

```

from tkinter import *
import requests
import json
from datetime import datetime

#Initialize Window

root =Tk()
root.geometry("400x400") #size of the window by default
root.resizable(0,0) #to make the window size fixed
#title of our window
root.title("Weather App - AskPython.com")

# -----Functions to fetch and display weather info
city_value = StringVar()

def time_format_for_location(utc_with_tz):
    local_time = datetime.utcfromtimestamp(utc_with_tz)
    return local_time.time()

city_value = StringVar()

def showWeather():
    #Enter you api key, copies from the OpenWeatherMap dashboard
    api_key = "7ef53fee3cfff1ec78bb3d7f4e759cb5" #sample API

    # Get city name from user from the input field (later in the code)

```

```
city_name=city_value.get()
```

```
# API url
```

```
weather_url = 'http://api.openweathermap.org/data/2.5/weather?q=' + city_name +  
'&appid='+api_key
```

```
# Get the response from fetched url
```

```
response = requests.get(weather_url)
```

```
# changing response from json to python readable
```

```
weather_info = response.json()
```

```
tfield.delete("1.0", "end") #to clear the text field for every new output
```

```
#as per API documentation, if the cod is 200, it means that weather data was successfully fetched
```

```
if weather_info['cod'] == 200:
```

```
    kelvin = 273 # value of kelvin
```

```
#-----Storing the fetched values of weather of a city
```

```
    temp = int(weather_info['main']['temp'] - kelvin)           #converting default kelvin  
    value to Celcius
```

```
    feels_like_temp = int(weather_info['main']['feels_like'] - kelvin)
```

```
    pressure = weather_info['main']['pressure']
```

```
    humidity = weather_info['main']['humidity']
```

```
    wind_speed = weather_info['wind']['speed'] * 3.6
```

```
    sunrise = weather_info['sys']['sunrise']
```

```
    sunset = weather_info['sys']['sunset']
```

```
    timezone = weather_info['timezone']
```

```
cloudy = weather_info['clouds']['all']
```

```
description = weather_info['weather'][0]['description']
```

```
sunrise_time = time_format_for_location(sunrise + timezone)
```

```
sunset_time = time_format_for_location(sunset + timezone)
```

#assigning Values to our weather variable, to display as output

```
weather = f"\nWeather of: {city_name}\nTemperature (Celsius): {temp}°\nFeels like in (Celsius):  
{feels_like_temp}°\nPressure: {pressure} hPa\nHumidity: {humidity}%\nSunrise at {sunrise_time} and  
Sunset at {sunset_time}\nCloud: {cloudy}%\nInfo: {description}"
```

```
else:
```

```
weather = f"\n\tWeather for '{city_name}' not found!\n\tKindly Enter valid City Name !!"
```

```
tfield.insert(INSERT, weather) #to insert or send value in our Text Field to display output
```

#-----Frontend part of code - Interface

```
city_head= Label(root, text = 'Enter City Name', font = 'Arial 12 bold').pack(pady=10) #to generate  
label heading
```

```
inp_city = Entry(root, textvariable = city_value, width = 24, font='Arial 14 bold').pack()
```

```
Button(root, command = showWeather, text = "Check Weather", font="Arial 10", bg='lightblue',  
fg='black', activebackground="teal", padx=5, pady=5 ).pack(pady= 20)
```

#to show output

```
weather_now = Label(root, text = "The Weather is:", font = 'arial 12 bold').pack(pady=10)
```

```
tfield = Text(root, width=46, height=10)
```

```
tfield.pack()
```

```
root.mainloop()
```