**Weather Data**

**Source:** **https://www.wunderground.com/**

Weather Underground is a company that collects and distributes data on various weather measurements around the globe.

​BeautifulSoup library is used in the Python Script to scrape the data from the table in the web page and save it into a CSV file

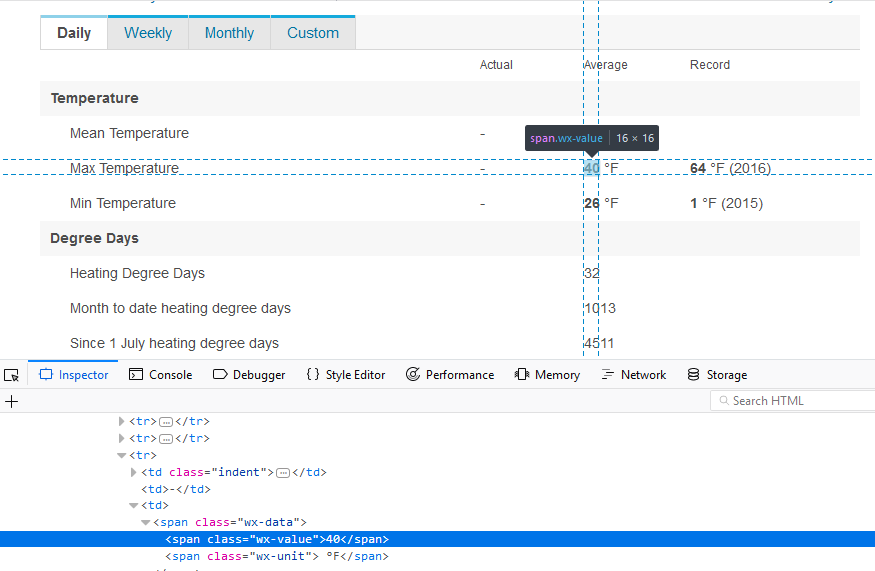
The nearest weather station for 60601 to 60607 is Midway Airport. So the URL is

"http://www.wunderground.com/history/airport/KMDW/" + str(YY) + "/" + str(MM) + "/" + str(DD) + "/DailyHistory.html"

Here the weather is noted for everyday and this data is collected for required range of years.

The date format-YY/MM/DD is looped through the range of data required, using FOR loop.

HTML table is divided into rows with the <tr> tag, and each row is divided into data cells with the <td> tag.



If there is a valid text value under td and tr ,then value is assigned to variable else “N/A” is assigned

Python code:

from bs4 import BeautifulSoup

import urllib.request

with open("Chicago Weather Data2017.csv", "wb") as f:

f.write(

b"Date,Mean Temperature,Max Temperature,Min Temperature,Heating Degree Days, Dew Point, Average Humidity, Max Humidity, Minimum Humidity, Precipitation, Sea Level Pressure, Average Wind Speed, Maximum Wind Speed, Visibility, Events" + b"\n")

for YY in range(2017, 2018):

for MM in range(1, 12):

for DD in range(1, 32):

url = "http://www.wunderground.com/history/airport/KMDW/" + str(YY) + "/" + str(MM) + "/" + str(DD) + "/DailyHistory.html"

req = urllib.request.urlopen(url)

soup = BeautifulSoup(req, "html.parser")

Min = Max = Mean= MaxWindSpeed = Visibility = Events = AvgWindSpeed = SeaLevelPressure = Precipitation = MinHumidity = MaxHumidity = AvgHumidity = DewPoint = HeatingDegreeDays = "N/A"

for data in soup.find\_all('tr'):

if data.text.strip().replace('\n', '')[:6] == 'Actual' or data.text.strip().replace('\n', '')[-6:] == "Record":

pass

elif data.text.replace('\n', '')[-7:] == "RiseSet":

break

elif data.find\_all('td')[0].text == "Mean Temperature":

if data.find\_all('td')[1].text.strip() == "-":

Mean = "N/A"

else:

Mean = data.find\_all('td')[1].find(attrs={"class": "wx-value"}).text

elif data.find\_all('td')[0].text == "Max Temperature":

if data.find\_all('td')[1].text.strip() == "-":

Max = "N/A"

else:

Max = data.find\_all('td')[1].find(attrs={"class": "wx-value"}).text

elif data.find\_all('td')[0].text == "Min Temperature":

if data.find\_all('td')[1].text.strip() == "-":

Min = "N/A"

else:

Min = data.find\_all('td')[1].find(attrs={"class": "wx-value"}).text

elif data.find\_all('td')[0].text == "Growing Degree Days":

if data.find\_all('td')[1].text.strip() == "-":

GrowingDegreeDays = "N/A"

else:

GrowingDegreeDays = data.find\_all('td')[1].text

elif data.find\_all('td')[0].text == "Heating Degree Days":

if data.find\_all('td')[1].text.strip() == "-":

HeatingDegreeDays = "N/A"

else:

HeatingDegreeDays = data.find\_all('td')[1].text

elif data.find\_all('td')[0].text == "Dew Point":

if data.find\_all('td')[1].text.strip() == "-" or data.find\_all('td')[1].text.strip() == "":

DewPoint = "N/A"

else:

DewPoint = data.find\_all('td')[1].find(attrs={"class": "wx-value"}).text

elif data.find\_all('td')[0].text == "Average Humidity":

if data.find\_all('td')[1].text.strip() == "-" or data.find\_all('td')[1].text.strip() == "":

AvgHumidity = "N/A"

else:

AvgHumidity = data.find\_all('td')[1].text

elif data.find\_all('td')[0].text == "Maximum Humidity":

if data.find\_all('td')[1].text.strip() == "-" or data.find\_all('td')[1].text.strip() == "":

MaxHumidity = "N/A"

else:

MaxHumidity = data.find\_all('td')[1].text

elif data.find\_all('td')[0].text == "Minimum Humidity":

if data.find\_all('td')[1].text.strip() == "-" or data.find\_all('td')[1].text.strip() == "":

MinHumidity = "N/A"

else:

MinHumidity = data.find\_all('td')[1].text

elif data.find\_all('td')[0].text == "Precipitation" and data.find\_all('td')[1].text.strip() != "":

if data.find\_all('td')[1].text.strip() == "-" or data.find\_all('td')[1].text.strip() == "":

Precipitation = "N/A"

else:

Precipitation = data.find\_all('td')[1].find(attrs={"class": "wx-value"}).text

elif data.find\_all('td')[0].text == "Sea Level Pressure" and data.find\_all('td')[1].text.strip() != "":

if data.find\_all('td')[1].text.strip() == "-":

SeaLevelPressure = "N/A"

else:

SeaLevelPressure = data.find\_all('td')[1].find(attrs={"class": "wx-value"}).text

elif data.find\_all('td')[0].text == "Wind Speed":

if data.find\_all('td')[1].text.strip() == "-" or data.find\_all('td')[1].text.strip().replace('\n','') == "- ()" or data.find\_all('td')[1].text.strip() == "":

AvgWindSpeed = "N/A"

elif data.find\_all('td')[0].text == "Max Wind Speed":

if data.find\_all('td')[1].text.strip() == "-" or data.find\_all('td')[1].text.strip() == "":

MaxWindSpeed = "N/A"

else:

MaxWindSpeed = data.find\_all('td')[1].find(attrs={"class": "wx-value"}).text

elif data.find\_all('td')[0].text == "Visibility":

if data.find\_all('td')[1].text.strip() == "-":

Visibility = "N/A"

else:

Visibility = data.find\_all('td')[1].find(attrs={"class": "wx-value"}).text

elif data.find\_all('td')[0].text == "Events":

if data.find\_all('td')[1].text.strip() == "-":

Events = "N/A"

else:

Events = data.find\_all('td')[1].text.strip().replace(",", " ").replace('\n', '').replace('\t','')

break

csvdata = (str(YY) + "/" + str(MM) + "/" + str(DD)) + "," + Mean + "," + Max + "," + Min + "," + HeatingDegreeDays + "," + DewPoint + "," + AvgHumidity + "," + MaxHumidity + "," + MinHumidity + "," + Precipitation + "," + SeaLevelPressure + "," + AvgWindSpeed + "," + MaxWindSpeed + "," + Visibility + "," + Events + "\n"

#csv data is written into the file

f.write(bytes(csvdata, encoding="ascii", errors='ignore'))