

Temperature Converter – Python Project

About

The "Temperature Converter" project is a Python-based application with a graphical user interface that allows users to convert temperatures between Fahrenheit, Celsius, and Kelvin. The application provides a convenient way to perform temperature conversions without the need for any third-party applications.

Prerequisites

To run this project, you need to have the following installed:

- Python (version 3.x recommended)

The project relies on the standard Python library Tkinter for creating the user interface.

Step By step guide To Build the Converter

- **Importing Tkinter Library**

Start by importing the Tkinter library at the beginning of your script:



```
1 #importing Tkinter and necessary libraries for the project
2 from tkinter import *
3 from tkinter import messagebox
4
```

- **Defining the Window**

Create the main window by initializing a Tkinter object, setting a title, and defining its size and size restriction:



```
1 #defining Window
2 root=Tk()
3 root.geometry("300x400")
4 root.title("Temp Converter")
5 root.resizable(0,0)
```

- **Creating a List of Tuples**

Define a list of tuples containing temperature units and their corresponding values (Fahrenheit, Celsius, Kelvin):



```
1 Units=[  
2     ("Fahrenheit","F"),  
3     ("Celsius","C"),  
4     ("Kelvin","K")  
5 ]
```

- **Creating Radio Buttons for Conversion From**

Generate radio buttons to allow the user to choose the "Conversion From" temperature unit using a loop:



```
1 #defining variable for radio button values  
2 b = StringVar()  
3 t= StringVar()  
4 b.set(0)  
5 t.set(0)
```



```
1 #choosing base  
2 Base=Label(root,text="Choose Conversion From:").grid(row=0,sticky=W)  
3 #using for loop to generate radiobuttons  
4 pos=1  
5 for text,val in Units:  
6     Radiobutton(root,text=text,value=val,variable=b).grid(row=pos,sticky=W)  
7     pos=pos+1  
8
```

- **Creating Radio Buttons for Conversion To**

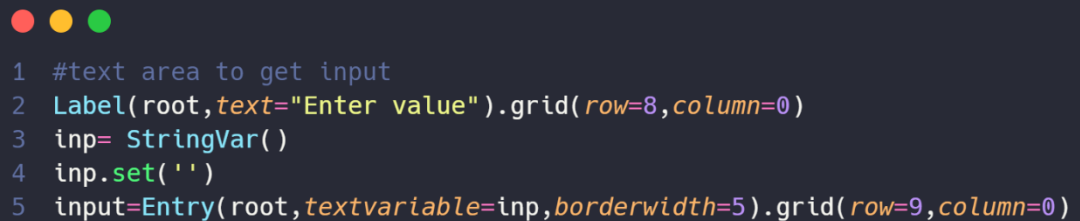
Similarly, use a loop to generate radio buttons for the "Conversion To" temperature unit:

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in Python and uses syntax highlighting. It shows a loop that iterates over units and creates radio buttons for each unit.

```
1 #choosing conversion to
2 To=Label(root,text="Choose Conversion To:").grid(row=4,sticky=W)
3 #using for loop to generate radiobuttons
4 pos=5
5 for text,val in Units:
6
7     Radiobutton(root,text=text,value=val,variable=t).grid(row=pos,sticky=W)
8     pos=pos+1
```

- **Adding Input Field**


Create an input field to get the temperature value from the user:

A code editor window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in Python and uses syntax highlighting. It shows the creation of a label for the input field and an Entry widget for the user to enter a value.

```
1 #text area to get input
2 Label(root,text="Enter value").grid(row=8,column=0)
3 inp= StringVar()
4 inp.set('')
5 input=Entry(root,textvariable=inp,borderwidth=5).grid(row=9,column=0)
```

- **Implementing the Conversion Functions**

Define functions to convert the input temperature from one unit to another:



```

1  #conversion functions
2  def FtoC(value):
3      return ((value-32)*5/9)
4  def CtoF(value):
5      return((value*9/5)+32)
6  def CtoK(value):
7      return (value+273.15)
8  def KtoC(value):
9      return (value-273.15)
10 def FtoK(value):
11     return((value-32)*5/9+273.15)
12 def KtoF(value):
13     return((value-273.15)*9/5+32)

```

- **Adding Output Field**

Create an output field to display the converted temperature:



```

1  #OUTPUT Field
2  Label(root,text="Answer").grid(row=10,column=0)
3  out= StringVar()
4  output=Entry(root,textvariable=out,borderwidth=5).grid(row=11,column=0)

```

- **Defining the Conversion Logic**

Create a function to perform the temperature conversion based on user input:

```
1  #defining converter logic
2  def click():
3      out.set('')
4      try:
5          val=float(inp.get())
6      except Exception as e:
7          messagebox.showerror("Error","NOT A NUMBER")
8
9      base=b.get()
10     to=t.get()
11     if(base==0 or to==0):
12         messagebox.showerror("Error","CHOOSE CONVERSIONS")
13     else:
14         if(base==to):
15             out.set(val)
16         if(base=='F' and to=='C'):
17             ans=FtoC(val)
18             ans="%.2f" % ans
19             out.set(ans)
20         if(base=='C' and to=='F'):
21             ans=CtoF(val)
22             ans="%.2f" % ans
23             out.set(ans)
24         if(base=='C' and to=='K'):
25             ans=CtoK(val)
26             ans="%.2f" % ans
27             out.set(ans)
28         if(base=='K' and to=='C'):
29             ans=KtoC(val)
30             ans="%.2f" % ans
31             out.set(ans)
32         if(base=='F' and to=='K'):
33             ans=FtoK(val)
34             ans="%.2f" % ans
35             out.set(ans)
36         if(base=='K' and to=='F'):
37             ans=KtoF(val)
38             ans="%.2f" % ans
39             out.set(ans)
```

- **Adding Calculate and Clear Buttons**


Create buttons to trigger the conversion and clear all entries:



```
1 #Calculate button
2 button1=Button(root,text='Calculate',command=click,padx=20,pady=10).grid(row=12,column=0)
3
4 #Clear button
5 button2=Button(root,text='Clear',command=clear,fg="white",bg="red").grid(row=12,column=1)
6
```

- **Defining the Clear Function**

Implement the function to clear all entries and reset the interface:



```
1 #function to clear
2 def clear():
3     b.set(0)
4     t.set(0)
5     inp.set('')
6     out.set('')
```

- **Running the Application**

Finally, start the main event loop to run the application:



That's it! With these steps, you have successfully built a Temperature Converter application using Tkinter in Python. The application provides a user-friendly interface to convert temperatures between Fahrenheit, Celsius, and Kelvin. Enjoy using your Temperature Converter!

Using the Application

The screenshot shows a window titled "Temp Conv...". Inside, there are two sections: "Choose Conversion From:" and "Choose Conversion To:". Each section has three radio buttons for Fahrenheit, Celsius, and Kelvin. In the "From" section, Fahrenheit is selected. In the "To" section, Celsius is selected. Below these is a text input field labeled "Enter value" containing the number "100". Below that is a text output field labeled "Answer" containing the result "37.78". At the bottom are two buttons: "Calculate" and "Clear".

- Choose the "Conversion From" temperature unit by selecting one of the radio buttons for Fahrenheit (F), Celsius (C), or Kelvin (K).
- Choose the "Conversion To" temperature unit by selecting one of the radio buttons for Fahrenheit (F), Celsius (C), or Kelvin (K).
- Enter the temperature value you want to convert in the "Enter value" field.
- Click the "Calculate" button to perform the conversion. The converted value will be displayed in the "Answer" field.
- To perform another conversion, click the "Clear" button to reset the interface.

Summary

The "Temperature Converter" project is an interactive Python application that showcases the use of Tkinter to build a graphical user interface. The application provides a simple and convenient way to convert temperatures between Fahrenheit, Celsius, and Kelvin. Users can easily input the temperature value and choose the desired conversion units, and the application will display the converted result promptly. The project can serve as an educational tool for learning GUI programming in Python and understanding the mathematical aspects of temperature conversions.

Enjoy experimenting with different temperature conversions using the "Temperature Converter" Python project!