Web Scraping with Python using Beautiful Soup

- The internet is an absolutely massive source of data- data that we can access using web scraping and Python!
- In fact, web scraping is often the only way we can access data

In [3]:

```
import requests
page = requests.get("https://dataquestio.github.io/web-scraping-pages/simple.html")
```

In [4]:

```
page
```

Out[4]:

```
<Response [200]>
```

In [5]:

```
page.status_code
```

Out[5]:

```
200
```

In [6]:

```
page.content
```

Out[6]:

```
b'<!DOCTYPE html>\n<html>\n    <head>\n        <title>A simple exa
mple page</title>\n    </head>\n    <body>\n        <p>Here is som
e simple content for this page.</p>\n    </body>\n</html>'
```

In [8]:

```
!pip install beautifulsoup4
```

```
Requirement already satisfied: beautifulsoup4 in /Users/venkatasa
i/opt/anaconda3/lib/python3.7/site-packages (4.8.2)
Requirement already satisfied: soupsieve>=1.2 in /Users/venkatasa
i/opt/anaconda3/lib/python3.7/site-packages (from beautifulsoup4)
(1.9.5)
```

In [9]:

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(page.content)
```

In [10]:

```
print(soup.prettify())
```

```
<!DOCTYPE html>
<html>
 <head>
  <title>
   A simple example page
  </title>
 </head>
 <body>
  <p>
   Here is some simple content for this page.
  </p>
 </body>
</html>
```

In [11]:
```
list(soup.children)
```

Out[11]:
```
['html',
 <html>
 <head>
 <title>A simple example page</title>
 </head>
 <body>
 <p>Here is some simple content for this page.</p>
 </body>
 </html>]
```

In [15]:
```
html = list(soup.children)[1]
list(html.children)
```

Out[15]:
```
['\n',
 <head>
 <title>A simple example page</title>
 </head>,
 '\n',
 <body>
 <p>Here is some simple content for this page.</p>
 </body>,
 '\n']
```

In [17]:
```
body = list(html.children)[3]
```

In [19]:
```
list(body.children)
```

Out[19]:
```
['\n', <p>Here is some simple content for this page.</p>, '\n']
```

In [20]:
```
p = list(body.children)[1]
```

In [21]:
```
p
```

Out[21]:
```
<p>Here is some simple content for this page.</p>
```

In [22]:
```
p.get_text()
```

Out[22]:
```
'Here is some simple content for this page.'
```

In [24]:
```
soup = BeautifulSoup(page.content, 'html.parser')
soup.find_all('p')
```

Out[24]:
```
[<p>Here is some simple content for this page.</p>]
```

In [25]:
```
soup.find_all('p')[0].get_text()
```

Out[25]:
```
'Here is some simple content for this page.'
```

In [26]:
```
page = requests.get("https://dataquestio.github.io/web-scraping-pages/ids_and_classes.html")
soup = BeautifulSoup(page.content, 'html.parser')
```

In [27]:

```
soup
```

Out[27]:

```
<html>
<head>
<title>A simple example page</title>
</head>
<body>
<div>
<p class="inner-text first-item" id="first">
                First paragraph.
            </p>
<p class="inner-text">
                Second paragraph.
            </p>
</div>
<p class="outer-text first-item" id="second">
<b>
                First outer paragraph.
            </b>
</p>
<p class="outer-text">
<b>
                Second outer paragraph.
            </b>
</p>
</body>
</html>
```

In [28]:

```
soup.find_all('p', class_='outer-text')
```

Out[28]:

```
[<p class="outer-text first-item" id="second">
 <b>
                 First outer paragraph.
             </b>
 </p>,
 <p class="outer-text">
 <b>
                 Second outer paragraph.
             </b>
 </p>]
```

In [29]:

```
soup.find_all(class_='outer-text')
```

Out[29]:

```
[<p class="outer-text first-item" id="second">
 <b>
                 First outer paragraph.
             </b>
 </p>,
 <p class="outer-text">
 <b>
                 Second outer paragraph.
             </b>
 </p>]
```

In [30]:
```
soup.find_all(id="first")
```

Out[30]:
```
[<p class="inner-text first-item" id="first">
                 First paragraph.
             </p>]
```

In [31]:
```
page = requests.get("https://forecast.weather.gov/MapClick.php?lat=4
0.71455000000003&lon=-74.00713999999994#.Yjk2UxBBy3I")

soup = BeautifulSoup(page.content)

seven_day = soup.find(id="seven-day-forecast")

forecast_items = seven_day.find_all(class_="tombstone-container")
```

In [33]:
```
tonight = forecast_items[0]
```

In [34]:
```
print(tonight.prettify())
```

```
<div class="tombstone-container">
 <p class="period-name">
  Tonight
  <br/>
  <br/>
 </p>
 <p>
  <img alt="Tonight: A 20 percent chance of showers after 2am.  Mo
stly cloudy, with a low around 45. Northwest wind around 8 mph. "
class="forecast-icon" src="DualImage.php?i=nbkn&amp;j=nshra&amp;jp
=20" title="Tonight: A 20 percent chance of showers after 2am.  Mo
stly cloudy, with a low around 45. Northwest wind around 8 mph. "/
>
 </p>
 <p class="short-desc">
  Mostly Cloudy
  <br/>
  then Slight
  <br/>
  Chance
  <br/>
  Showers
 </p>
 <p class="temp temp-low">
  Low: 45 °F
 </p>
</div>
```

In [40]:
```
period = tonight.find(class_="period-name").get_text()
short_desc = tonight.find(class_="short-desc").get_text()
temp = tonight.find(class_="temp").get_text()
print(period)
print(short_desc)
print(temp)
```

```
Tonight
Mostly Cloudythen SlightChanceShowers
Low: 45 °F
```

In [45]:
```
period_tags = seven_day.select(".tombstone-container .period-name")
periods = [pt.get_text() for pt in period_tags]
```

In [44]:
```
periods
```

Out[44]:
```
['Tonight',
 'Tuesday',
 'TuesdayNight',
 'Wednesday',
 'WednesdayNight',
 'Thursday',
 'ThursdayNight',
 'Friday',
 'FridayNight']
```

In [46]:
```
short_descs = [sd.get_text() for sd in seven_day.select(".tombstone-container .short-desc")]
temps = [t.get_text() for t in seven_day.select(".tombstone-container .temp")]
```

In [47]:
```
import pandas as pd

weather = pd.DataFrame({"period": periods, "short_desc": short_descs,
                        "temp": temps})
```

In [48]:
```
weather
```

Out[48]:

	period	short_desc	temp
0	Tonight	Mostly Cloudythen SlightChanceShowers	Low: 45 °F
1	Tuesday	Mostly Sunny	High: 58 °F
2	TuesdayNight	Partly Cloudy	Low: 41 °F
3	Wednesday	Mostly Cloudythen ChanceRain	High: 44 °F
4	WednesdayNight	Rain	Low: 45 °F⇑
5	Thursday	Rain	High: 54 °F
6	ThursdayNight	Rain Likely	Low: 50 °F
7	Friday	Chance Rain	High: 56 °F
8	FridayNight	Partly Cloudy	Low: 45 °F

In [49]:

```
def weather_forecast(url):

    page = requests.get(url)

    soup = BeautifulSoup(page.content)

    seven_day = soup.find(id="seven-day-forecast")

    period_tags = seven_day.select(".tombstone-container .period-name")
    periods = [pt.get_text() for pt in period_tags]

    short_descs = [sd.get_text() for sd in seven_day.select(".tombstone-container .short-desc")]
    temps = [t.get_text() for t in seven_day.select(".tombstone-container .temp")]

    weather_df = pd.DataFrame({"period": periods, "short_desc": short_descs,
                               "temp": temps})

    return weather_df
```

In [50]:

```
weather_forecast("https://forecast.weather.gov/MapClick.php?lat=40.4131&lon=-82.7112#.Yjk6JxBBy3I")
```

Out[50]:

	period	short_desc	temp
0	Tonight	IncreasingClouds	Low: 46 °F
1	Tuesday	ShowersLikely	High: 58 °F
2	TuesdayNight	ShowersLikely	Low: 51 °F
3	Wednesday	Showers	High: 65 °F
4	WednesdayNight	ChanceT-storms thenChanceShowers	Low: 49 °F
5	Thursday	ChanceShowers	High: 55 °F
6	ThursdayNight	ChanceShowers	Low: 40 °F
7	Friday	ChanceShowers	High: 46 °F
8	FridayNight	Mostly Cloudy	Low: 37 °F

In [ ]: