**TEMPERATURE CONVERTER**

In temperature conversions, it is often necessary to convert temperatures between different scales. One common conversion is from Celsius to Fahrenheit and vice-versa. In this article, we will see how to convert Celsius to Fahrenheit and vice-versa.

**Input1**: 40 Celsius  
**Output1**: 104.00 Fahrenheit  
**Input2**: 140 Fahrenheit  
**Output2**: 60 Celsius

We would be given the temperature in Celsius or Fahrenheit and our task will be to convert the temperature value in the Fahrenheit or Celsius scale and display it.  
To convert the temperature from Celsius to Fahrenheit or vice versa in Python firstly we will take the input in Celsius or Fahrenheit and then convert that temperature to another unit using the required formula and then print the output.

## Convert Celsius to Fahrenheit using Python

To convert Celsius to Fahrenheit we are going to use the formula “**F = (C \* 9/5) + 32″.**We will take Celsius temperature as input from the user, apply the conversion formula of Fahrenheit from Celsius, and display the result. We have used **“%.2f”** to represent the floating point number up to two decimal places in [Python](https://www.geeksforgeeks.org/python-programming-language/).

# Temperature in celsius degree

celsius **=** 47

# Converting the temperature to

# fehrenheit using the formula

fahrenheit **=** (celsius **\*** 1.8) **+** 32

# printing the result

**print**('%.2f Celsius is equivalent to: %.2f Fahrenheit'

**%** (celsius, fahrenheit))

**Output**

47.00 Celsius is equivalent to: 116.60 Fahrenheit

In the below example, we will simply take temperature input from the user in Celsius and then applied the conversion formula and after that print the result.

celsius **=** float(input("Enter temperature\

**in** celsius: "))

fahrenheit **=** (celsius **\*** 1.8) **+** 32

**print**(str(celsius )**+** " degree Celsius\

**is** equal to " **+** str(fahrenheit )**+**

       " degree Fahrenheit.")

**Output:**

Enter temperaturein celsius: 40

40.0 degree Celsiusis equal to 104.0 degree Fahrenheit.

To convert Celsius to Fahrenheit we are going to use the formula “**C = (F – 32)/1.8**“. We will take Fahrenheit temperature as input from the user, apply the conversion formula of Celsius from Fahrenheit, and display the result.

# Temperature in Fahrenheit degree

fahrenheit **=** 104

# Converting the temperature to

# fehrenheit

celsius **=** (fahrenheit**-**32)**/**1.8

# printing the result

**print**('%.2f Fahrenheit is equivalent to: %.2f Celsius'

**%** (fahrenheit ,celsius))

**Output**

104.00 Fahrenheit is equivalent to: 40.00 Celsius

In the below example, we will simply take temperature input from the user in Fahrenheit and then applied the conversion formula and after that, we print the result.

|  |
| --- |
| fahrenheit **=** float(input("Enter temperature in fahrenheit: "))    celsius **=** (fahrenheit **-** 32)**/**1.8    print(str(fahrenheit )**+** " degree Fahrenheit **is** equal\  to " + str(celsius ) + " degree Celsius." ) |

**Output:**

Enter temperature in fahrenheit: 140

140.0 degree Fahrenheit is equalto 60.0 degree Celsius.

   First I must import and initialize TKInter to use it.

**import** tkinter **as** tk

root = tk.Tk**()**

Then I will initialize a Label and an `EntryField` and also set their position using grid layout options.

input\_label = tk.Label**(**root, text="Enter temperature"**)**

input\_entry = tk.Entry**(**root, textvariable=numberInput**)**

input\_label.grid**(**row=0**)**

input\_entry.grid**(**row=0, column=1**)**

Here I need to set ‘numberInput’ to a string.

numberInput = tk.StringVar**()**

Next, I will use an OptionMenu (a drop-down menu) to display Celsius, Fahrenheit, and Kelvin temperatures and also set the dropdown default value to Celsius and set the position of the OptionMenu

dropDownList = **[**"Celsius", "Fahrenheit", "Kelvin"**]**

dropdown = tk.OptionMenu**(**root, var, \*dropDownList, command=store\_temp**)**

var.set**(**dropDownList**[**0**])**

dropdown.grid**(**row=0, column=3**)**

I also need to set var in the above snippet to a string.

var = tk.StringVar**()**

Here I need to convert the input temperature to two other temperature formats based on the input temperature format whether it is Celsius, Fahrenheit, or Kelvin unit, and store it somewhere so I can use it for conversion.

If you notice in the above snippet I used a command parameter for the OptionMenu, this command is nothing but it calls a function, here it calls store\_temp when an OptionMenu(dropdown) value is changed.

**def** store\_temp**(**sel\_temp**)**:

**global** tempVal

tempVal = sel\_temp

So here I stored the value of the selected temperature in tempVal variable which is declared globally. For declaring a variable globally, I just use a variable with some value for it.

tempVal = "Celsius"

So now every time the drop-down is changed, its value is stored in tempVal and can be accessed anywhere in our app.

#### **The temperature Converter**

So when the user inputs some value and selects the temperature from the drop-down, the next thing is to press a button that shows the desired results.

I will initialize a button with a command to call the conversion function.

So when the user inputs some value and selects the temperature from the drop-down, the next thing is to press a button that shows the desired results.

I will initialize a button with a command to call the conversion function.

result\_button = tk.Button**(**root, text="Convert", command=call\_convert**)**

result\_button.grid**(**row=0, columnspan=4**)**

I will pass 2 labels to show the result and the input value which is entered in the `Entry`. Here I use partial from functools to pass parameters to the function call\_convert.

So before I call call\_convert I will place the following line.

call\_convert = partial**(**call\_convert, result\_label1, result\_label2, numberInput**)**

and of course, I need to import `partial`

**from** functools **import** partial

So the logic resides in this call\_convert method.

#### The Conversion

First I will get the entered value, then check the drop-down value and then perform the conversion and return the result to the labels.

**def** call\_convert**(**rlabel1, rlabe12, inputn**)**:

The 3 parameters in the above line are the 2 labels to set the result and a value entered in the Entry field.

So First I will get the input value.

tem = inputn.get**()**

Now I will check the dropdown value, as I previously store this value in a global variable tempVal So now will check whether it is Celsius, Fahrenheit, or Kelvin and perform the conversions based on this value.

#for Celsius

**if** tempVal == 'Celsius':

...

...

#for Fahrenheit

**if** tempVal == 'Fahrenheit':

...

...

#for Kelvin

**if** tempVal == 'Kelvin':

...

...

So if the value is Celsius

I will use some calculations to convert and set those results to the appropriate labels.

**if** tempVal == 'Celsius':

f = float**((**float**(**tem**)** \* 9/5**)** + 32**)**

k = float**((**float**(**tem**)** + 273.15**))**

rlabel1.config**(**text="%f Fahrenheit" % f**)**

rlabe12.config**(**text="%f Kelvin" % k**)**

Here the variables fand k store the converted results based on some formulae to convert Celsius to Fahrenheit and Kelvin

Similarly for Fahrenheit and Kelvin,

**if** tempVal == 'Fahrenheit':

c = float**((**float**(**tem**)** - 32**)** \* 5 / 9**)**

k = c + 273

rlabel1.config**(**text="%f Celsius" % c**)**

rlabe12.config**(**text="%f Kelvin" % k**)**

**if** tempVal == 'Kelvin':

c = float**((**float**(**tem**)** - 273.15**))**

f = float**((**float**(**tem**)** - 273.15**)** \* 1.8000 + 32.00**)**

rlabel1.config**(**text="%f Celsius" % c**)**

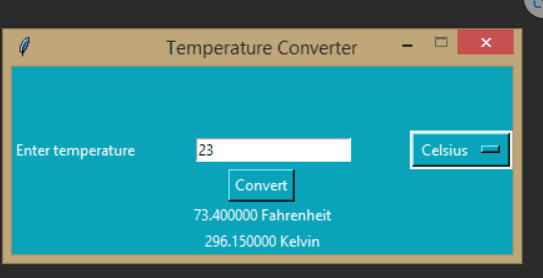
rlabe12.config**(**text="%f Fahrenheit" % f**)**

This converts the input temperature to the other two required temperatures.

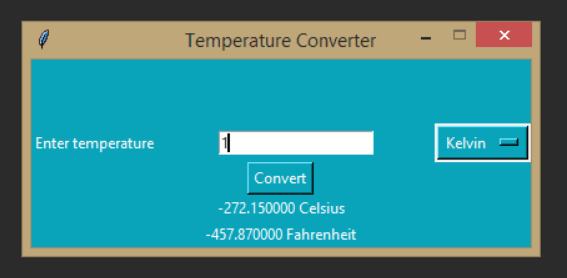
#### Finishing Touch

Here I will show you How can we set the size of the window, set colors,  background, and foreground colors to widgets, and also to set the window to a fixed size and all.

* Set the size of the window
  + root.geometry**(**'400x150+100+200'**)**
* Set the title of the app
  + root.title**(**'Temperature Converter'**)**
* Set background and foreground colors
  + root.configure**(**background='#e35237'**)**
  + input\_label = tk.Label**(**root, text="Enter temperature", background='#e35237', foreground="#FFFFFF"**)**
  + dropdown.config**(**background='#e35237', foreground="#FFFFFF"**)**
  + dropdown**[**"menu"**]**.config**(**background='#e35237', foreground="#FFFFFF"**)**
* Set the window to a fixed size
* root.resizable**(**width=**False**, height=**False)**
* \_\_author\_\_ = 'Avinash'
* **import** tkinter **as** tk
* **from** functools **import** partial
* # global variable
* tempVal = "Celsius"
* # getting drop down value
* **def** store\_temp**(**sel\_temp**)**:
* **global** tempVal
* tempVal = sel\_temp
* # the main conversion
* **def** call\_convert**(**rlabel1, rlabe12, inputn**)**:
* tem = inputn.get**()**
* **if** tempVal == 'Celsius':
* f = float**((**float**(**tem**)** \* 9 / 5**)** + 32**)**
* k = float**((**float**(**tem**)** + 273.15**))**
* rlabel1.config**(**text="%f Fahrenheit" % f**)**
* rlabe12.config**(**text="%f Kelvin" % k**)**
* **if** tempVal == 'Fahrenheit':
* c = float**((**float**(**tem**)** - 32**)** \* 5 / 9**)**
* k = c + 273
* rlabel1.config**(**text="%f Celsius" % c**)**
* rlabel1.config**(**text="%f Celsius" % c**)**
* rlabe12.config**(**text="%f Kelvin" % k**)**
* **if** tempVal == 'Kelvin':
* c = float**((**float**(**tem**)** - 273.15**))**
* f = float**((**float**(**tem**)** - 273.15**)** \* 1.8000 + 32.00**)**
* rlabel1.config**(**text="%f Celsius" % c**)**
* rlabe12.config**(**text="%f Fahrenheit" % f**)**
* **return**
* # app window configuration and UI
* root = tk.Tk**()**
* root.geometry**(**'400x150+100+200'**)**
* root.title**(**'Temperature Converter'**)**
* root.configure**(**background='#09A3BA'**)**
* root.resizable**(**width=**False**, height=**False)**
* root.grid\_columnconfigure**(**1, weight=1**)**
* root.grid\_rowconfigure**(**0, weight=1**)**
* numberInput = tk.StringVar**()**
* var = tk.StringVar**()**
* # label and entry field
* input\_label = tk.Label**(**root, text="Enter temperature", background='#09A3BA', foreground="#FFFFFF"**)**
* input\_entry = tk.Entry**(**root, textvariable=numberInput**)**
* input\_label.grid**(**row=1**)**
* input\_entry.grid**(**row=1, column=1**)**
* # result label's for showing the other two temperatures
* result\_label1 = tk.Label**(**root, background='#09A3BA', foreground="#FFFFFF"**)**
* result\_label1.grid**(**row=3, columnspan=4**)**
* result\_label2 = tk.Label**(**root, background='#09A3BA', foreground="#FFFFFF"**)**
* result\_label2.grid**(**row=4, columnspan=4**)**
* # drop down initalization and setup
* dropDownList = **[**"Celsius", "Fahrenheit", "Kelvin"**]**
* dropdown = tk.OptionMenu**(**root, var, \*dropDownList, command=store\_temp**)**
* var.set**(**dropDownList**[**0**])**
* dropdown.grid**(**row=1, column=3**)**
* dropdown.config**(**background='#09A3BA', foreground="#FFFFFF"**)**
* dropdown**[**"menu"**]**.config**(**background='#09A3BA', foreground="#FFFFFF"**)**
* # button click
* call\_convert = partial**(**call\_convert, result\_label1, result\_label2, numberInput**)**
* result\_button = tk.Button**(**root, text="Convert", command=call\_convert, background='#09A3BA', foreground="#FFFFFF"**)**
* result\_button.grid**(**row=2, columnspan=4**)**
* root.mainloop**()**



How to create a temperature converter app in Python GUI using TKInter



How to create a temperature converter app in Python GUI using TKInter

