

```

# DSC510 T301
# Week 12 Weather Program
# Programming Assignment Final Project
# Moshe Burnstein
# 4/27/2022
# Access Current Weather by City or Zipcode
# Use City, Zipcode to Get Coordinates to Pass Through OpenWeatherMap API


import json
import requests
from pprint import pprint


# Change #1
# Learned how to pass variables through functions
# Allows me to omit dreaded global variables
# Date of change: May 30, 2022
# Author Moshe Burnstein


# Ensure Use of Variables Among Different Functions
#global locale
#global latitude
#global longitude


def prompt_degrees_type():
    while True:
        print("How would you like to view degrees?")
        choose_temp = input('Please enter "C" for Celsius, "F" for Fahrenheit, and "K"
for Kelvin: ')
        if choose_temp.upper() == "C":
            degrees_temp = "metric"
        elif choose_temp.upper() == "F":
            degrees_temp = "imperial"
        elif choose_temp.upper() == "K":
            degrees_temp = "standard"
        else:
            print("You have not entered a valid entry.")
            continue
        return degrees_temp


def find_by_city():
    #global latitude
    #global longitude
    locale = input("Please enter the city name: ")
    state_abbreviation = input("Please enter your state by abbreviation: ")
    url =
'https://api.openweathermap.org/geo/1.0/direct?q={}, {},US&limit=2&appid=fab5265f59e2fe91d

```

```

a54d4dfef6df75a'.format(
    locale, state_abbreviation)
try:
    response = requests.get(url)
except requests.ConnectionError as e:
    print(e)
except requests.HTTPError as e:
    print(e)
except Exception as e:
    print(e)
else:
    weather_facts = response.json()
    pprint(weather_facts)
    # for dictionary in weather_facts:
    # for key in dictionary:
    # print(dictionary[key])
    # print(dictionary['lat'])
    # print(dictionary['lon'])
    # print(weather_facts['lat'])
    # global latitude
    # latitude = dictionary['lat']
    # longitude = dictionary['lon']
    print("Attempt to parse lat, lon")
    # weather_facts = json.loads(weather_facts)
    # print(weather_facts)
    # with open(weather_facts) as json_file:
    # weather_facts.load(json_file)
    # print(type(weather_facts))
    # Determine data structure of json...this is a list of dictionaries.
    print(type(weather_facts))
    print(weather_facts[0])
    print(type(weather_facts[0]))
    print(weather_facts[0]['lat'])
    print(weather_facts[0]['lon'])
    latitude = (weather_facts[0]['lat'])
    longitude = (weather_facts[0]['lon'])
    return latitude, longitude, locale

    # pprint(weather_facts)
    # latitude = weather_facts['lat']
    # print(latitude)
    # global longitude
    # longitude = weather_facts['lon']
    # print(longitude)

def find_by_zipcode():
    locale = input("Please enter your 5 digit zipcode:")
    url =
'http://api.openweathermap.org/geo/1.0/zip?zip={},US&appid=5c59712d5d2f4678bde5bb17947f84
21'.format(locale)
    try:
        response = requests.get(url)

```

```

except requests.ConnectionError as e:
    print(e)
except requests.HTTPError as e:
    print(e)
except Exception as e:
    print(e)
# response.status_code
data = response.json()
pprint(data)
#global latitude
latitude = data['lat']
print(data['lat'])
# global longitude
longitude = data['lon']
# latitude = data['lat']
# if 'lat' in data:
# print("Latitude is:", latitude)
# latitude = data['lat']
# print(latitude)
longitude = data['lon']
print(longitude)
return latitude, longitude, locale

def get_weather_info(
    latitude, longitude, degrees_temp, units
):
    # latitude = find_by_zipcode()
    # longitude = find_by_zipcode()
    # degrees_temp = input('Please enter "imperial" for Fahrenheit, "metric" for Celsius,
and "standard" for Kelvin: ')
    url =
'https://api.openweathermap.org/data/2.5/weather?lat={}&lon={}&appid=5c59712d5d2f4678bde5
bb17947f8421&units={}'.format(
        latitude, longitude, units
    )
    try:
        response = requests.get(url)
    except requests.ConnectionError as e:
        print(e)
    except requests.HTTPError as e:
        print(e)
    except Exception as e:
        print(e)
    else:
        current_weather = response.json()
        pprint(current_weather)
        print(type(current_weather))
        #print("Currently there is a ", current_weather['weather']['description'])
        print("        Current weather conditions for",
              current_weather['weather'][0]['description'].capitalize(), ":")
        print("        -----")
        print("The temperature is ", current_weather['main']['temp'], "degrees.")
        print("The high is ", current_weather['main']['temp_max'], "degrees.")
        print("The low is ", current_weather['main']['temp_min'], "degrees.")
        print("It feels like ", current_weather['main']['feels_like'], "degrees.")
        print("The humidity is ", current_weather['main']['humidity'], "percent.")

```

```

print("The barometric pressure is ", current_weather['main']['pressure'], "hPa")
print("The current wind speed is ", current_weather['wind']['speed'], "m/s.")
print("The visibility is ", current_weather['visibility'], "meters.")
# Must index for list after first dictionary
print("Current weather conditions: ", current_weather['weather'][0]['description'])
# for dict in current_weather:
# for key in dict:
# print(key)
# print("Currently there is a ", dict['weather']['description'])
# print("The current wind speed is ", dict['wind']['speed'], "m/s.")

def main():
    print("Welcome to my weather program. It would be my pleasure to provide you with
weather facts.")
    #global degrees_temp
    #global latitude
    #global longitude
    # degrees_temp = input('Please enter "imperial" for Fahrenheit, "metric" for Celsius,
and "standard" for Kelvin: ')
    #if degrees_temp.lower() == "imperial" or "metric" or "standard":
    # print("You entered a proper standard.")
    # degrees_temp = input(
    # 'Please enter "imperial" for Fahrenheit, "metric" for Celsius, and "standard"
for Kelvin: ')
    # if degrees_temp.lower() != "imperial" or "metric" or "standard":
    # print("Make new choice.")
    # degrees_temp = input(
    # 'Please enter "imperial" for Fahrenheit, "metric" for Celsius, and "standard"
for Kelvin: ')
    another_weather_query = 'Y'
    # print('You chose', choose_locale)
    while another_weather_query.upper() == "Y":
        try:
            choose_locale = int(
                input('Please enter "1" to search by U.S. city, "2" to search by zipcode,
and "3" to finish:'))
        except ValueError as e:
            print("You did not enter a valid number,", e)
            continue
        if choose_locale == 1:
            print("You have chosen to search by city.")

            #find_by_city()
            latitude, longitude, locale = find_by_city()
            # Change #2
            # May 31,2022
            # Originally prompted for degrees metric once for program
            # Assumed user would want only one standard
            # Professor want option for each rerun...maybe user wishes to compare
            # Author Moshe Burnstein
            degrees_temp = prompt_degrees_type()
            get_weather_info(latitude, longitude, locale, degrees_temp)

```

```
if choose_locale == 2:
    print("You have chosen to search by zipcode.")

    #find_by_zipcode()
    latitude, longitude, locale = find_by_zipcode()
    degrees_temp = prompt_degrees_type()
    get_weather_info(latitude, longitude, locale, degrees_temp)

if choose_locale == 3:
    print("Thank you for using my program.")
    break
# get_weather_info()
# another_weather_query.upper() == input("\nWould you like to request another
location's weather?")
if another_weather_query != "Y":
    break

if __name__ == "__main__":
    main()
```