69', '58', '47'],
'District Of Columbia': ['44', '53', '64', '75', '83', '84', '78', '67', '5
5', '45'], 'Florida': ['67', '74', '80', '87', '91', '92', '88', '81', '7
3', '65'], 'Georgia': ['57', '64', '72', '81', '86', '88', '82', '73', '6
4', '54'], 'Hawaii': ['80', '81', '83', '85', '87', '89', '89', '87', '84',
'81'], 'Idaho': ['45', '55', '62', '72', '81', '90', '79', '65', '48', '3
8'], 'Illinois': ['36', '46', '59', '70', '81', '82', '75', '63', '48', '3
6'], 'Indiana': ['40', '51', '63', '73', '82', '83', '77', '65', '52', '3
9'], 'Iowa': ['36', '49', '62', '72', '82', '84', '76', '63', '48', '34'],
'Kansas': ['45', '56', '67', '76', '85', '89', '80', '68', '55', '42'], 'Ke
ntucky': ['45', '55', '66', '75', '83', '86', '79', '68', '55', '44'], 'Lou
isiana': ['65', '72', '78', '85', '89', '91', '87', '80', '72', '64'], 'Mai
ne': ['32', '40', '53', '65', '74', '78', '70', '57', '45', '33'], 'Marylan
d': ['46', '54', '65', '75', '85', '87', '80', '68', '58', '46'], 'Massachu
setts': ['39', '45', '56', '66', '76', '80', '72', '61', '51', '41'], 'Mich
igan': ['33', '44', '58', '69', '78', '80', '73', '60', '47', '34'], 'Minne
sota': ['31', '43', '58', '71', '80', '82', '73', '59', '42', '29'], 'Missi
ssippi': ['60', '69', '76', '83', '89', '92', '87', '77', '67', '58'], 'Mis
souri': ['45', '56', '67', '75', '83', '88', '80', '69', '56', '43'], 'Mont
ana': ['39', '48', '58', '67', '76', '85', '73', '59', '43', '32'], 'Nebras
ka': ['37', '50', '63', '73', '84', '86', '77', '64', '48', '36'], 'Nevad
a': ['50', '57', '63', '71', '81', '88', '80', '68', '54', '45'], 'New Hamp
shire': ['35', '44', '57', '69', '77', '81', '73', '60', '48', '36'], 'New
Jersey': ['42', '51', '62', '72', '82', '84', '77', '65', '55', '44'], 'New
Mexico': ['48', '56', '65', '74', '83', '83', '78', '67', '53', '43'], 'New
York': ['42', '50', '60', '71', '79', '83', '76', '65', '54', '44'], 'North
Carolina': ['55', '63', '72', '79', '86', '87', '81', '72', '62', '53'], 'N
orth Dakota': ['28', '40', '57', '68', '77', '83', '72', '58', '40', '26'],
'Ohio': ['40', '52', '63', '73', '82', '84', '77', '65', '52', '41'], 'Okla
homa': ['55', '63', '72', '80', '88', '93', '85', '73', '62', '51'], 'Orego
n': ['52', '56', '61', '68', '74', '82', '77', '64', '53', '46'], 'Pennsylv
ania': ['44', '53', '64', '74', '83', '85', '78', '67', '56', '45'], 'Puert
o Rico': ['83', '83', '85', '86', '88', '88', '88', '87', '85', '83'], 'Rho
de Island': ['40', '48', '59', '68', '78', '81', '74', '63', '53', '42'],
'South Carolina': ['63', '70', '76', '83', '88', '89', '85', '77', '70', '6
2'], 'South Dakota': ['27', '39', '57', '69', '78', '82', '72', '58', '39',
'25'], 'Tennessee': ['55', '64', '73', '81', '89', '91', '85', '74', '63',
'52'], 'Texas': ['65', '72', '80', '87', '92', '97', '91', '82', '71', '6
3'], 'Utah': ['44', '53', '61', '71', '82', '89', '78', '65', '50', '40'],
'Vermont': ['31', '40', '55', '67', '76', '79', '70', '57', '46', '33'], 'V
irginia': ['51', '60', '70', '78', '86', '88', '81', '71', '61', '51'], 'Wa
shington': ['44', '53', '64', '75', '83', '84', '78', '67', '55', '45'], 'W
est Virginia': ['47', '56', '68', '75', '82', '84', '78', '68', '57', '4
6'], 'Wisconsin': ['33', '42', '54', '65', '75', '78', '71', '59', '46', '3
3'], 'Wyoming': ['40', '47', '55', '65', '75', '81', '72', '59', '47', '3
8']}
```

Save to CSV file

- Lastly, we might want to write all this data to a CSV file.
- Here's a quick easy way to do that.

In [22]:

```python
import csv

with open('data/high_temps.csv','w') as f:
    w = csv.writer(f)
    w.writerows(data.items())
```

In [ ]:

```
 
```