A tutorial to scrape the web.

This example scrapes the BBC weather website for any specific city, and collects weather forecast for the next 14 days and saves it as a csv file.

Web scraping might not be legal always. It is a good idea to check the terms of the website you plan to scrape before proceeding. Also, if your code requests a url from a server multiple times, it is a good practice to either cache your requests, or insert a timed delay between consecutive requests.

In [ ]:

```
import json                     # to convert API to json format

from urllib.parse import urlencode

import requests                 # to get the webpage
from bs4 import BeautifulSoup # to parse the webpage

import pandas as pd
import re                       # regular expression operators

from datetime import datetime
```

We now GET the webpage of interest, from the server

In [ ]:

```
required_city = "Mumbai"
location_url = 'https://locator-service.api.bbci.co.uk/locations?' + urlencode({
   'api_key': 'AGbFAKx58hyjQScCXIYrxuEwJh2W2cmv',
   's': required_city,
   'stack': 'aws',
   'locale': 'en',
   'filter': 'international',
   'place-types': 'settlement,airport,district',
   'order': 'importance',
   'a': 'true',
   'format': 'json'
})
location_url
```

Out[ ]:

```
'https://locator-service.api.bbci.co.uk/locations?api_key=AGbFAKx58hyjQScCXIYrxuEwJh2W2cm
v&s=Mumbai&stack=aws&locale=en&filter=international&place-types=settlement%2Cairport%2Cdi
strict&order=importance&a=true&format=json'
```

In [ ]:

```
result = requests.get(location_url).json()
result
```

Out[ ]:

```
{'response': {'results': {'results': [{'container': 'India',
     'containerId': 1269750,
     'country': 'IN',
     'id': '1275339',
     'language': 'en',
     'latitude': 19.07283,
     'longitude': 72.88261,
     'name': 'Mumbai',
     'placeType': 'settlement',
     'timezone': 'Asia/Kolkata'}],
   'totalResults': 1}}}
```

In [ ]:

```
# url      = 'https://www.bbc.com/weather/1275339' # url to BBC weather, corresponding to a specific city (Mumbai, in this example)
url      = 'https://www.bbc.com/weather/'+result['response']['results']['results'][0]['id']
response = requests.get(url)
```

Next, we initiate an instance of BeautifulSoup.

In [ ]:

```
soup = BeautifulSoup(response.content,'html.parser')
```

The information we want (daily high and low temp., and daily weather summary), are in specific blocks on the webpage. We need to find the block type, type of identifier, and the identifier name (all these can be figured out by right clicking on the webpage and selecting 'Inspect' on the Chrome browser; similar modus operandi for other browsers)

In [ ]:

```
daily_high_values = soup.find_all('span', attrs={'class': 'wr-day-temperature__high-value'}) # block-type: span; identifier type: class; and class name: wr-day-temperature__high-value
daily_high_values
```

Out[ ]:

```
[<span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">32°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">90°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">33°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">92°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">33°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">91°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">33°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">92°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">33°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">92°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">32°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">90°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">31°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">88°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">32°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">89°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">32°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">89°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">32°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">89°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">33°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">91°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">33°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">91°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">32°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">90°</span></span></span>,
 <span class="wr-day-temperature__high-value"><span class="wr-value--temperature "><span class="wr-value--temperature--c">32°</span><span class="wr-hide"> </span><span class="wr-value--temperature--f">89°</span></span></span>]
```

In [ ]:

```
daily_low_values  = soup.find_all('span', attrs={'class': 'wr-day-temperature__low-value
'})
daily_low_values
```

Out[ ]:

```
[<span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">25°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">77°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">25°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">78°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">26°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">79°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">27°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">80°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">26°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">78°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">26°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">78°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">25°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">78°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">25°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">78°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">25°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">78°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">25°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">78°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">25°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">78°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">25°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">78°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">25°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">77°</span></span></span>,
 <span class="wr-day-temperature__low-value"><span class="wr-value--temperature "><span c
lass="wr-value--temperature--c">25°</span><span class="wr-hide"> </span><span class="wr-v
alue--temperature--f">76°</span></span></span>]
```

In [ ]:

```
daily_summary = soup.find('div', attrs={'class': 'wr-day-summary'})
daily_summary
```

Out[ ]:

```
<div class="wr-day-summary"><div class="gel-wrap"><span class="">Sunny intervals and a ge
ntle breeze</span><span class="wr-hide">Sunny intervals and a gentle breeze</span><span c
lass="wr-hide">Sunny intervals and a gentle breeze</span><span class="wr-hide">Sunny inte
rvals and a gentle breeze</span><span class="wr-hide">Sunny intervals and a gentle breeze
</span><span class="wr-hide">Light cloud and a gentle breeze</span><span class="wr-hide">
Light cloud and a gentle breeze</span><span class="wr-hide">Light cloud and a gentle bree
ze</span><span class="wr-hide">Light cloud and a gentle breeze</span><span class="wr-hide
">Light cloud and a gentle breeze</span><span class="wr-hide">Sunny intervals and a gentl
e breeze</span><span class="wr-hide">Sunny intervals and a gentle breeze</span><span clas
s="wr-hide">Sunny intervals and a gentle breeze</span><span class="wr-hide">Light cloud a
nd a gentle breeze</span></div></div>
```

In [ ]:

In [ ]:

```
daily_summary.text
```

Out[ ]:

```
'Sunny intervals and a gentle breezeSunny intervals and a gentle breezeSunny intervals an
d a gentle breezeSunny intervals and a gentle breezeSunny intervals and a gentle breezeLi
ght cloud and a gentle breezeLight cloud and a gentle breezeLight cloud and a gentle bree
zeLight cloud and a gentle breezeLight cloud and a gentle breezeSunny intervals and a gen
tle breezeSunny intervals and a gentle breezeSunny intervals and a gentle breezeLight clo
ud and a gentle breeze'
```

General book keeping.

With the code snippet in the cell above, we get forecast data for 14 days, including today. We will now post process the data to first extract the required information/text and discard all the html wrapper code, then combine all variables into one common list, and finally convert it into a pandas data frame.

In [ ]:

```
daily_high_values[0].text.strip()
```

Out[ ]:

```
'32° 90°'
```

In [ ]:

```
daily_high_values[5].text.strip()
```

Out[ ]:

```
'32° 90°'
```

In [ ]:

```
daily_high_values[0].text.strip().split()[0]
```

Out[ ]:

```
'32°'
```

In [ ]:

```
daily_high_values_list = [daily_high_values[i].text.strip().split()[0] for i in range(le
n(daily_high_values))]
daily_high_values_list
```

Out[ ]:

```
['32°',
 '33°',
 '33°',
 '33°',
 '33°',
 '32°',
 '31°',
 '32°',
 '32°',
 '32°',
 '33°',
 '33°',
 '32°',
 '32°']
```

In [ ]:

```
daily_low_values_list = [daily_low_values[i].text.strip().split()[0] for i in range(len(
daily_low_values))]
daily_low_values_list
```

Out[ ]:

```
['25°',
 '25°',
 '26°',
 '27°',
 '26°',
 '26°',
 '25°',
 '25°',
 '25°',
 '25°',
 '25°',
 '25°',
 '25°',
 '25°']
```

In [ ]:

```
daily_summary.text
```

Out[ ]:

```
'Sunny intervals and a gentle breezeSunny intervals and a gentle breezeSunny intervals an
d a gentle breezeSunny intervals and a gentle breezeSunny intervals and a gentle breezeLi
ght cloud and a gentle breezeLight cloud and a gentle breezeLight cloud and a gentle bree
zeLight cloud and a gentle breezeLight cloud and a gentle breezeSunny intervals and a gen
tle breezeSunny intervals and a gentle breezeSunny intervals and a gentle breezeLight clo
ud and a gentle breeze'
```

In [ ]:

```
daily_summary_list = re.findall('[a-zA-Z][^A-Z]*', daily_summary.text) #split the string
on uppercase
daily_summary_list
```

Out[ ]:

```
['Sunny intervals and a gentle breeze',
 'Sunny intervals and a gentle breeze',
 'Sunny intervals and a gentle breeze',
 'Sunny intervals and a gentle breeze',
 'Sunny intervals and a gentle breeze',
 'Light cloud and a gentle breeze',
 'Light cloud and a gentle breeze',
 'Light cloud and a gentle breeze',
 'Light cloud and a gentle breeze',
 'Light cloud and a gentle breeze',
 'Sunny intervals and a gentle breeze',
 'Sunny intervals and a gentle breeze',
 'Sunny intervals and a gentle breeze',
 'Light cloud and a gentle breeze']
```

In [ ]:

```
datelist = pd.date_range(datetime.today(), periods=len(daily_high_values)).tolist()
datelist
```

Out[ ]:

```
[Timestamp('2021-10-03 02:17:31.148587', freq='D'),
 Timestamp('2021-10-04 02:17:31.148587', freq='D'),
 Timestamp('2021-10-05 02:17:31.148587', freq='D'),
 Timestamp('2021-10-06 02:17:31.148587', freq='D'),
 Timestamp('2021-10-07 02:17:31.148587', freq='D'),
 Timestamp('2021-10-08 02:17:31.148587', freq='D'),
 Timestamp('2021-10-09 02:17:31.148587', freq='D'),
 Timestamp('2021-10-10 02:17:31.148587', freq='D'),
 Timestamp('2021-10-11 02:17:31.148587', freq='D'),
 Timestamp('2021-10-12 02:17:31.148587', freq='D'),
 Timestamp('2021-10-13 02:17:31.148587', freq='D'),
 Timestamp('2021-10-14 02:17:31.148587', freq='D'),
 Timestamp('2021-10-15 02:17:31.148587', freq='D'),
 Timestamp('2021-10-16 02:17:31.148587', freq='D')]
```

In [ ]:

```
datelist = [datelist[i].date().strftime('%y-%m-%d') for i in range(len(datelist))]
datelist
```

Out[ ]:

```
['21-10-03',
 '21-10-04',
 '21-10-05',
 '21-10-06',
 '21-10-07',
 '21-10-08',
 '21-10-09',
 '21-10-10',
 '21-10-11',
 '21-10-12',
 '21-10-13',
 '21-10-14',
 '21-10-15',
 '21-10-16']
```

In [ ]:

```
zipped = zip(datelist, daily_high_values_list, daily_low_values_list, daily_summary_list)
```

In [ ]:

```
df = pd.DataFrame(list(zipped), columns=['Date', 'High','Low', 'Summary'])
```

In [ ]:

```
display(df)
```

	Date	High	Low	Summary
0	21-10-03	32°	25°	Sunny intervals and a gentle breeze
1	21-10-04	33°	25°	Sunny intervals and a gentle breeze
2	21-10-05	33°	26°	Sunny intervals and a gentle breeze
3	21-10-06	33°	27°	Sunny intervals and a gentle breeze
4	21-10-07	33°	26°	Sunny intervals and a gentle breeze
5	21-10-08	32°	26°	Light cloud and a gentle breeze
6	21-10-09	31°	25°	Light cloud and a gentle breeze
7	21-10-10	32°	25°	Light cloud and a gentle breeze
8	21-10-11	32°	25°	Light cloud and a gentle breeze
9	21-10-12	32°	25°	Light cloud and a gentle breeze
10	21-10-13	33°	25°	Sunny intervals and a gentle breeze
11	21-10-14	33°	25°	Sunny intervals and a gentle breeze
12	21-10-15	32°	25°	Sunny intervals and a gentle breeze
13	21-10-16	32°	25°	Light cloud and a gentle breeze

In [ ]:

```
# remove the 'degree' character
df.High = df.High.replace('\°','',regex=True).astype(float)
df.Low  = df.Low.replace('\°','',regex=True).astype(float)
```

In [ ]:

```
display(df)
```

	Date	High	Low	Summary

	Date	High	Low	Summary
0	21-10-03	32.0	25.0	Sunny intervals and a gentle breeze
1	21-10-04	33.0	25.0	Sunny intervals and a gentle breeze
2	21-10-05	33.0	26.0	Sunny intervals and a gentle breeze
3	21-10-06	33.0	27.0	Sunny intervals and a gentle breeze
4	21-10-07	33.0	26.0	Sunny intervals and a gentle breeze
5	21-10-08	32.0	26.0	Light cloud and a gentle breeze
6	21-10-09	31.0	25.0	Light cloud and a gentle breeze
7	21-10-10	32.0	25.0	Light cloud and a gentle breeze
8	21-10-11	32.0	25.0	Light cloud and a gentle breeze
9	21-10-12	32.0	25.0	Light cloud and a gentle breeze
10	21-10-13	33.0	25.0	Sunny intervals and a gentle breeze
11	21-10-14	33.0	25.0	Sunny intervals and a gentle breeze
12	21-10-15	32.0	25.0	Sunny intervals and a gentle breeze
13	21-10-16	32.0	25.0	Light cloud and a gentle breeze

Extract the name of the city for which data is gathered.

In [ ]:

```
#location = soup.find('div', attrs={'class':'wr-c-location'})
location = soup.find('h1', attrs={'id':'wr-location-name-id'})
location.text.split()
```

Out[ ]:

```
['Mumbai', '-', 'Weather', 'warnings', 'issued']
```

In [ ]:

```
# create a recording
filename_csv = location.text.split()[0]+'.csv'
df.to_csv(filename_csv, index=None)
```

In [ ]:

```
filename_xlsx = location.text.split()[0]+'.xlsx'
df.to_excel(filename_xlsx)
```

In [ ]:

```
```