How the Code works

File Name: weather.py

Functions

def weatherStatus():

1. Lines [12-15] - Parses the html on the page and finds the classes based on the html tag and class name for weathertype, location, temp, and time. This is done using beautifulsoup4. Specifically the text part the html is taken.

```
12    weatherType = soup.find('div', class_="wob_dcp").text #find the specified div and class and save the text to the waetherType variable
13    location = soup.find('div',class_="wob_loc mfMhoc").text #find the specified div and class and save the text to the location variable
14    temp = soup.find('span', class_="wob_t TVtOme").text+'°F' #find the specified span and class and save the text to the temp variable
15    time = soup.find('div', class_="wob_dts").text  #find the specified span and class and save the text to the temp variable
```

2. Lines [17-20]- Populates the GUI with the information that was just parsed.

```
17    locationTypography.config(text=location) #Populates locaton element on gui
18    tempTypography.config(text=temp) #Populates temperature element on gui
19    weatherTypeLabel.config(text=weatherType) #Populates weathertpe element on gui
20    timelabel.config(text=time) #Populates time elementon gui
```

3. Lines [22-24]- Updates the temperature and time according to the set time for updates

```
tempTypography.after(120000,weatherStatus)#updates the weather inforamtion every 2 minutes
timelabel.after(900000,weatherStatus)#looks to update the time inforamtion every 15 minutes
master.update()
```

User Input

[Lines 26-31]- The user inputs the:

1. City:
2. State:
3. County:

Each data point is saved to a variable to create a unique google query. The url of the queried result is then saved to a variable and then the html page is then parsed.

```
26  locationInput_town= input('Enter City: ') # user inputs City
27  locationInput_State= input('Enter State: ') # user inputs State
28  locationInput_country= input('Enter Country: ') # user inputs State
29  url = "https://www.google.com/search?q="+locationInput_town+'+'+locationInput_State+'+'+locationInput_country+'+'+"weather" #url for search query
30  page =requests.get(url) #fetch html page to parse
31  soup =BeautifulSoup(page.content, "html.parser") #html page to parse through
```

Weather Image upload

Weather image/icon are saved on a local drive and are used later in the conditional statements based on weather type. The fields are fetched from your local drive and resized to fit appropriately.

Gui Design

Lines[63-71] - The GUI design and placement of the Weather type, temperature, location, time, and title are created.

```
63  #Location design
64  locationTypography = Label(master, font=("Calibri bold",20),bg ="white") # text design bg=background
65  locationTypography.grid(row=0, sticky = "N", padx=100) #location on the Guiin a table like foramt, sticky = compass direction of widget
66  #times design
67  timelabel = Label(master, font = ("Calibri", 11),bg = "white") # text design
68  timelabel.grid(row=1,sticky="N", padx = 20) #location on the Gui
69  #temperature design
70  tempTypography = Label(master, font = ("Calibri bold", 50),bg = "white") # text design
71  tempTypography.grid(row=1,sticky="W", padx = 20) #location on the Gui
```

Conditional flow statements

[Lines 74-85] - The text for the weather type is assessed to output the correct image icon corresponding to the weather type.

```
73  #image design depends on type of weather
74  if weatherType2 == 'Sunny':
75      Label(master, image=Sunny,bg="white").grid(row=1,sticky="E") # insert the image of the weather icon on the east side
76  elif weatherType2 =='Windy':
77      Label(master, image=windy,bg="white").grid(row=1,sticky="E") # insert the image of the weather icon on the east side
78  elif weatherType2 =='Cloudy':
79      Label(master, image=cloudy,bg="white").grid(row=1,sticky="E") # insert the image of the weather icon on the east side
80  elif weatherType2 =='Partly Cloudy':
81      Label(master, image=partlycloudy,bg="white").grid(row=1,sticky="E") # insert the image of the weather icon on the east side
82  elif weatherType2 =='Snow':
83      Label(master, image=snow,bg="white").grid(row=1,sticky="E") # insert the image of the weather icon on the east side
84  else:
85      Label(master, image=rain,bg="white").grid(row=1,sticky="E") # insert the image of the weather icon on the east side
```

Summary

Start with a paragraph summarizing the purpose or goal of your project, the technologies used, and the results. The purpose of our project is to create an app that will display the weather for any location that is input. The user would first input the location in which they want to know the weather for, and then the application will grab that information from a weather website and display it. Our project does just that so we would say that it is successful at what it does. This can be used to find the weather anywhere in the world while displaying a corresponding weather image and prediction that shows the current weather. The information updates every 2 minutes in terms of temperature as well.

More ideas

What would you do next if you had to (or chose to) do a follow-up project? We would make a search box that the user can input the location into instead of

having them input it into the terminal. We would also make it so that the application would display the forecast over the next couple of days instead of just displaying 1 day.