Title: UNIT CONVERTOR

Team Member Details:

Ayush Kale (PES2202100357)

Ayush Sisodia (PES2202100754)

Yaman Gupta (PES2202100858)

Table Of Contents

Abstract:

This project is about making a Unit Convertor for multiple dimensions like Length, Energy, Temperature and Weight. This project will help convert the given quantity into another Unit intra-dimension. This project uses If-Else loops, Dictionaries and the TK-Interface for making it an interactive and appealing tool.

Introduction:

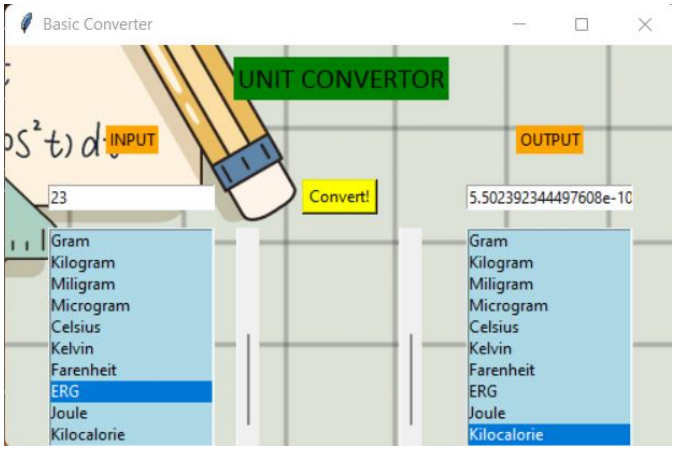
This tool will help while converting units in a complex equation to reduce the burden of the user. It will help many students and researchers that deal with the conversion of units on a regular basis.

Design & Implementation:

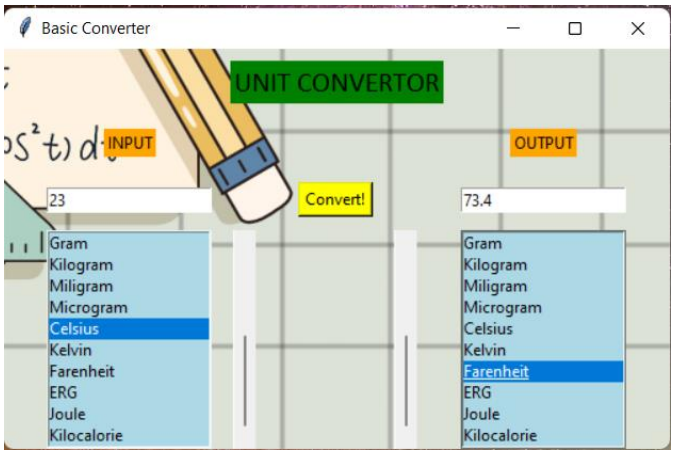
```
from tkinter import *
convertor = Tk()
convertor.title("Basic Converter")
convertor.geometry("500x300")
filename=PhotoImage(file="\\Users\\sisod\\Downloads\\math.png")
background_label=Label(image=filename)
background_label.place(x=0,y=0,relwidth=3,relheight=2)
measurement1 = ""
measurement2 = ""
def convert_SI(val, unit_in, unit_out):
    if unit_in=='Celsius' and unit_out=='Kelvin':
        return(val+273.15)
    if unit_in=='Celsius' and unit_out=='Farenheit':
        return((val*1.8)+32)
    if unit_in=='Kelvin' and unit_out=='Celsius':
        return(val-273.15)
    if unit_in=='Kelvin' and unit_out=='Farenheit':
        return(((val-273.15)*1.8)+32)
    if unit_in=='Farenheit' and unit_out=='Celsius':
        return(((val-32)*5)/9)
    if unit_in=='Farenheit' and unit_out=='Kelvin':
        return(((val-32)*5)/9+273.15)
    if unit_in=='Celsius' and unit_out=='Celsius':
        return(val)
    if unit_in=='Farenheit' and unit_out=='Farenheit':
        return(val)
    if unit_in=='Kelvin' and unit_out=='Kelvin':
        return(val)
    else:
      SI = {'Meter':1, 'Kilometer':1000, 'Centimeter':0.01, 'Millimeter':0.001,
        'Micrometer':0.000001, 'Mile':1609.35, 'Yard':0.9144, 'Foot':0.3048,
'Inch':0.0254,'Gram':1,'Kilogram':1000,'Miligram':0.001,'Microgram':0.000001,'ERG':1,'Joule':10**
7,'Kilocalorie':0.418*(10**11)}
      return val*SI[unit_in]/SI[unit_out]
def selectedInput():
    global measurement1
    measurement1 = listbox.get(listbox.curselection())
def selectedOutput():
```

```
    global measurement2
    measurement2 = listbox1.get(listbox1.curselection())
def converter():
    try:
        global measurement1, measurement2
        result.set(str(convert_SI(float(inputEntry.get()), measurement1, measurement2)))
    except:
        result.set("Error")
title = Label(convertor, text="UNIT CONVERTOR", font="Calibri 16",bg="green")
title.grid(columnspan=3)
result = StringVar()
a=Label(convertor,text="INPUT",bg="orange").grid(row=1,column=0)
b=Label(convertor,text="OUTPUT",bg="orange").grid(row=1,column=2)
inputEntry = Entry(convertor)
inputEntry.grid(row=2, column=0)
outputEntry = Entry(convertor, textvariable=result).grid(row=2, column=2)
convertButton = Button(convertor, text='Convert!', command=converter,bg="yellow").grid(row=2,
column=1)
scrollbar = Scrollbar(convertor)
scrollbar.grid(row=3, column=0, sticky = NE + SE)
listbox = Listbox(convertor, exportselection=False,bg="light blue")
listbox.grid(row=3, column=0)
measurement_list = ['Meter', 'Kilometer', 'Centimeter', 'Millimeter',
                    'Micrometer', 'Mile', 'Yard', 'Foot',
'Inch','Gram','Kilogram','Miligram','Microgram','Celsius','Kelvin','Farenheit','ERG','Joule','Kilocalori
e']
for measurement in measurement_list:
    listbox.insert(END, measurement)
listbox.bind("<<ListboxSelect>>", lambda x: selectedInput())
listbox.config(yscrollcommand=scrollbar.set)
scrollbar.config(command=listbox.yview)
scrollbar1 = Scrollbar(convertor)
scrollbar1.grid(row=3, column=1, sticky = NE + SE)
listbox1 = Listbox(convertor, exportselection=False,bg="light blue")
listbox1.grid(row=3, column=2)
for measurement in measurement_list:
    listbox1.insert(END, measurement)
listbox1.bind("<<ListboxSelect>>", lambda x: selectedOutput())
listbox1.config(yscrollcommand=scrollbar1.set)
scrollbar1.config(command=listbox1.yview)
for i in range(3):
    convertor.grid_rowconfigure(i, weight=1)
    convertor.grid_columnconfigure(i, weight=1)
convertor.mainloop()
```

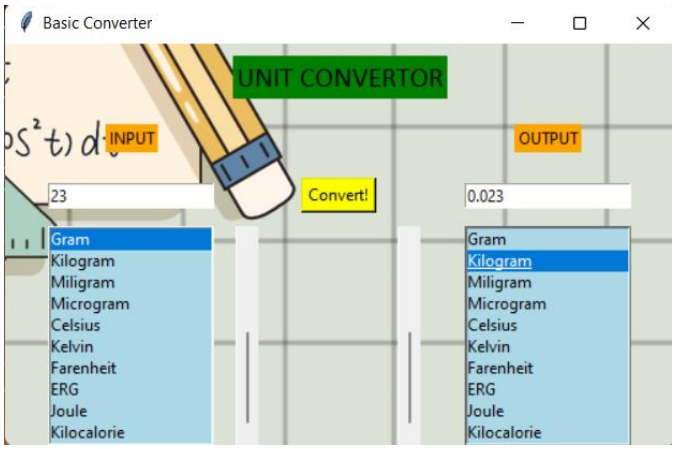
Result & Analysis:



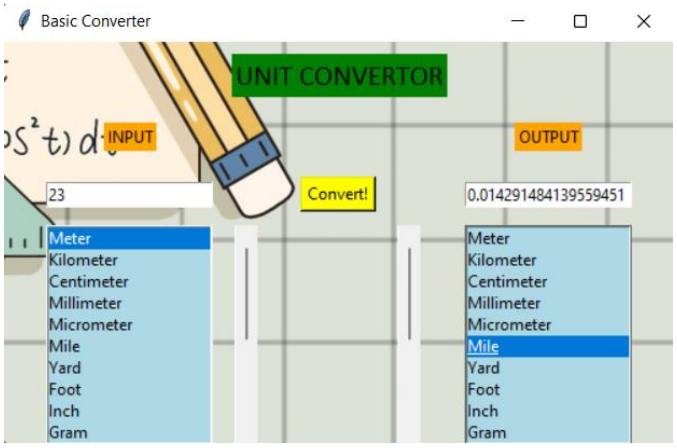
Conversion of 23 ERG to KiloCalorie (Energy)



Conversion of 23 ^{o}C to Fahrenheit (Temprature)



Conversion of 23 Grams to Kg (Weight)



Conversion of 23 Meters to Miles (Length)

Conclusions & Future Enhancement:

This tool converts units of only 4 dimensions as of now, but the future enhancements will increase the number of Dimensions and Units that can be inter-converted.