

CEIS110 PROGRAMMING WITH PYTHON

JAMES GARLIE
DEVRY UNIVERSITY:
FEBRUARY 2022

DATA IS GROWING
EXPONENTIALLY.

THIS PROJECT USES A
CLOUD-BASED SYSTEM TO
GATHER TEMPERATURE
AND HUMIDITY DATA.

THEN THE DATA IS
ANALYZED USING
PROGRAMMING AND DATA
ANALYTICS.

INTRODUCTION

BEFORE DEVELOPING THIS
PROGRAMMING WITH
DATA PROJECT, ALL
SOFTWARE MUST BE
DOWNLOADED AND
INSTALLED.

SOFTWARE NEEDED,
INCLUDED: SQLITE STUDIO,
EXCEL, AND A PYTHON IDE
- ANACONDA

SOFTWARE INVENTORY

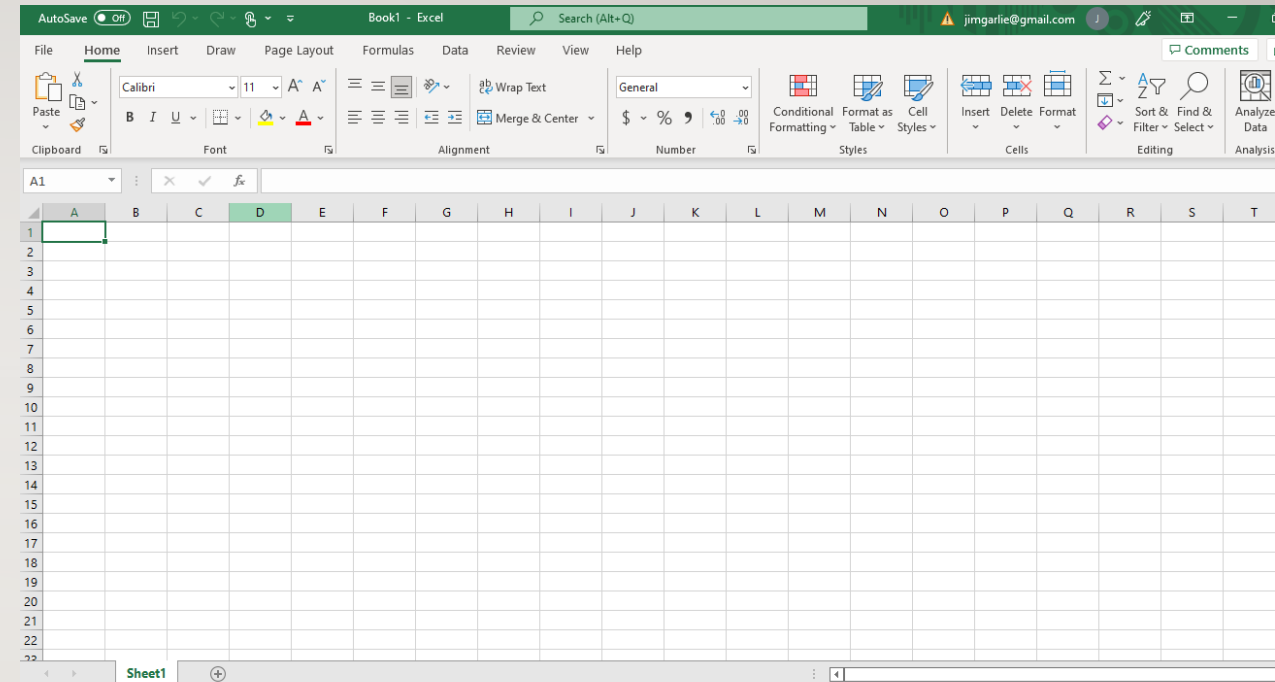
SOFTWARE

The software need for this project includes:

Microsoft Excel

SQLite

Anaconda Navigator with Spyder



What Is SQLite?

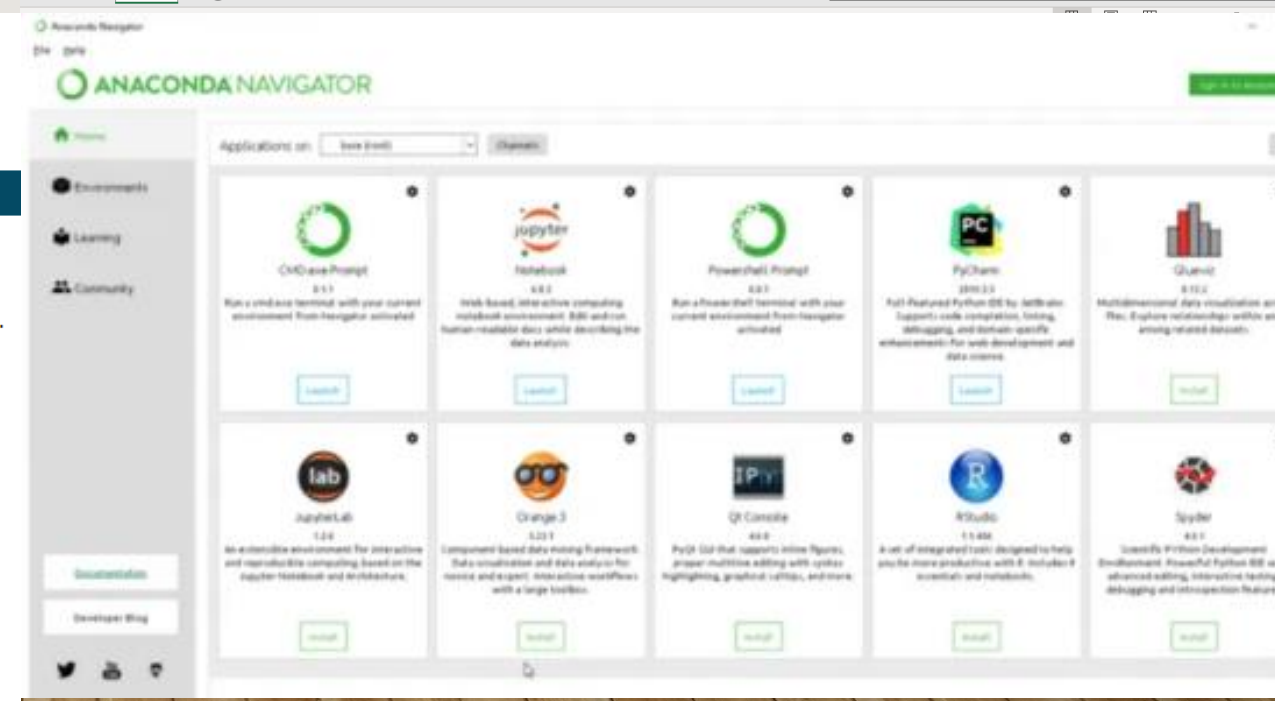
SQLite is a C-language library that implements a [small](#), [fast](#), [self-contained](#), [high-reliability](#), [full-featured](#), SQL database engine. SQLite is the [most used](#) database engine in the world. SQLite is built into all mobile phones and most computers and comes bundled inside countless other applications that people use every day. [More Information...](#)

The SQLite [file format](#) is stable, cross-platform, and backwards compatible and the developers pledge to keep it that way [through the year 2050](#). SQLite database files are commonly used as containers to transfer rich content between systems [1] [2] [3] and as a long-term archival format for data [4]. There are over 1 trillion (1e12) SQLite databases in active use [5].

SQLite [source code](#) is in the [public-domain](#) and is free to everyone to use for any purpose.

Latest Release

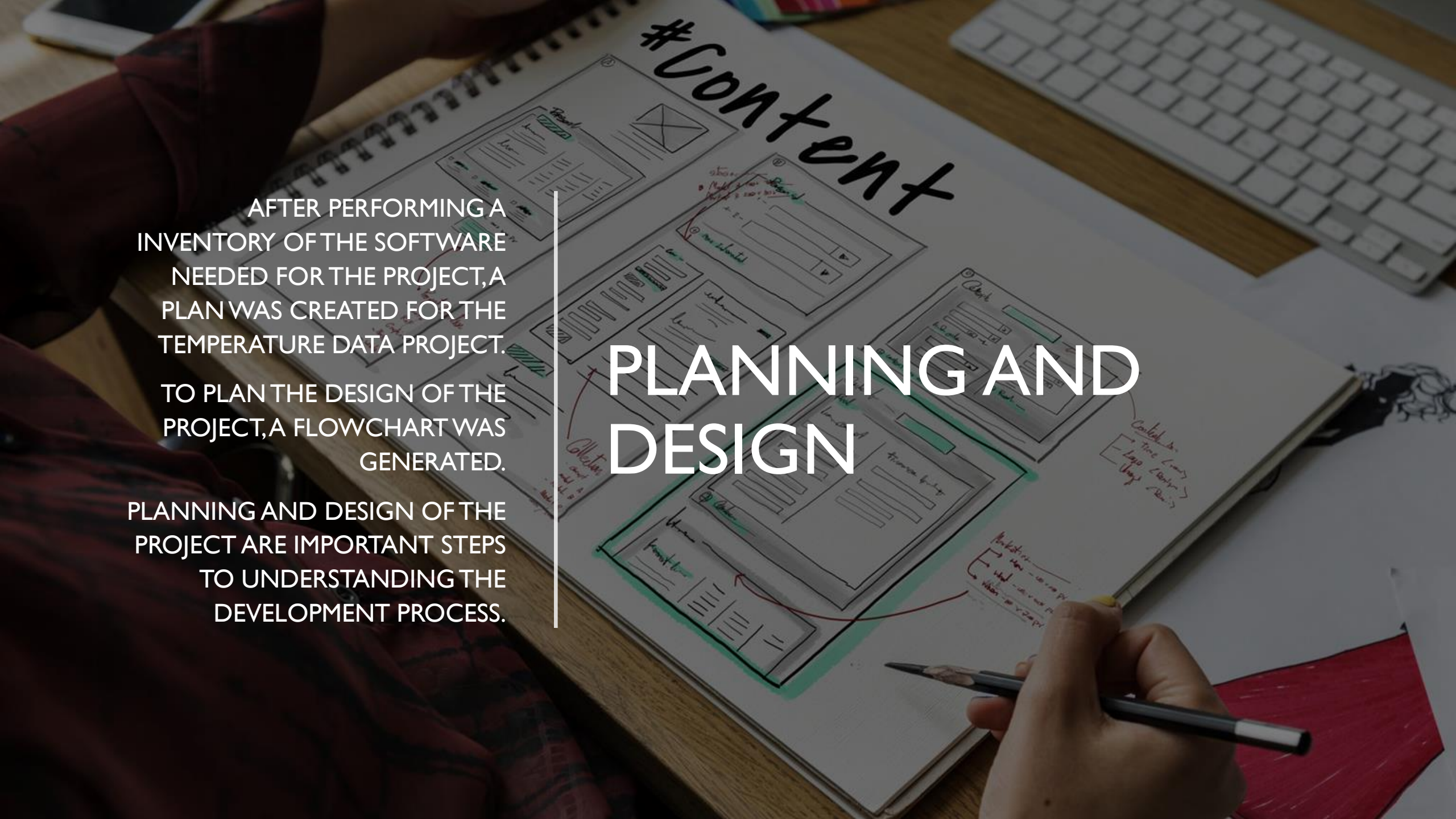
[Version 3.37.2](#) (2022-01-06). [Download](#) [Prior Releases](#)



AFTER PERFORMING A
INVENTORY OF THE SOFTWARE
NEEDED FOR THE PROJECT, A
PLAN WAS CREATED FOR THE
TEMPERATURE DATA PROJECT.
TO PLAN THE DESIGN OF THE
PROJECT, A FLOWCHART WAS
GENERATED.

PLANNING AND DESIGN OF THE
PROJECT ARE IMPORTANT STEPS
TO UNDERSTANDING THE
DEVELOPMENT PROCESS.

PLANNING AND DESIGN



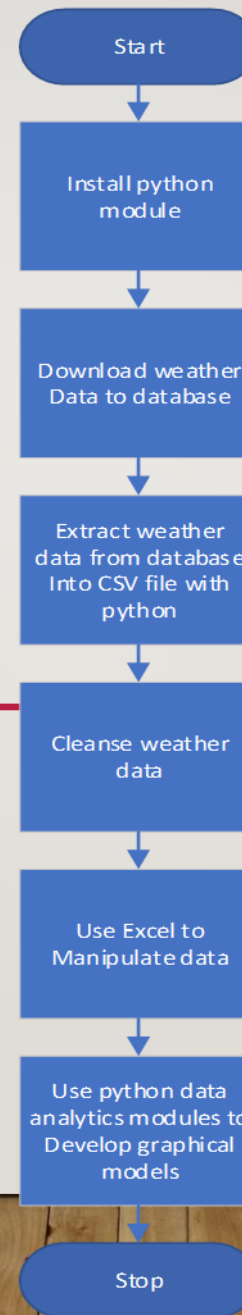
FLOWCHARTS ARE GRAPHICAL
PRESENTATIONS OF PROCESSES
OR WORKFLOWS, AND THE
FLOWCHART DEVELOPED
ILLUSTRATES THE PROCESS AND
OUTPUT OF THIS SOFTWARE
DEVELOPMENT PROJECT.

MANY COMPANIES USE
FLOWCHARTS AS SIMPLE
NARRATIVES.

WHAT ARE FLOWCHARTS?

FLOWCHART

- 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



```
9,>=2.5
-2.8-py2.py3-none-any.whl (58 kB)
| 58 kB 1.7 MB/s
ed packages: urllib3, idna, chardet, requests, noaa-sdk
tall: urllib3
g installation: urllib3 1.26.7
urllib3-1.26.7:
y uninstalled urllib3-1.26.7
tall: idna
g installation: idna 3.2
dna-3.2:
y uninstalled idna-3.2
tall: chardet
g installation: chardet 4.0.0
harden-4.0.0:
y uninstalled chardet-4.0.0
tall: requests
g installation: requests 2.26.0
requests-2.26.0:
y uninstalled requests-2.26.0
endency resolver does not currently ta
f the following dependency conflicts:
res PyQt5<5.13, which is not installed.
res PyQtWebEngine<5.13, which is not installed.
0.4 requires pathlib, which is not installed.
0.10.1 requires ruamel-yaml, which is not installed.
! requires requests>=2.23.0, but you have requests 2.22.0 which is incompatible.
alled chardet-3.0.4 idna-2.8 noaa-sdk-0.1.19 requests-2.22.0 urllib3-1.25.11
```

ADDING LIBRARY

IN ORDER FOR PYTHON TO CONNECT TO THE US GOVERNMENT'S NATIONAL OCEANIC & ATMOSPHERIC ADMINISTRATION (NOAA) WEATHER DATA SERVICE USING A CLOUD-BASED APPLICATION PROGRAMMING INTERFACE (API), A LIBRARY MODULE MUST BE INSTALLED TO USE THIS SERVICE.

THE SCREENSHOT SHOWS THE NOAA-SDK LIBRARY IS INSTALLED. This behaviour



GATHERING TEMPERATURE & HUMIDITY DATA

AFTER PLANNING AND DESIGN, THE CODE WAS DEVELOPED TO DOWNLOAD A SET OF WEATHER OBSERVATIONS. THIS DATA WAS STORED ON A LOCAL DATABASE IN A TABLE FOR LATER ANALYSIS.

BUILDWEATHERDB.PY CODE (SCREENSHOT)

The code will create a table named Observations with the fields: timestamp, windSpeed, temperature, relativeHumidity, windDirection, barometricPressue, visibility, and textDescription.

The database will be named weather.db and stored in the same directory as the python code.

The screenshots are of the code and output in Spyder.

```
temp.py x
#Purpose: build weather database from NOAA data
#Name: James Garlie
#Date: 01/21/2022
# See https://pypi.org/project/noaa-sdk/ for details on noaa_sdk package used

from noaa_sdk import noaa
import sqlite3
import datetime

# parameters for retrieving NOAA weather data
zipCode = "89103" # change to your postal code
country = "US"
#date-time format is yyyy-mm-ddThh:mm:ssZ, times are Zulu time (GMT)
#gets the most recent 14 days of data
today = datetime.datetime.now()
past = today - datetime.timedelta(days=14)
startDate = past.strftime("%Y-%m-%dT00:00:00Z")
endDate = today.strftime("%Y-%m-%dT23:59:59Z")

#create connection - this creates database if not exist
print("Preparing database...")
dbFile = "weather.db"
conn = sqlite3.connect(dbFile)
#create cursor to execute SQL commands
cur = conn.cursor()

table if any so we start fresh each time
CREATE IF EXISTS observations;
```

```
Console 1/A x
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.29.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/16123/.spyder-py3/temp.py', wdir='C:/Users/16123/.spyder-py3')
Preparing database...
Database prepared
Getting weather data...
Inserting rows...
164 rows inserted
Database load complete!

In [2]: runfile('C:/Users/16123/.spyder-py3/temp.py', wdir='C:/Users/16123/.spyder-py3')
Preparing database...
Database prepared
```



Date modified

THE SCREENSHOT BELOW
SHOWS THE
WEATHERDB.PY FILE WAS
CREATED AND IS STORED
IN WINDOWS EXPLORER.

WEATHER.DB FILE
(SCREENSHOT)

Module3_Project_Template... 1/22/2022 10:52 AM
1/22/2022 10:42 AM

```
object to mirror  
mirror_mod.mirror_object
```

```
operation == "MIRROR_X":  
    mirror_mod.use_x = True  
    mirror_mod.use_y = False  
    mirror_mod.use_z = False  
operation == "MIRROR_Y":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = True  
    mirror_mod.use_z = False  
operation == "MIRROR_Z":  
    mirror_mod.use_x = False  
    mirror_mod.use_y = False  
    mirror_mod.use_z = True
```

```
selection at the end -add  
mirror_ob.select=1  
mirror_ob.select=1  
context.scene.objects.active  
("Selected" + str(modifier  
mirror_ob.select = 0  
= bpy.context.selected_obj  
data.objects[ne. me  
print("please select exactly
```

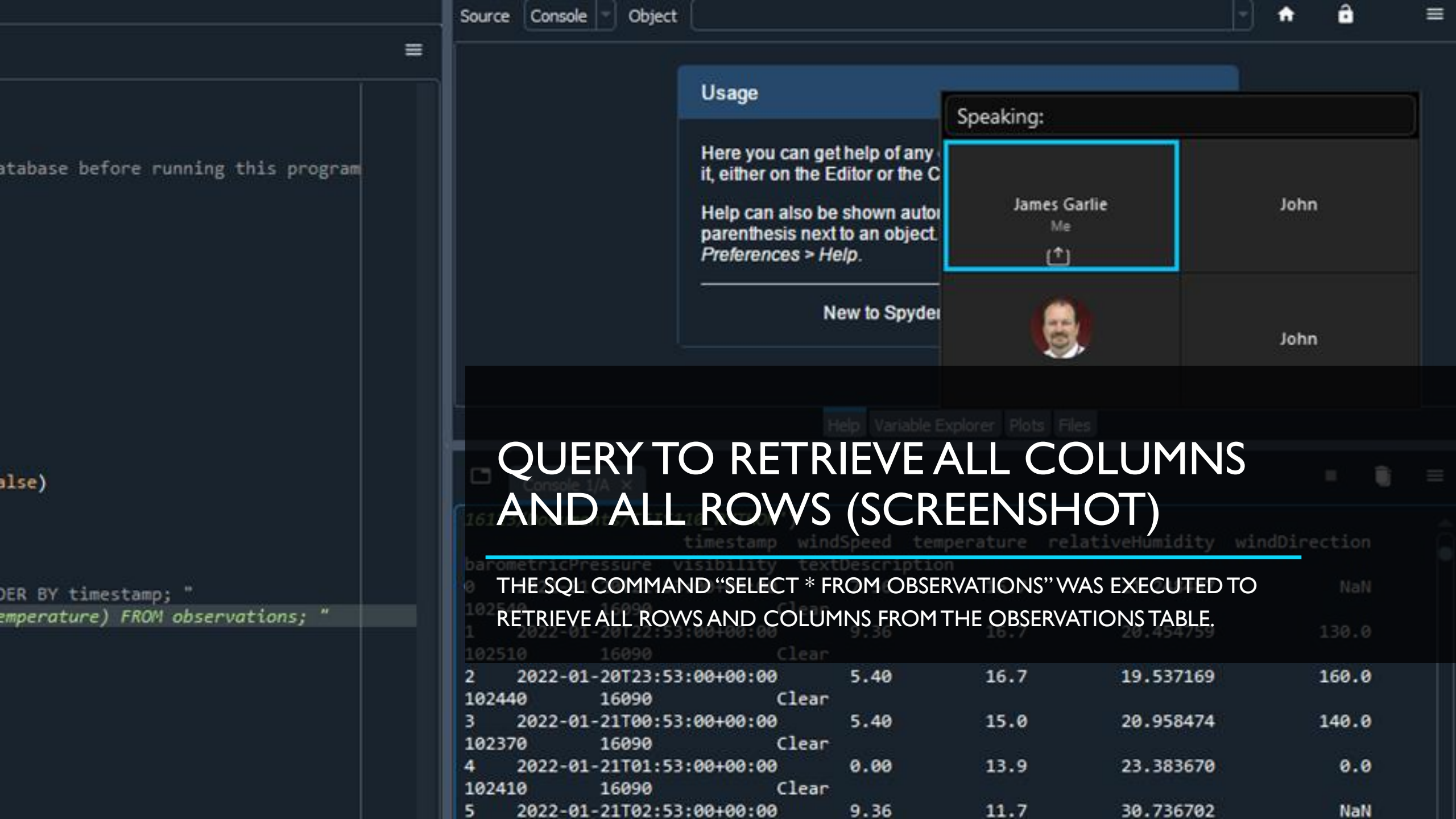
```
-- OPERATOR
```

```
types.Operator):  
    on X mirror to the selected  
    object.mirror_mirror_x"  
    mirror X"
```

QUERYING THE DATABASE

STRUCTURED QUERY LANGUAGE (SQL) IS A PROGRAMMING LANGUAGE USED FOR WORKING WITH RELATIONAL DATABASES.

SOLITESTUDIO WAS USED TO QUERY THE DATABASE AND VIEW THE RESULTS.



QUERY TO RETRIEVE ALL COLUMNS AND ALL ROWS (SCREENSHOT)

THE SQL COMMAND "SELECT * FROM OBSERVATIONS" WAS EXECUTED TO RETRIEVE ALL ROWS AND COLUMNS FROM THE OBSERVATIONS TABLE.

	timestamp	windSpeed	temperature	relativeHumidity	windDirection
0	2022-01-20T23:53:00+00:00	5.40	16.7	19.537169	160.0
1	2022-01-21T00:53:00+00:00	5.40	15.0	20.958474	140.0
2	2022-01-21T01:53:00+00:00	0.00	13.9	23.383670	0.0
3	2022-01-21T02:53:00+00:00	9.36	11.7	30.736702	NaN

```
'C:/Users/16123/Documents/CEIS110_PYTHON/QueryWeatherDB.py', wdir  
'CEIS110_PYTHON')  
re) MAX(temperature)  
3.3          18.9
```

QUERY TO RETRIEVE LOWEST AND HIGHEST TEMPERATURES (SCREENSHOT)

THE MINIMUM AND MAXIMUM TEMPERATURES WERE RETRIEVED. THESE TEMPERATURES ARE CAPTURED BASED ON THE CELSIUS SCALE.



Console 1/A ×

129	2022-01-26T06:53:00+00:00	14.76	10.0	25.500250
102270	16090	Clear		
130	2022-01-26T07:53:00+00:00	11.16	8.9	28.770836
102300	16090	Clear		
131	2022-01-26T08:53:00+00:00	7.56	7.8	30.997984
102340	16090	Clear		
132	2022-01-26T09:53:00+00:00	7.56	6.7	33.419479
102370	16090	Clear		
133	2022-01-26T10:53:00+00:00	12.36	6.1	34.828855
102340	16090	Clear		
134	2022-01-26T11:53:00+00:00	Nan	5.0	37.588391
102410	16090	Clear		
135	2022-01-26T12:53:00+00:00	11.16	5.0	37.588391
102440	16090	Clear		
136	2022-01-26T13:53:00+00:00	9.36	4.4	37.405178
102470	16090	Clear		

QUERY TO RETRIEVE ALL CLEAR DAYS (SCREENSHOT)

ANOTHER QUERY WAS PERFORMED TO RETRIEVE ALL THE CLEAR DAYS.

IPython console

History



DATA CLEANSING

DATA OUTPUT FROM MACHINES MAY HAVE ERRORS OR EXTRANEOUS DATA. WHEN CLEANSING THE DATA, PROGRAMS CAN AUTOMATICALLY PUT IN THE FORMAT NEEDED TO BE READ BY OTHER PROGRAMS.

A PYTHON PROGRAM READING THE DATA OUTPUT BY THE PYTHON PROGRAM AND SAVING IT IN A CSV FILE SO THAT IT CAN BE READ BY EXCEL.

OFTEN DATA MUST BE CLEANSED OF SPURIOUS OR MISSING VALUES IN A DATASET. THE DATA MUST BE COMPLETE, VALID, AND STANDARDIZED.

```
#convert Celsius temperature to Fahrenheit
def convertCtoF(tempC):
    return (tempC*9.0/5.0) + 32.0
```

```
#file names for database and output file
dbFile = "weather.db"
output_file_name='formatdata2.csv'
```

```
#connect to and query weather database and
dbFile = "weather.db"
conn = sqlite3.connect(dbFile)
#create cursor to execute SQL commands
cur = conn.cursor()
selectCmd = """ SELECT temperature, humidity FROM weather
ORDER BY timestamp; """
```

```
cur.execute(selectCmd)
allRows = cur.fetchall()
#limit the number of rows output to half
rowCount = len(allRows)//2 # double slash does integer division
rows = allRows[rowCount:]
```

```
#write data to output file
```

EXTRACTING TEMPERATURE AND HUMIDITY USING PYTHON CODE

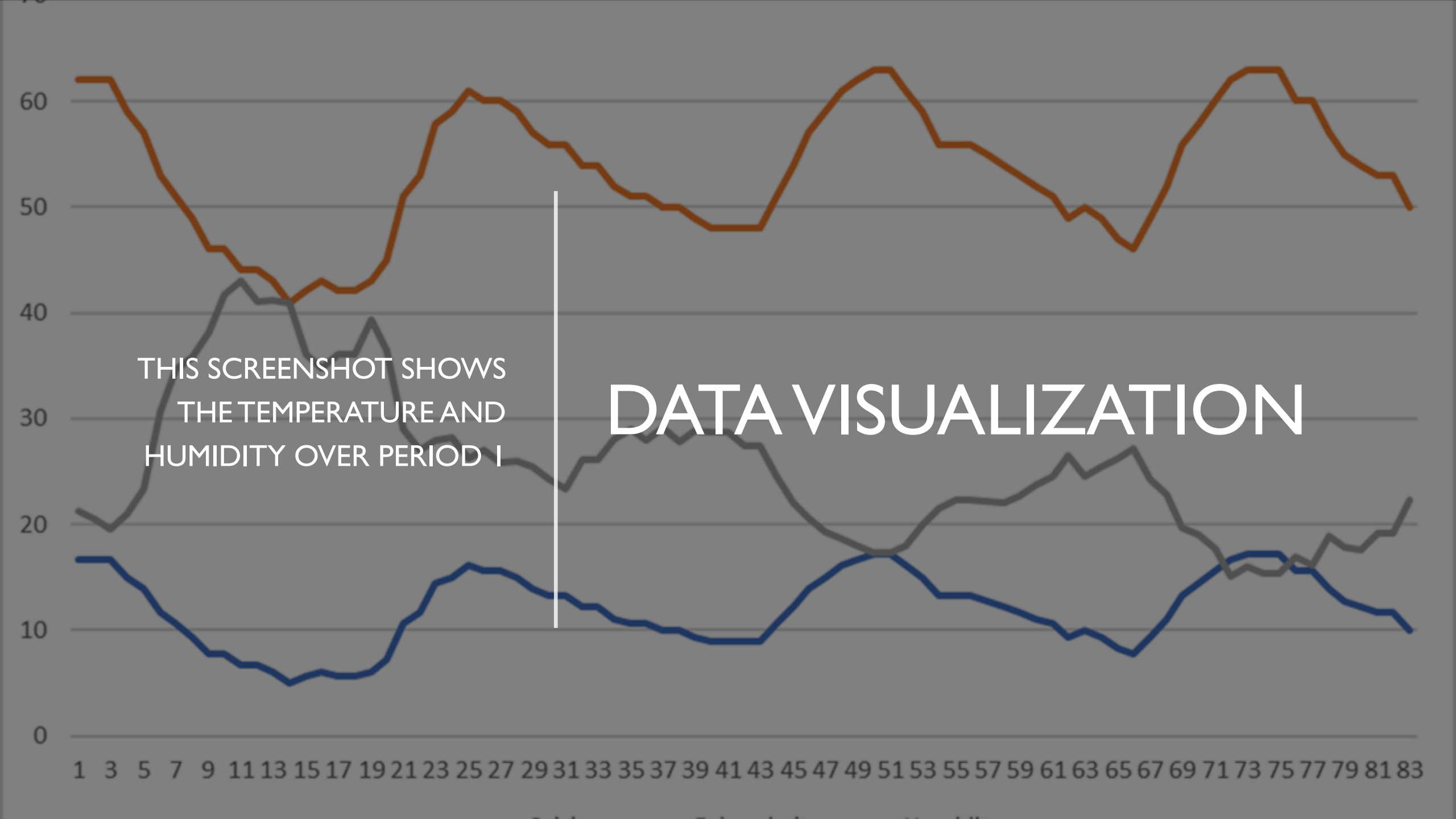
THE CODE USED RETRIEVES ONLY THE TEMPERATURE AND HUMIDITY VALUES, AND WRITES THEM TO A COMMA SEPARATED VALUES (CSV) FILE.

THE PYTHON PROGRAM
CREATED A
FORMATDATA.CSV FILE

THIS FILE CONTAINS 3
COLUMNS: CELSIUS,
FAHRENHEIT, AND
HUMIDITY
STATISTICS CAN BE
PERFORMED ON THIS
SPREADSHEET

DATA FORMATTED IN AN EXCEL SPREADSHEET

8	10.6	51.08	34.5922
9	9.4	48.92	35.82567
10	7.8	46.04	38.14626
11	7.8	46.04	41.76861
12	6.7	44.06	43.03947
13	6.7	44.06	41.12616



THIS SCREENSHOT SHOWS
THE TEMPERATURE AND
HUMIDITY OVER PERIOD I

DATA VISUALIZATION

PYTHON DATA ANALYTICS
MODULES ALLOW USERS TO
DEVELOP CHARTS AND GRAPHS
DEPICTING DATA.

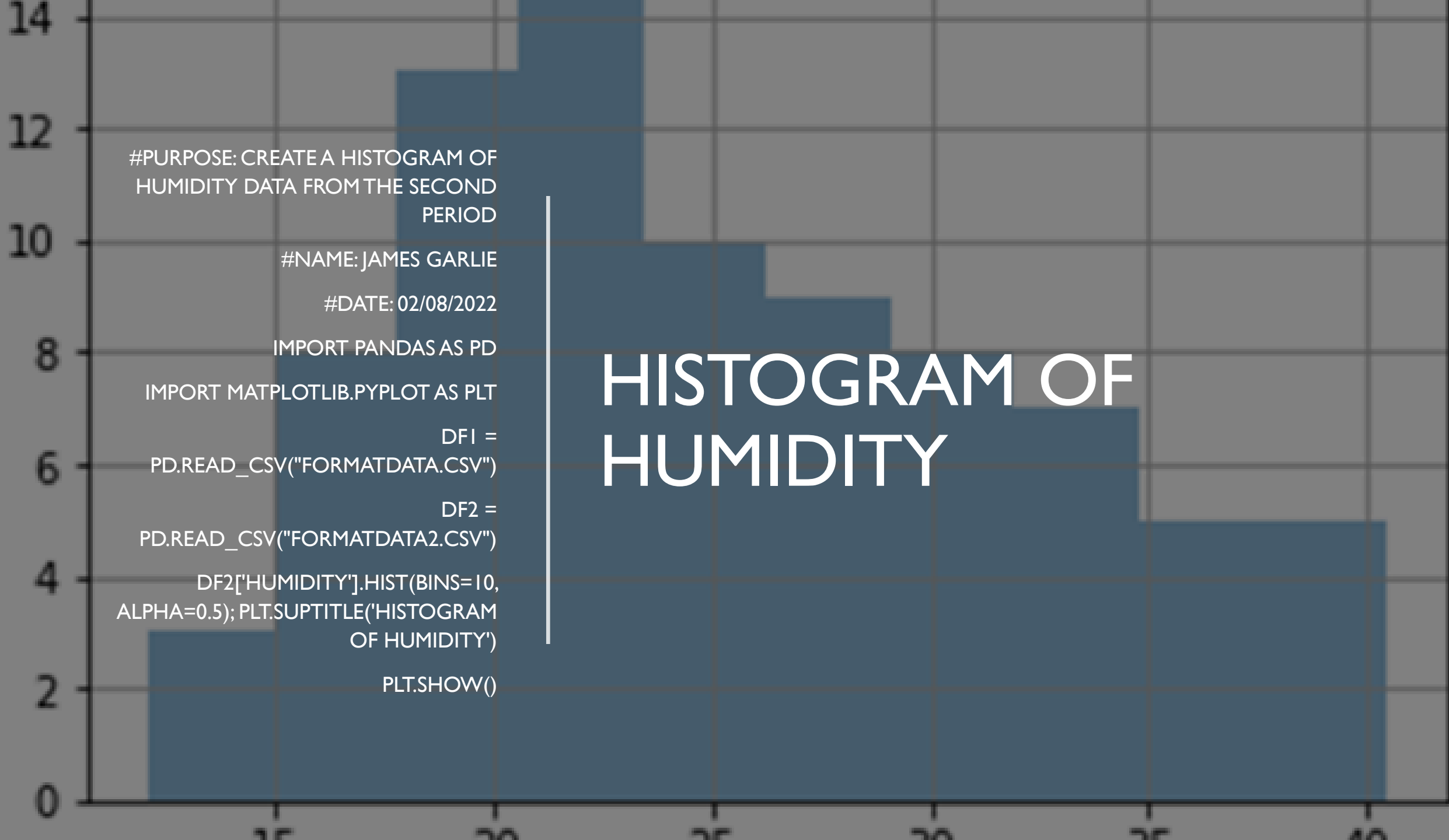
THE DATA SETS ARE MANIPULATED
AS WELL AND SAVED INTO
TABULAR FORMAT.

THE DATA ANALYTICS MODULES
ARE AVAILABLE AS PART OF
ANACONDA.

SEVERAL PLOTS WERE GENERATED
LOOKING AT HUMIDITY AND
TEMPERATURE.

THEN A PREDICTION WAS MADE
ABOUT THE DATA.

DATA ANALYTICS



```
#PURPOSE: CREATE BOX PLOT  
FOR PERIOD 2 DATA
```

```
#NAME: JAMES GARLIE
```

```
#DATE: 02/08/2022
```

```
IMPORT PANDAS AS PD
```

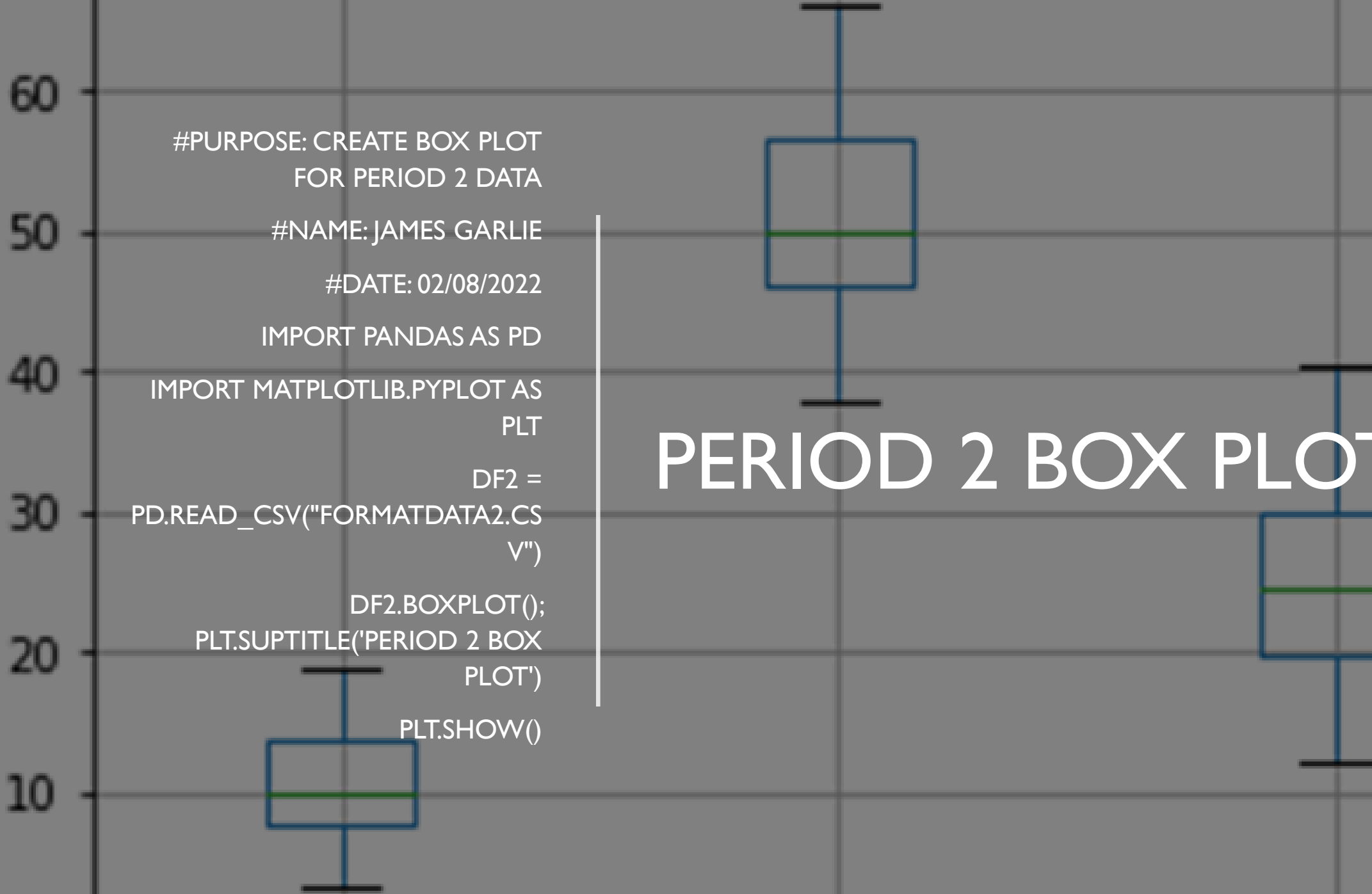
```
IMPORT MATPLOTLIB.PYPLOT AS  
PLT
```

```
DF2 =  
PD.READ_CSV("FORMATDATA2.CS  
V")
```

```
DF2.BOXPLOT();  
PLT.SUPTITLE('PERIOD 2 BOX  
PLOT')
```

```
PLT.SHOW()
```

PERIOD 2 BOX PLOT



A LINE CHART SHOWING THE FLUCTUATIONS IN FAHRENHEIT FOR PERIOD 1 AND PERIOD 2.

WHAT TEMPERATURES IN FAHRENHEIT SHOW IN THE LINE GRAPH FOR THE TWO PERIODS.

THE LINE GRAPH SHOWS THE SAME FLUCTUATIONS, BUT THE TEMPERATURE HITS A HIGH OF OVER 65 DEGREES FAHRENHEIT IN PERIOD 2.

ANALYSIS

period
period 2

A LINE CHART SHOWING THE FLUCTUATIONS IN FAHRENHEIT FOR PERIOD 1 AND PERIOD 2.

WHAT TEMPERATURES IN FAHRENHEIT SHOW IN THE LINE GRAPH FOR THE TWO PERIODS.

THE LINE GRAPH SHOWS THE SAME FLUCTUATIONS, BUT THE TEMPERATURE HITS A HIGH OF OVER 65 DEGREES FAHRENHEIT IN PERIOD 2.

ANALYSIS

period
period 2

A LINE CHART SHOWING THE FLUCTUATIONS IN FAHRENHEIT FOR PERIOD 1 AND PERIOD 2.

WHAT TEMPERATURES IN FAHRENHEIT SHOW IN THE LINE GRAPH FOR THE TWO PERIODS.

THE LINE GRAPH SHOWS THE SAME FLUCTUATIONS, BUT THE TEMPERATURE HITS A HIGH OF OVER 65 DEGREES FAHRENHEIT IN PERIOD 2.

ANALYSIS

period
period 2

A LINE CHART SHOWING THE FLUCTUATIONS IN FAHRENHEIT FOR PERIOD 1 AND PERIOD 2.

WHAT TEMPERATURES IN FAHRENHEIT SHOW IN THE LINE GRAPH FOR THE TWO PERIODS.

THE LINE GRAPH SHOWS THE SAME FLUCTUATIONS, BUT THE TEMPERATURE HITS A HIGH OF OVER 65 DEGREES FAHRENHEIT IN PERIOD 2.

ANALYSIS

period
period 2

A LINE CHART SHOWING THE FLUCTUATIONS IN FAHRENHEIT FOR PERIOD 1 AND PERIOD 2.

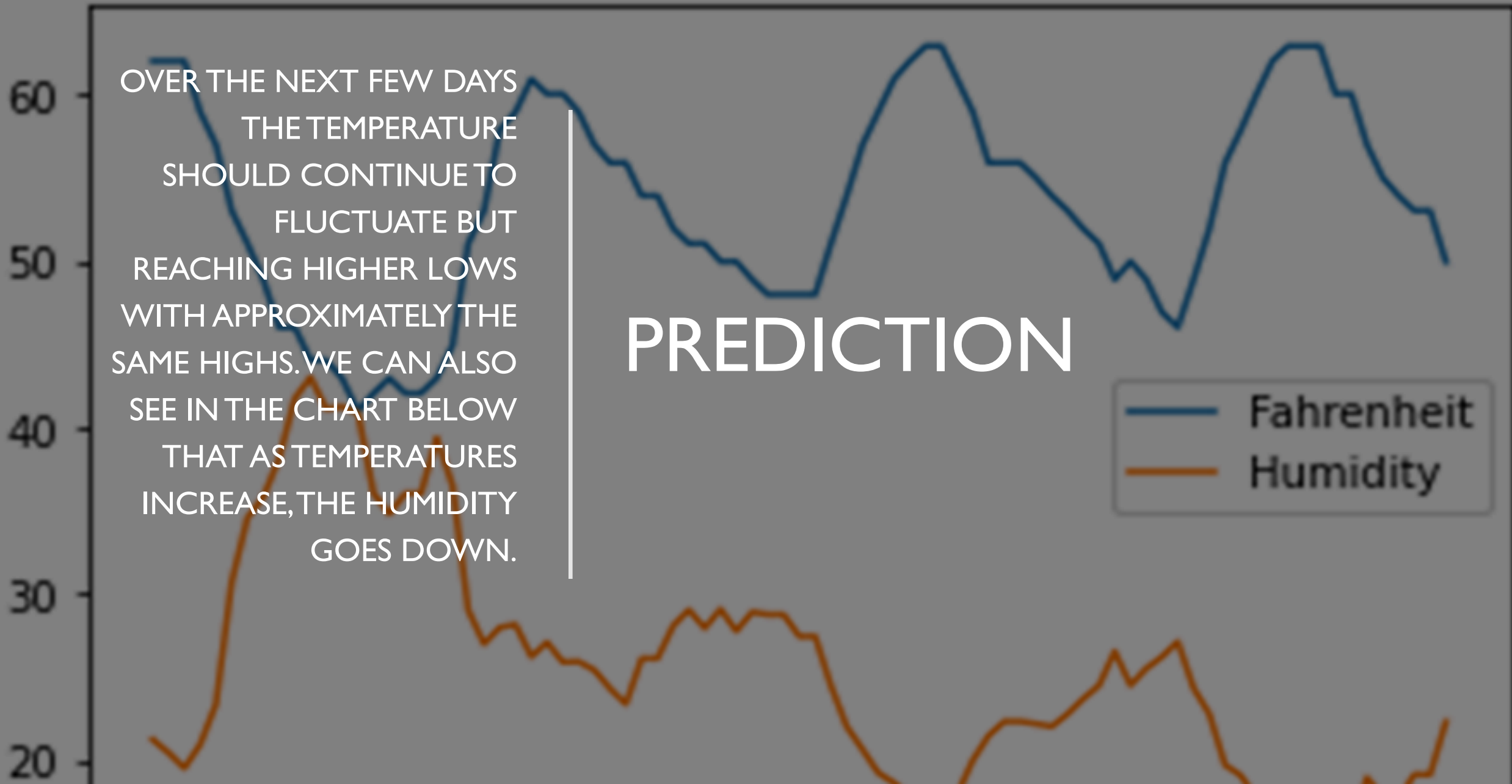
WHAT TEMPERATURES IN FAHRENHEIT SHOW IN THE LINE GRAPH FOR THE TWO PERIODS.

THE LINE GRAPH SHOWS THE SAME FLUCTUATIONS, BUT THE TEMPERATURE HITS A HIGH OF OVER 65 DEGREES FAHRENHEIT IN PERIOD 2.

ANALYSIS

period
period 2

Fahrenheit vs Humidity



WHEN CREATING THE
PYTHON PROGRAM, I NEED
TO RESET THE SPYDER TO
PRINT THE OUTPUT.
LEARNING TO EDIT THE
CODE PROPERLY TO MAKE
CHANGES.

CHALLENGES

SEVERAL CAREER SKILLS WERE
GAINED IN THIS PROJECT:

COMMUNICATION – CHARTS TO
DEPICT THE PLAN OF A PROJECT

DATABASE DEVELOPMENT

PROGRAMMING USING PYTHON

TROUBLESHOOTING ERRORS IN
THE CODE AND DATA CLEANSING

ANALYSIS – REVIEWING CHARTS
AND GRAPHS TO MAKE
PREDICTIONS ON THE DATA

CAREER SKILLS

THIS PROJECT COVERED
FUNDAMENTAL TOPICS OF
PROGRAMMING WITH
DATA BY USING DATA
GATHERING CLOUD
SERVICE TO PERFORM DATA
ANALYTICS OPERATIONS.

BUILDING THIS PROJECT
PROVIDED A HANDS-ON
LEARNING OPPORTUNITY
TO PUT INTO PRACTICE
THE TOPICS COVERED IN
THIS COURSE.

CONCLUSION