

Jeanine Carhart

CEIS 110 | FINAL PROJECT | JUNE 25, 2022

Introduction

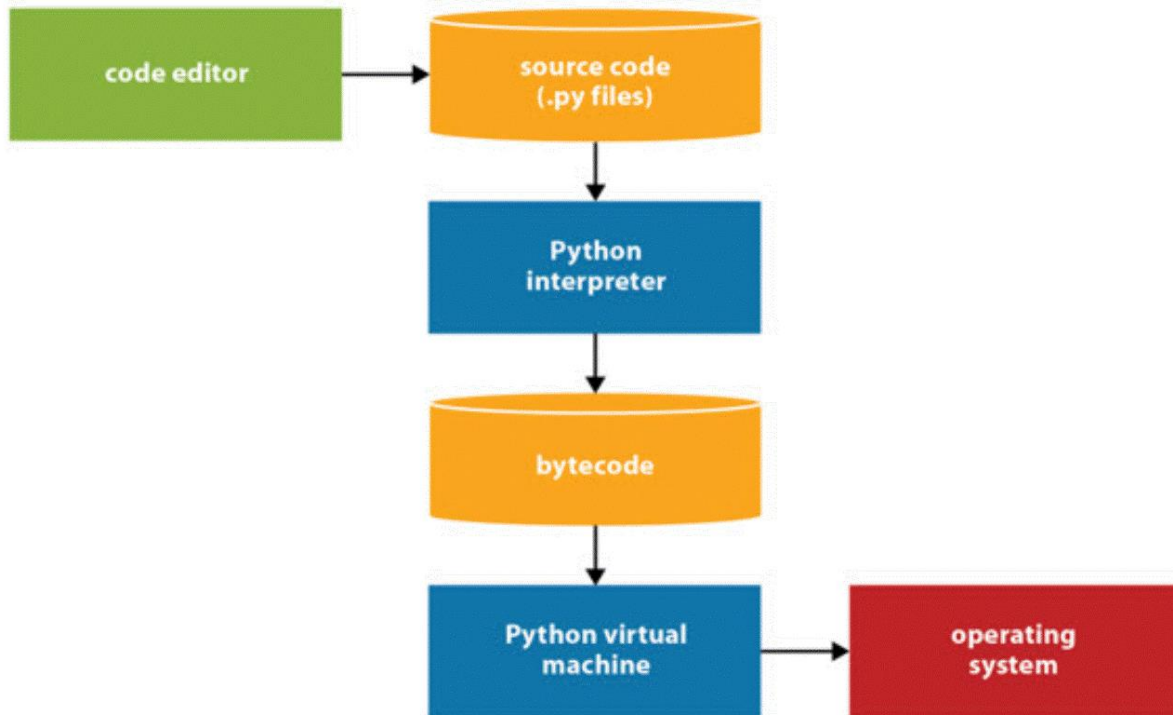
Why Learn Python?

- Accessible for beginning programmers
- Simple syntax
- Valuable skill set
- Software quality
- Program portability
- Developer productivity promotion
- Libraries support
- Component Integration
- Free
- Powerful
- Easy to use
- Portable

Creating a Program

The process for creating and running a program is as follows.

1. A programmer uses an IDE to enter source code into the editor windows. Python programs end with a .py extension.
2. The source code gets compiled by the Python interpreter into bytecode.
3. The bytecode is translated by the Python virtual machine into target code so the CPU can understand it directly.



Module 2

Design and Library Setup

Flowchart

Include the following processes:

Install python

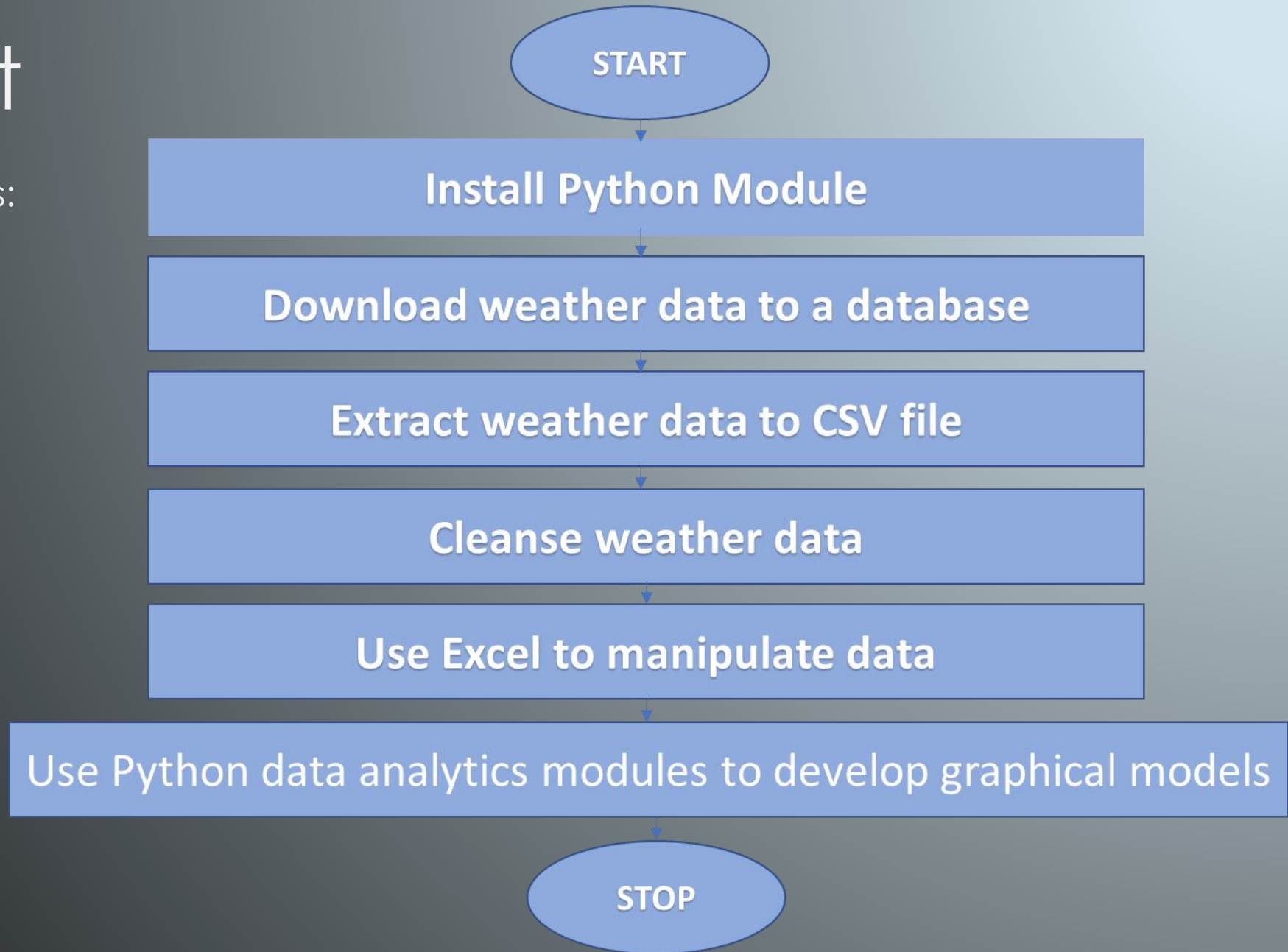
Download weather data to a database

Extract weather data from database into a comma separated file with python

Cleanse weather data

Use Excel to manipulate data

Use python data analytics modules to develop graphical models



Software Inventory

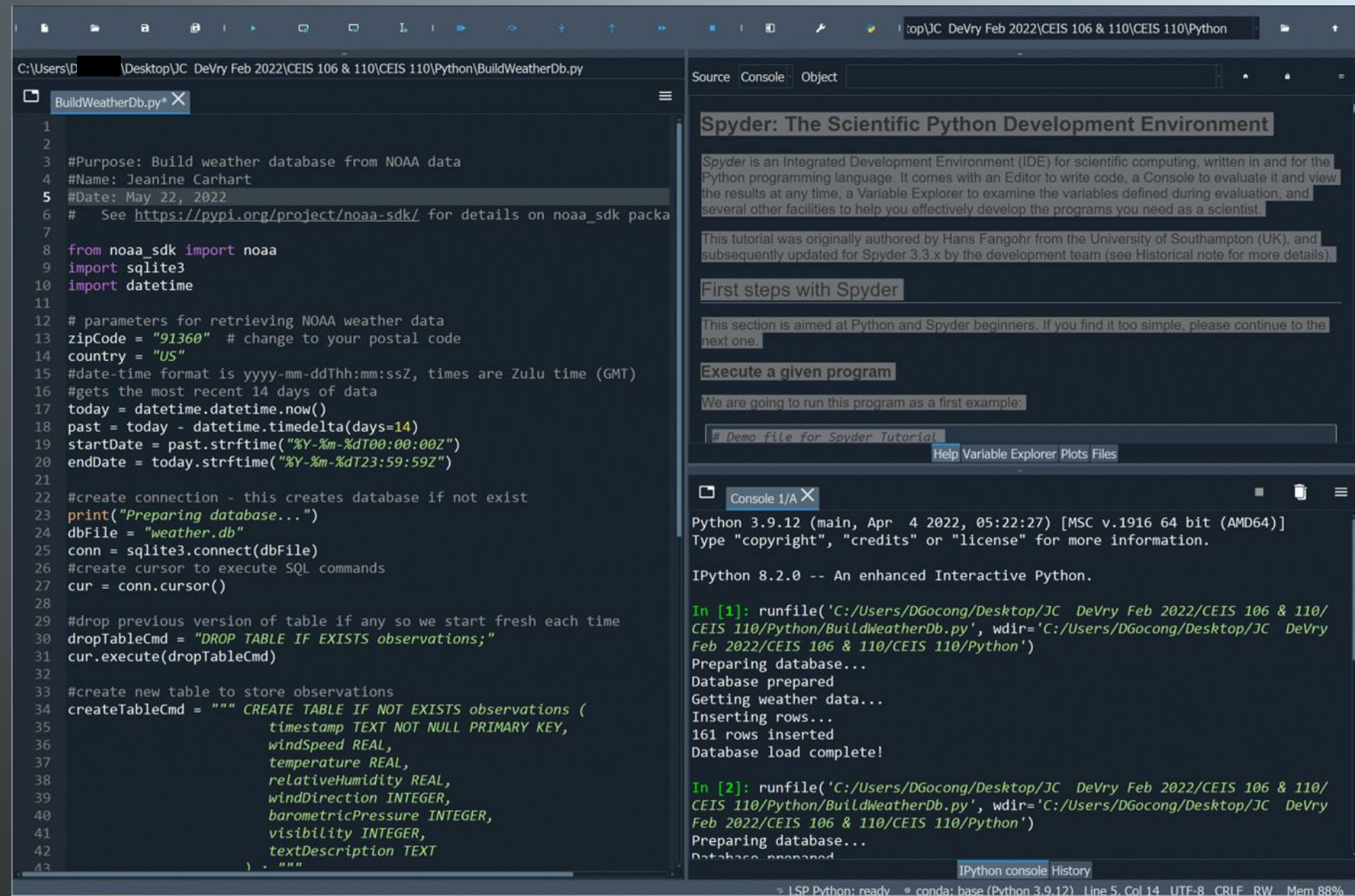
NOAA-SDK Library Installed

```
Anaconda Prompt (Anaconda3)
(base) C:\Users\ [redacted] >pip install noaa_sdk
Defaulting to user installation because normal site-packages is not writeable
Collecting noaa_sdk
  Using cached noaa_sdk-0.1.21-py3-none-any.whl (11 kB)
Requirement already satisfied: requests>=2.22.0 in c:\programdata\anaconda3\lib\site-packages (from noaa_sdk) (2.27.1)
Requirement already satisfied: charset-normalizer~=2.0.0 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.22.0->noaa_sdk) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.22.0->noaa_sdk) (2021.10.8)
Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.22.0->noaa_sdk) (3.3)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests>=2.22.0->noaa_sdk) (1.26.9)
Installing collected packages: noaa-sdk
Successfully installed noaa-sdk-0.1.21
```

Module 3

Downloading Weather Data

BuildWeatherDb.py Code



```
1
2
3 #Purpose: Build weather database from NOAA data
4 #Name: Jeanine Carhart
5 #Date: May 22, 2022
6 # See https://pypi.org/project/noaa-sdk/ for details on noaa_sdk packa
7
8 from noaa_sdk import noaa
9 import sqlite3
10 import datetime
11
12 # parameters for retrieving NOAA weather data
13 zipCode = "91360" # change to your postal code
14 country = "US"
15 #date-time format is yyyy-mm-ddThh:mm:ssZ, times are Zulu time (GMT)
16 #gets the most recent 14 days of data
17 today = datetime.datetime.now()
18 past = today - datetime.timedelta(days=14)
19 startDate = past.strftime("%Y-%m-%dT00:00:00Z")
20 endDate = today.strftime("%Y-%m-%dT23:59:59Z")
21
22 #create connection - this creates database if not exist
23 print("Preparing database...")
24 dbFile = "weather.db"
25 conn = sqlite3.connect(dbFile)
26 #create cursor to execute SQL commands
27 cur = conn.cursor()
28
29 #drop previous version of table if any so we start fresh each time
30 dropTableCmd = "DROP TABLE IF EXISTS observations;"
31 cur.execute(dropTableCmd)
32
33 #create new table to store observations
34 createTableCmd = """ CREATE TABLE IF NOT EXISTS observations (
35     timestamp TEXT NOT NULL PRIMARY KEY,
36     windSpeed REAL,
37     temperature REAL,
38     relativeHumidity REAL,
39     windDirection INTEGER,
40     barometricPressure INTEGER,
41     visibility INTEGER,
42     textDescription TEXT
43 ) . """
```

Spyder: The Scientific Python Development Environment

Spyder is an Integrated Development Environment (IDE) for scientific computing, written in and for the Python programming language. It comes with an Editor to write code, a Console to evaluate it and view the results at any time, a Variable Explorer to examine the variables defined during evaluation, and several other facilities to help you effectively develop the programs you need as a scientist.

This tutorial was originally authored by Hans Fangohr from the University of Southampton (UK), and subsequently updated for Spyder 3.3.x by the development team (see Historical note for more details).

First steps with Spyder

This section is aimed at Python and Spyder beginners. If you find it too simple, please continue to the next one.

Execute a given program

We are going to run this program as a first example:

```
# Demo file for Spyder Tutorial
```

Help Variable Explorer Plots Files

Console 1/A X

Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

```
In [1]: runfile('C:/Users/DGocong/Desktop/JC DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/BuildWeatherDb.py', wdir='C:/Users/DGocong/Desktop/JC DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
Preparing database...
Database prepared
Getting weather data...
Inserting rows...
161 rows inserted
Database load complete!

In [2]: runfile('C:/Users/DGocong/Desktop/JC DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/BuildWeatherDb.py', wdir='C:/Users/DGocong/Desktop/JC DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
Preparing database...
Database prepared
```

IPython console History

ISP Python: ready • conda: base (Python 3.9.12) Line 5, Col 14 UTF-8 CRLF RW Mem 88%

Python Console

The image shows the Spyder Python IDE interface. On the left, a code editor displays a Python script named `BuildWeatherDb.py`. The script's purpose is to build a weather database from NOAA data. It includes comments, imports for `noaa_sdk`, `sqlite3`, and `datetime`, and defines parameters like `zipCode` and `country`. It calculates a date range for the most recent 14 days of data and uses SQL commands to create a new table named `observations` with various weather-related fields.

```
1
2
3 #Purpose: Build weather database from NOAA data
4 #Name: Jeanine Carhart
5 #Date: May 22, 2022
6 # See https://pypi.org/project/noaa-sdk/ for details on noaa_sdk packa
7
8 from noaa_sdk import noaa
9 import sqlite3
10 import datetime
11
12 # parameters for retrieving NOAA weather data
13 zipCode = "91360" # change to your postal code
14 country = "US"
15 #date-time format is yyyy-mm-ddThh:mm:ssZ, times are Zulu time (GMT)
16 #gets the most recent 14 days of data
17 today = datetime.datetime.now()
18 past = today - datetime.timedelta(days=14)
19 startDate = past.strftime("%Y-%m-%dT00:00:00Z")
20 endDate = today.strftime("%Y-%m-%dT23:59:59Z")
21
22 #create connection - this creates database if not exist
23 print("Preparing database...")
24 dbFile = "weather.db"
25 conn = sqlite3.connect(dbFile)
26 #create cursor to execute SQL commands
27 cur = conn.cursor()
28
29 #drop previous version of table if any so we start fresh each time
30 dropTableCmd = "DROP TABLE IF EXISTS observations;"
31 cur.execute(dropTableCmd)
32
33 #create new table to store observations
34 createTableCmd = """ CREATE TABLE IF NOT EXISTS observations (
35     timestamp TEXT NOT NULL PRIMARY KEY,
36     windSpeed REAL,
37     temperature REAL,
38     relativeHumidity REAL,
39     windDirection INTEGER,
40     barometricPressure INTEGER,
41     visibility INTEGER,
42     textDescription TEXT
43 ) ; """
```

On the right, the Spyder console window shows the output of the script execution. It displays the title "Spyder: The Scientific Python Development Environment" and introductory text. Below that, it shows the execution of the script, including the output of the `print` statement and the successful execution of the SQL commands. The console also shows the IPython prompt and the execution of the script again, demonstrating the database preparation process.

Python 3.9.12 (main, Apr 4 2022, 05:22:27) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 8.2.0 -- An enhanced Interactive Python.

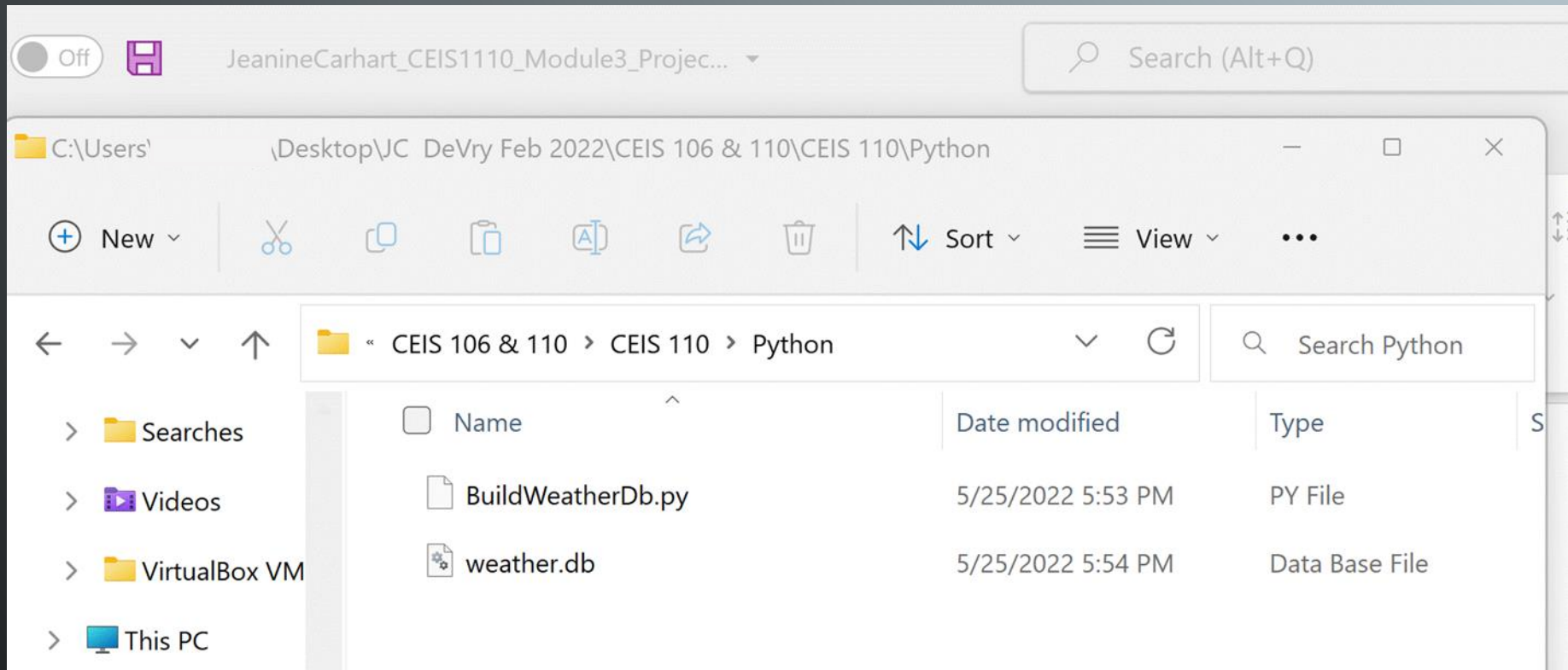
In [1]: runfile('C:/Users/DGocong/Desktop/JC DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/BuildWeatherDb.py', wdir='C:/Users/DGocong/Desktop/JC DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
Preparing database...
Database prepared
Getting weather data...
Inserting rows...
161 rows inserted
Database load complete!

In [2]: runfile('C:/Users/DGocong/Desktop/JC DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/BuildWeatherDb.py', wdir='C:/Users/DGocong/Desktop/JC DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
Preparing database...
Database prepared

IPython console History

ISP Python: ready conda: base (Python 3.9.12) Line 5, Col 14 UTF-8 CRLF RW Mem 89%

Weather.db File



Module 4

Querying the Database with SQL

Querying to Retrieve all Columns and Rows

```
In [2]: runfile('C:/Users/          /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/QueryWeatherDB.py',  
wdir='C:/Users/          /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
```

	timestamp	windSpeed	temperature	relativeHumidity	windDirection	barometricPressure	visibility	textDescription
0	2022-05-23T01:00:00+00:00	11.880	17.9	60.989512	289.0	None	None	
1	2022-05-23T03:00:00+00:00	7.200	15.5	68.005278	332.0	None	None	
2	2022-05-23T04:00:00+00:00	NaN	NaN	NaN	NaN	None	None	
3	2022-05-23T05:00:00+00:00	5.040	12.6	78.980694	343.0	None	None	
4	2022-05-23T06:00:00+00:00	NaN	NaN	NaN	NaN	None	None	
5	2022-05-23T07:00:00+00:00	1.440	12.0	82.990709	319.0	None	None	
6	2022-05-23T08:00:00+00:00	1.440	11.8	82.019994	340.0	None	None	
7	2022-05-23T09:00:00+00:00	1.440	11.3	83.017312	292.0	None	None	
8	2022-05-23T10:00:00+00:00	1.080	10.9	83.987953	270.0	None	None	
9	2022-05-23T11:00:00+00:00	1.800	10.5	83.998564	323.0	None	None	
10	2022-05-23T12:00:00+00:00	2.160	10.2	84.999740	305.0	None	None	
11	2022-05-23T13:00:00+00:00	3.240	9.7	88.012004	284.0	None	None	
12	2022-05-23T14:00:00+00:00	NaN	NaN	NaN	NaN	None	None	
13	2022-05-23T15:00:00+00:00	3.240	10.1	89.977946	334.0	None	None	
14	2022-05-23T16:00:00+00:00	3.240	10.9	89.977390	330.0	None	None	
15	2022-05-23T17:00:00+00:00	6.120	13.0	85.018590	297.0	None	None	
16	2022-05-23T18:00:00+00:00	6.480	16.2	69.989095	302.0	None	None	
17	2022-05-23T19:00:00+00:00	10.440	18.0	62.994305	298.0	None	None	
18	2022-05-23T20:00:00+00:00	10.440	19.1	58.959058	295.0	None	None	
19	2022-05-23T21:00:00+00:00	11.520	19.7	57.979868	294.0	None	None	
20	2022-05-23T22:00:00+00:00	12.960	19.5	59.961070	287.0	None	None	
21	2022-05-23T23:00:00+00:00	12.960	19.0	60.963333	293.0	None	None	
22	2022-05-24T00:00:00+00:00	11.880	18.6	61.970067	289.0	None	None	
23	2022-05-24T01:00:00+00:00	NaN	NaN	NaN	NaN	None	None	
24	2022-05-24T02:00:00+00:00	9.360	16.8	66.966703	291.0	None	None	
25	2022-05-24T03:00:00+00:00	7.920	15.5	71.983043	301.0	None	None	
26	2022-05-24T04:00:00+00:00	5.400	13.9	78.015818	302.0	None	None	
27	2022-05-24T05:00:00+00:00	

Query to retrieve highest and lowest temperatures

```
Console 1/A X
157 2022-05-29T18:00:00+00:00 9.360 19.4 56.989699 289.0 None None
158 2022-05-29T19:00:00+00:00 11.880 20.4 53.959323 286.0 None None
159 2022-05-29T20:00:00+00:00 14.040 20.8 50.954512 295.0 None None
160 2022-05-29T21:00:00+00:00 15.120 21.8 41.975955 283.0 None None
161 2022-05-29T22:00:00+00:00 15.840 21.5 41.976617 295.0 None None
162 2022-05-29T23:00:00+00:00 13.680 22.2 37.981908 285.0 None None

In [3]: runfile('C:/Users/ /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/QueryWeatherDB.py',
wdir='C:/Users/ /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
MIN(temperature) MAX(temperature)
0 9.7 23.4

In [4]: runfile('C:/Users/ /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/BuildWeatherDb.py',
wdir='C:/Users/ /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
Preparing database...
Database prepared
Getting weather data...
Inserting rows...
163 rows inserted
Database load complete!

In [5]: runfile('C:/Users/ /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/QueryWeatherDB.py',
wdir='C:/Users/ /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
MIN(temperature) MAX(temperature)
0 9.7 23.4

In [6]:
```

Query to retrieve all clear days

```
Console 1/A X
In [8]: runfile('C:/Users/          /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/BuildWeatherDb.py',
wdir='C:/Users/          /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
Preparing database...
Database prepared
Getting weather data...
Inserting rows...
163 rows inserted
Database load complete!

In [9]: runfile('C:/Users/          /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/QueryWeatherDB.py',
wdir='C:/Users/          /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
Empty DataFrame
Columns: [temperature, windSpeed, textDescription]
Index: []

In [10]: runfile('C:/Users/          /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/BuildWeatherDb.py',
wdir='C:/Users/          /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
Preparing database...
Database prepared
Getting weather data...
Inserting rows...
163 rows inserted
Database load complete!

In [11]: runfile('C:/Users/          /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python/QueryWeatherDB.py',
wdir='C:/Users/          /Desktop/Jeanine Carhart DeVry Feb 2022/CEIS 106 & 110/CEIS 110/Python')
Empty DataFrame
Columns: [temperature, windSpeed, textDescription]
Index: []

In [12]:
```

Module 5

Querying and Manipulating Data with SQL and Python

Python code: ExtractTempHumidity.py

The image shows a Python IDE with a code editor on the left and a console on the right. The code editor displays the following Python code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Jun 7 23:23:38 2022
4 @author: JCarhart
5 """
6 #Purpose: Extract temperature, humidity data from weather database
7 #Name: Jeanine Carhart
8 #Date: June 5, 2022
9 # Run BuildWeatherDB.py to build weather database before running
10
11 import sqlite3
12
13 #convert Celsius temperature to Fahrenheit
14 def convertCtoF(tempC):
15     return (tempC*9.0/5.0) + 32.0
16
17 #file names for database and output file
18 dbFile = "weather.db"
19 output_file_name='formatdata.csv'
20
21 #connect to and query weather database and
22 dbFile = "weather.db"
23 conn = sqlite3.connect(dbFile)
24 #create cursor to execute SQL commands
25 cur = conn.cursor()
26 selectCmd = """ SELECT temperature, relativeHumidity FROM observ
27                 ORDER BY timestamp; """
28 cur.execute(selectCmd)
29 allRows = cur.fetchall()
30 #limit the number of rows output to half
31 rowCount = len(allRows)//2 # double slash does integer division
32 rows = allRows[:rowCount]
33
34 #write data to output file
35 with open(output_file_name,"w+") as outf:
36     outf.write('Celsius,Fahrenheit,Humidity')
37     outf.write('\n')
38     for row in rows:
39         tempC = row[0]
```

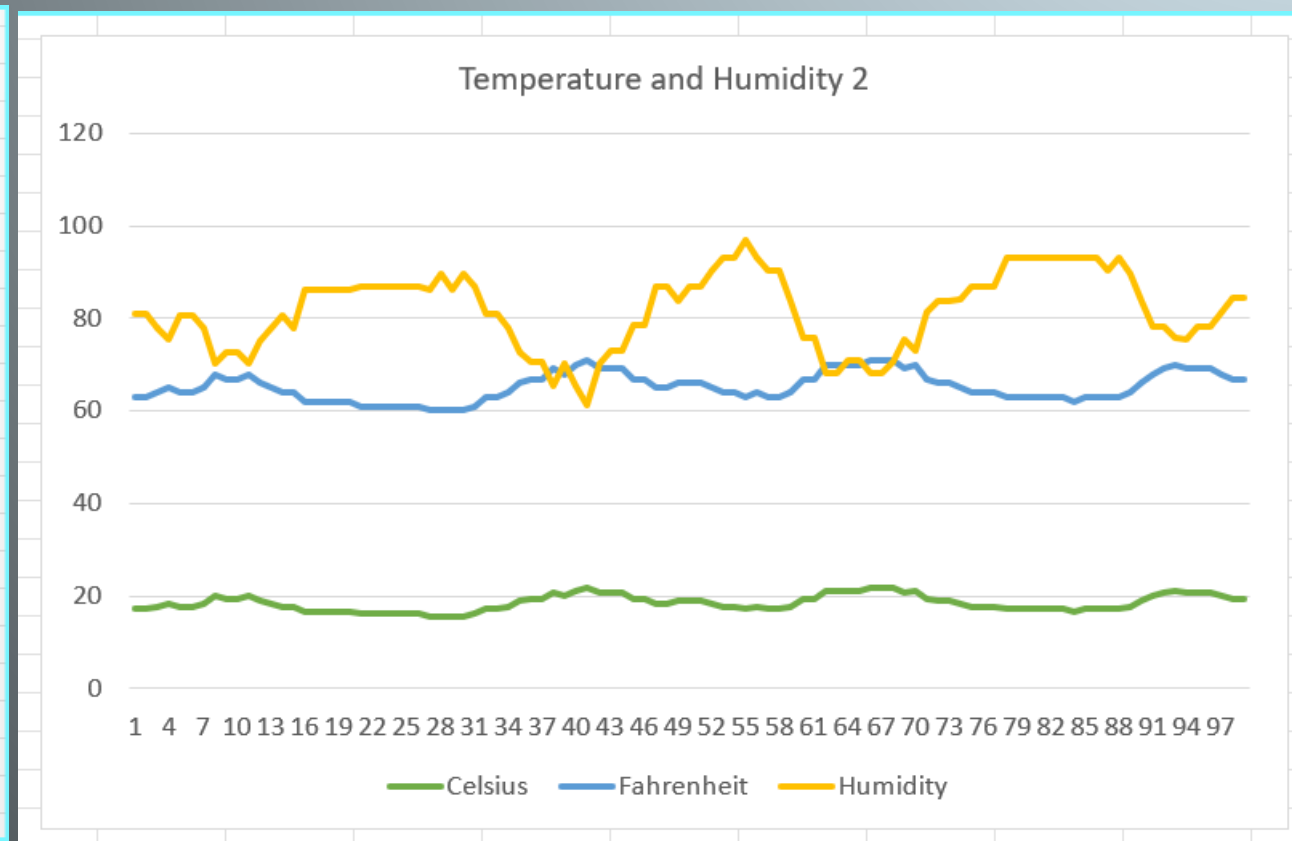
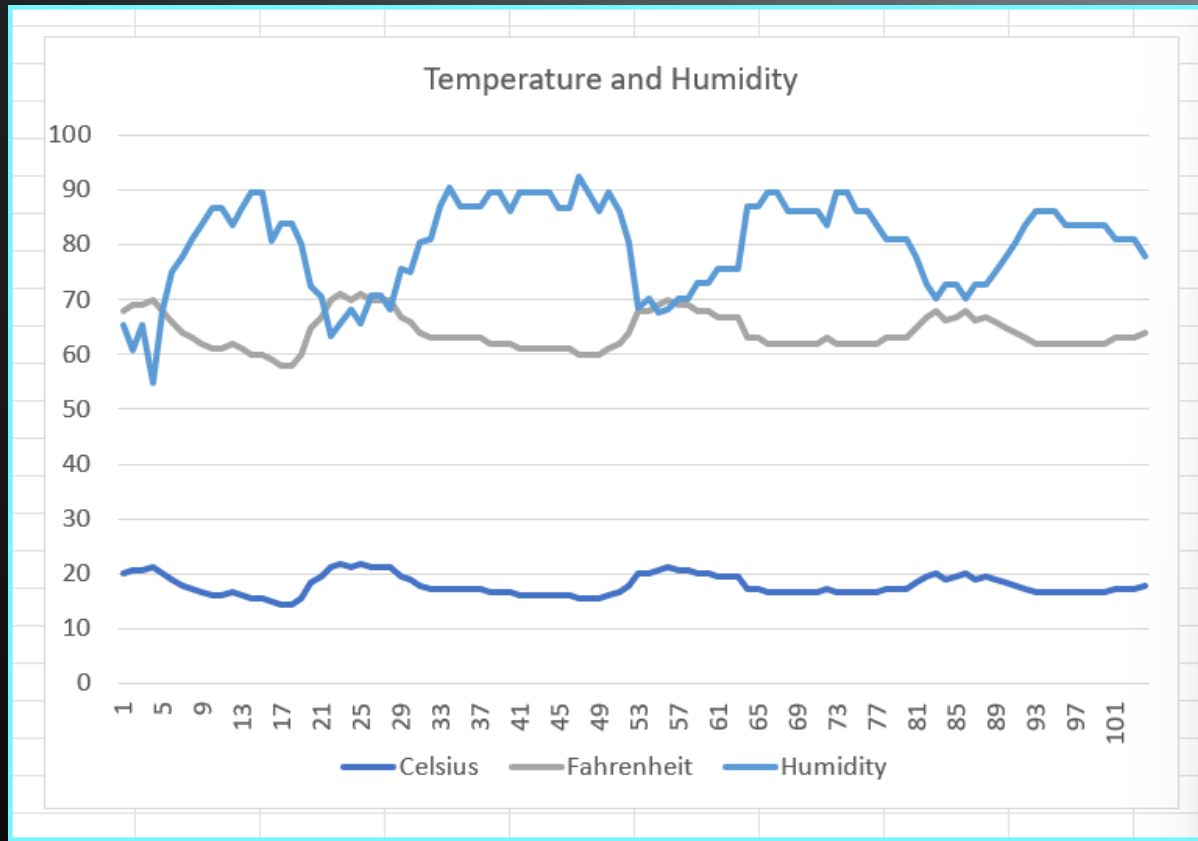
The console window shows the output of the code, displaying a table of weather data:

182	2022-06-07T03:51:00+00:00	5.40	18.3	84.273097	230.0	100980.0	16090	Mostly	Cloudy
183	2022-06-07T04:12:00+00:00	NaN	NaN	NaN	NaN	NaN	16090	Mostly	Clear
184	2022-06-07T04:51:00+00:00	5.40	17.8	86.962936	190.0	101020.0	16090		Clear
185	2022-06-07T05:51:00+00:00	0.00	17.8	86.962936	0.0	101050.0	16090		Clear
186	2022-06-07T06:51:00+00:00	5.40	17.8	86.962936	240.0	101020.0	16090		Clear
187	2022-06-07T07:51:00+00:00	5.40	17.2	93.251499	210.0	101020.0	16090		Clear
188	2022-06-07T08:51:00+00:00	0.00	17.2	93.251499	0.0	100980.0	14480	Mostly	Clear
189	2022-06-07T09:12:00+00:00	0.00	17.2	93.251499	0.0	100980.0	16090	Mostly	Cloudy
190	2022-06-07T09:44:00+00:00	0.00	17.2	93.251499	0.0	100980.0	14480		Cloudy
191	2022-06-07T09:51:00+00:00	0.00	17.2	93.251499	0.0	100980.0	14480		Cloudy
192	2022-06-07T10:00:00+00:00	0.00	17.2	93.251499	0.0	100980.0	16090		Cloudy
193	2022-06-07T10:51:00+00:00	0.00	16.7	93.226340	0.0	100980.0	14480		Cloudy
194	2022-06-07T11:31:00+00:00	0.00	17.2	93.251499	0.0	100980.0	14480		Cloudy
195	2022-06-07T11:40:00+00:00	5.40	17.2	93.251499	170.0	101020.0	14480		Cloudy
196	2022-06-07T11:51:00+00:00	NaN	17.2	90.318702	0.0	101020.0	12870		Cloudy
197	2022-06-07T12:51:00+00:00	0.00	17.2	93.251499	0.0	101050.0	12870		Cloudy
198	2022-06-07T13:51:00+00:00	5.40	17.8	89.786765	50.0	101080.0	9660		
199	2022-06-07T15:51:00+00:00	0.00	18.9	83.803385	0.0	101120.0	11270		Cloudy

Retrieve and Convert Data to CSV Format

A	B	C
Celsius	Fahrenheit	Humidity
20	68	65.33508183
20.6	69.08	60.93509447
20.6	69.08	65.46328848
21.1	69.98	54.96814752
20	68	67.93498729
18.9	66.02	75.13736573
17.8	64.04	77.93714995
17.2	62.96	80.94462547
16.7	62.06	83.55048294
16.1	60.98	86.80249244
16.1	60.98	86.80249244
16.7	62.06	83.55048294
16.1	60.98	86.80249244
15.6	60.08	89.62111213
15.6	60.08	89.62111213
15	59	80.65022333
14.4	57.92	83.83170254
14.4	57.92	83.83170254
15.6	60.08	80.202282
18.3	64.94	72.63605068
19.4	66.92	70.51195827
21.1	69.98	63.48053948
21.7	71.06	65.69668793
21.1	69.98	68.15622481
21.7	71.06	65.69668793
21.1	69.98	70.83212646
21.1	69.98	70.83212646

Temperature and Humidity Chart

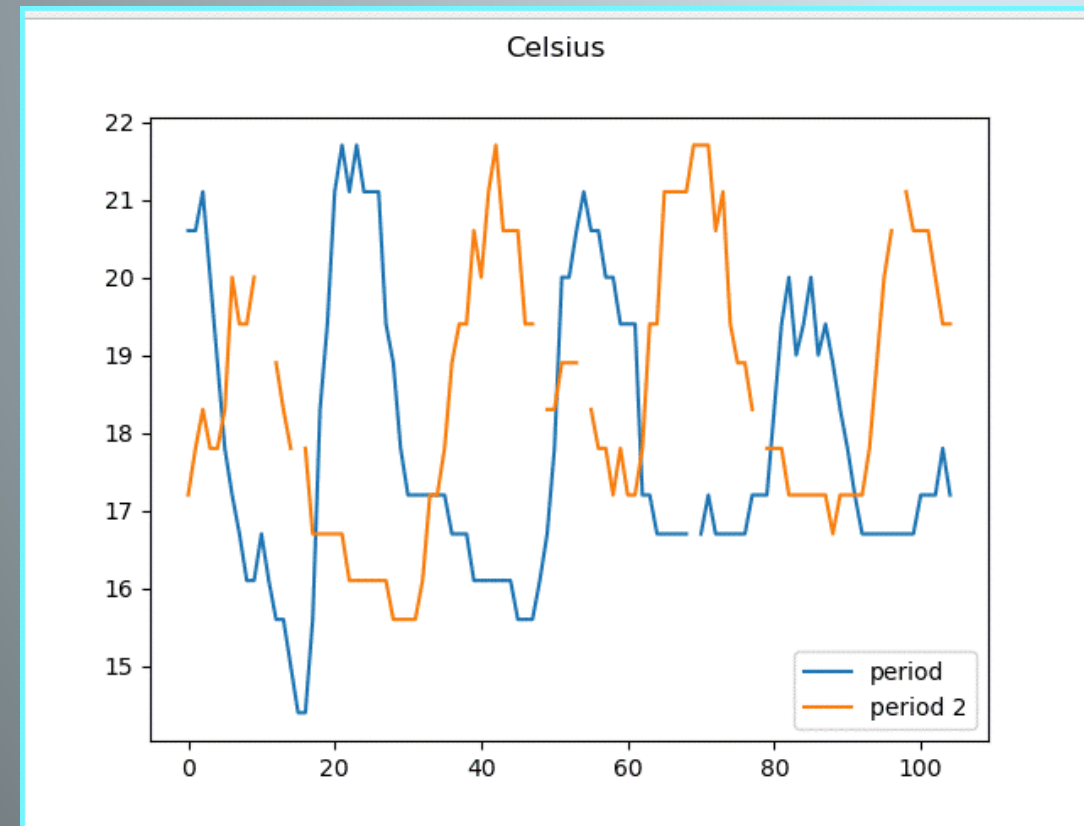


Module 6

Develop Graphical Models
and Interpret Results

Plot #1

Plot and Code



```
#Purpose: Create a histogram of celsius data comparing the first and second periods
```

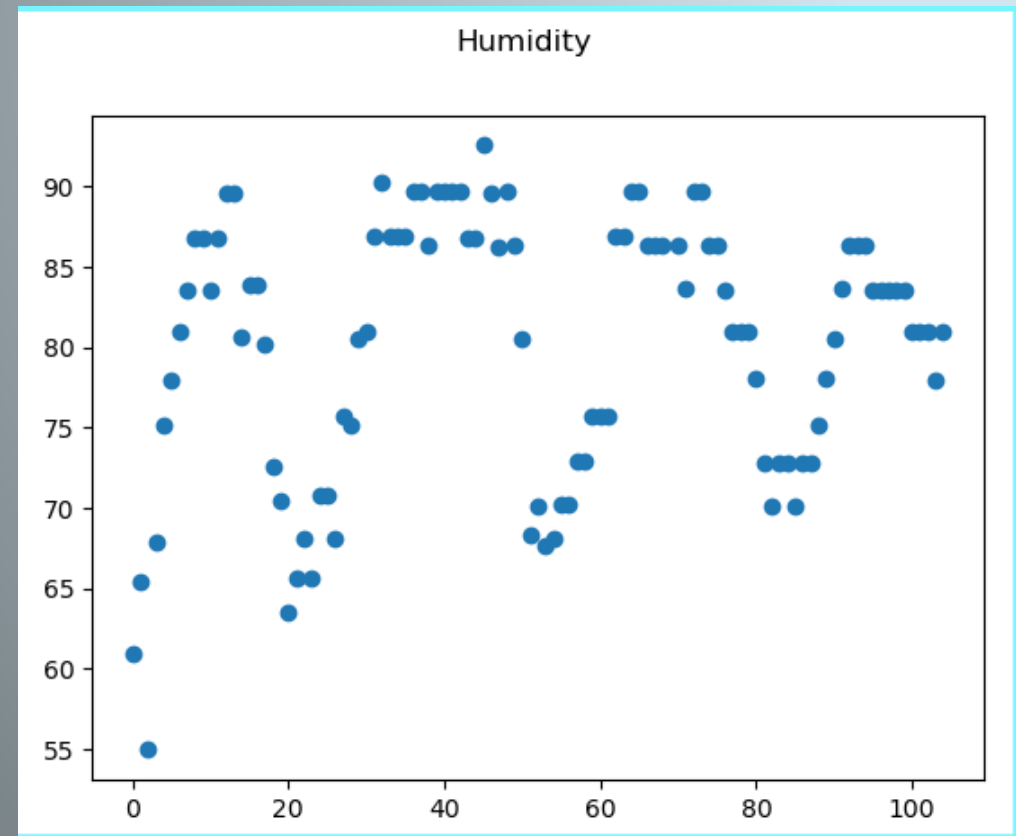
```
#Name: Jeanine Carhart
```

```
#Date: June 12, 2022
```

```
import pandas as pd
import matplotlib.pyplot as plt
df1 = pd.read_csv("formatdata.csv") #baseline data is period 1 (older)
df2 = pd.read_csv("formatdata2.csv") #data for period 2 (more recent)
plt.figure(); df1.Celsius.plot(label='period'); df2.Celsius.plot(label='period 2');
plt.legend(loc='best');plt.suptitle('Celsius')
plt.show()
```

Plot #2

Create a plot from
your own question



```
#Purpose: Create a scatter plot of humidity data from the first and second period
```

```
#Name: Jeanine Carhart
```

```
#Date: June 12, 2022
```

```
import pandas as pd
import matplotlib.pyplot as plt
df1 =pd.read_csv("formatdata.csv")
df2 =pd.read_csv("formatdata2.csv")
plt.scatter(df1.index.values,df1['Humidity']);plt.suptitle('Humidity')
plt.show()
```

Analysis

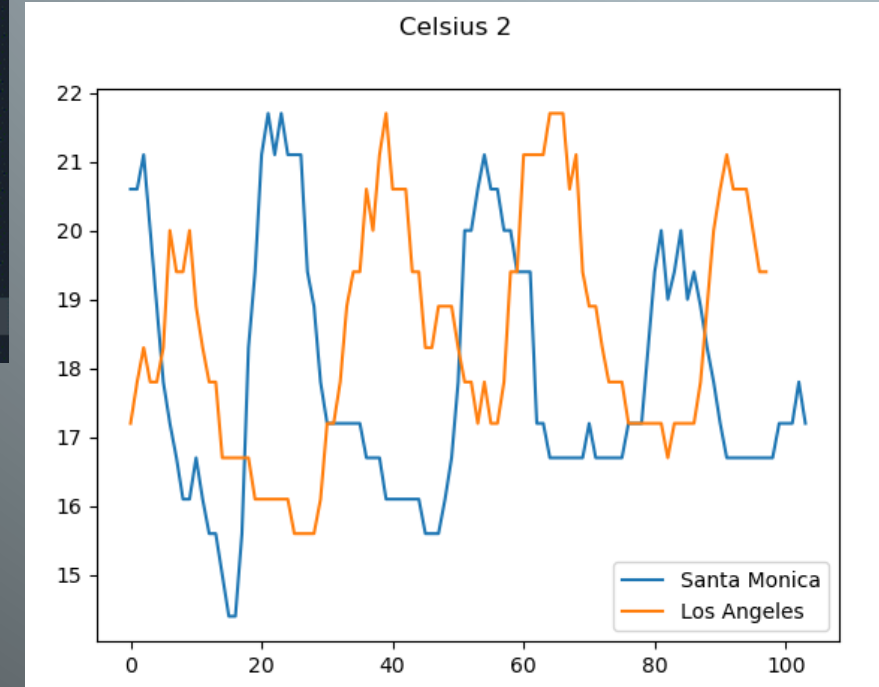
Are temperatures similar in Santa Monica and Los Angeles?

HistogramOfCelsius2_Week6.py X

```
#Purpose: Create a histogram of celsius data comparing the first and second periods
#Name: Jeanine Carhart
#Date: June 12, 2022

import pandas as pd
import matplotlib.pyplot as plt
df1 =pd.read_csv("formatdata.csv") #baseline data is period 1 (older)
df2 =pd.read_csv("formatdata3.csv") #data for period 2 (more recent)
plt.figure(); df1.Celsius.plot(label='Santa Monica'); df2.Celsius.plot(label='Los Angeles');
plt.legend(loc='best');plt.suptitle('Celsius 2')
plt.show()
```

Answer: Yes



Prediction

Develop a prediction based on the data. What variations in temperature and humidity do you expect over the next few hours or days? How would humidity change if temperature goes up or down?

Santa Monica is on the ocean and has less traffic and streets which increases the heat, however, the ocean breeze reduces the temperature and increases humidity.

Santa Monica: Average humidity for May and June 2022: 82% Average temperature: 64°F

Los Angeles is more populous than Santa Monica, has more traffic and streets, and is a little bit further away from the ocean, resulting in higher temperatures, but cools down in the evenings. At this time of year, however, the temperatures between Los Angeles and Santa Monica vary slightly.

Los Angeles: Average humidity for May and June 2022: 74% Average temperature: 66°F

Challenges

Challenges

Remembering the commands is challenging.
Practice will help.

Figuring out how to undo a mistake took some
work to figure out.

Planning the time for writing code will take some
time.

Career Skills

Career Skills

Attention to detail: When the code isn't working, it can come down to a missing period or comma. The details are very important.

Persistence: This is needed if the missing period in the code is ever going to be found.

Research and analytical skills: Very important for interpreting the results of analyzing data.

Conclusion

Why Learn Python?

- Accessible for beginning programmers
- Simple syntax
- Valuable skill set
- Software quality
- Program portability
- Developer productivity promotion
- Libraries support
- Component Integration
- Free
- Powerful
- Easy to use
- Portable

LESSONS LEARNED

- ✓ Python is a Valuable Skill
- ✓ Creating programs
- ✓ Design and Library Setup
- ✓ Create Flowcharts
- ✓ Software Inventory
- ✓ Build a Database
- ✓ Work with Python Console
- ✓ Work with Weather Database
- ✓ Query to Retrieve Columns and Rows
- ✓ Query to Retrieve High/Low Temperatures
- ✓ Query to Retrieve Clear Days
- ✓ Query and Manipulate SQL and Python Data
- ✓ Query the Database with SQL
- ✓ Develop Graphical Models and Interpret Results
- ✓ Create Plots and Predict Results

Challenges

- Remembering the commands are challenging
- Careful coding can reduce the time spent finding errors
- Manage the time needed for writing code

Career Skills

- ❖ Attention to detail
- ❖ Persistence
- ❖ Research and analytical skills