Tomine Bergseth

ITP-116: Python for Programmers

Final Project: The Global CO2 Tracker

- Proposal - 2.0% of course grade
- Implementation – 18.0% of course grade

Goals

- Utilize an API
- Utilize json formatting techniques
- Utilize Tkinter to create a GUI for the user to run/utilize the program
- Utilize object-oriented programming
- Utilize functions
- Utilize dictionaries
- Utilize lists and list methods
- Utilize if statements
- Utilize loops to repeat certain code blocks

Overall Description

- The program will appear as a GUI to the user where they input two countries. Once the program collects this input, it will connect to the Climatiq API and retrieve data on those countries' CO2 emissions in for electricity generation – WWT in 2021. The data source behind this measure is BEIS. The program will then determine which country had the highest emissions and output each country's emissions in corresponding colors as well as the difference between these numbers as a percentage.
- Please run the program in PyCharm and make sure you access the "globe" image in the folder to successfully run the program and implement the picture.
- The API I will use: https://www.climatiq.io/#scroll-down
- Link to the electricity generation measure: https://www.climatiq.io/explorer?category=Electricity&year=2021

Tomine Bergseth

Requirements

- Create a new Python file. It must begin with comments in the following format (replace the name and email with your actual information and write text for the description):
- # Tomine Bergseth, bergseth@usc.edu

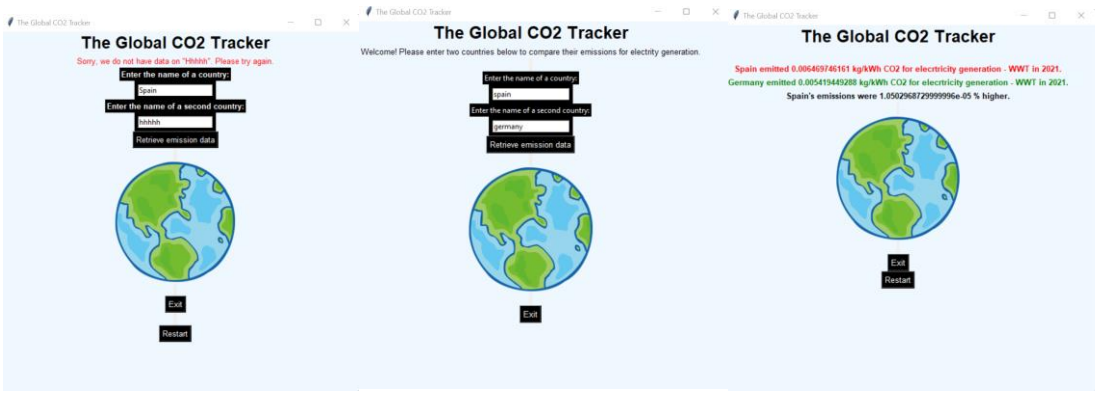
  # ITP 116, Spring 2022, 11-11.50am

  # Final Project

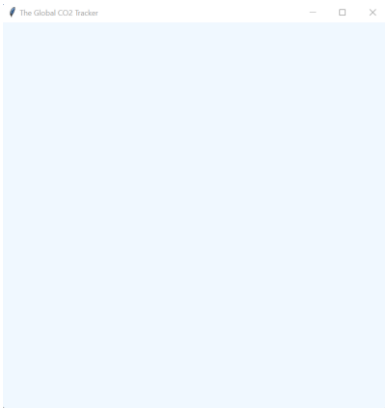
  # Description:
- The program will appear as a GUI to the user.
- The GUI will appear as a light blue screen with a globe on it. At first, it will have a title and welcome text and two entry boxes where the user can enter their country requests and subsequent labels asking the user to do so. Below that, there will be a button to retrieve the emission data and finally, an exit button to exit the program.
- The entry boxes and subsequent labels will ask the user to input two country requests for which they would like to receive data on CO2 emissions.
  - Accept country name input case insensitively. The program needs to ensure that the country name is formatted correctly – i.e. Saudi Arabia instead of Saudi arabia.
  - The program will validate whether the API has data on the country requested. This will prompt error messages for invalid country names as well as countries for which the API does not have data, The Climatiq API is fairly limited, and I believe it only has data on about 62 countries for the "Energy Generation – WWT" measure I am using.
    - If the program cannot access data on the country input request, the program will display an error message and the user will need to enter the country names again to run the program.
- When the user successfully enters two country names and clicks the retrieve emission data button, the program will display each countries CO2 emissions for electricity generation – WWT in 2021. The information for the country with the highest CO2 emission will be written in red and the information for the country with the lowest emissions will be written in green.
  - The program will also calculate the difference between the two countries emissions and display the country with the highest emissions as well as how much more CO2 that country emits as a percentage.

- After the program has output the results in the GUI, the user can either choose to click the restart button to start over and get data on two new countries or click the exit button to exit the program.

Sample Output



Tomine Bergseth



Deliverables and Submission Instructions

- Create a folder on your computer called **ITP499_*FinalProject*_LastName_FirstName** (replace *LastName* with your last/family name and *FirstName* with your first name).
- Inside the folder, put your Python source code.
- Compress the folder (make a zip file). This cannot be done within PyCharm. Find the folder on your computer and compress it.

  - Windows:
    - Select your file
    - Right click
    - Send to ->
    - Compressed (zipped) folder
  - macOS:
    - Select your file
    - Right click
    - Compress
- Upload the zip file to your Blackboard section:
  - On Blackboard, click on the Assignments item in the course menu on the left.
  - Click on the specific item for this assignment.
  - Click on the Browse My Computer button and select your zip file.
  - Click the Submit button.

Grading Criteria

- 3% Connecting to the API
- 1% Calling the API for region names and the total CO2 value for the electricity generation – WWT measure and storing it using dictionaries
- 1% Setting up the GUI
- 0.5%Creating a class object using init and self for the program
- 0.5%Using functions to write the program
- 0.5% Changing background colors, font colors types and sizes in the GUI
- 0.5% Creating a welcome text label in the GUI
- 0.5% Creating two separate entry boxes for the user to input two countries
- 0.5% Creating labels above the entry boxes that prompt the user to enter country names

Tomine Bergseth

- 0.5% Creating a button for retrieving emission data
- 0.5% Successfully accessing the text input in the entry fields and saving each input in a country variable in the program
- 0.5% Accepting country name inputs case insensitively
- 1% Formatting the country name correctly
- 0.5% Validating if the API has data on the country names requested
- 0.5% Displaying an error message if the API does not have data on the country input (including for invalid inputs, empty fields, and countries the API lacks data on)
- 0.5% Requiring the user to enter two valid country names before displaying output results
- 0.5% Removing the unnecessary entry boxes and input labels for the start page when the user clicks the retrieve data button
- 0.5% Displaying the countries and their subsequent CO2 emissions when the user clicks the retrieve data button
- 0.5% Calculating the difference in emissions between the two requested countries
- 1% Determining which country had the highest emissions and by how much it was a higher as a percentage and displaying the outcome along with the actual CO2 emissions for each country
- 1% Displaying the information for the country with the highest emissions in red and the information for the country with the lowest emissions in green (if they are equal, it will be displayed in yellow; however, this is not a part of the grading requirement as it is an unrealistic scenario)
- 0.5% Displaying an image of a globe as part of the design in my GUI
- 1% Resizing the globe image to fit the GUI screen as desired
- 0.5% Creating a restart button that restarts the program when clicked
- 0.5% Creating an exit button that successfully exits the program when clicked