Convertor

Team Member

- Vũ Anh Tuấn Kiệt (kiệt vũ)
- Hà Đức Thành Viên (Kevin Ha)

Description of the project

Our software provides a conversion calculator for types of measurement units (Temperature, Length, Weight).

- It is used to change the units of a measured quantity without changing its value.
- It's very easy to systematically apply the unit conversion process to solve conversions within or between measurement systems.
- It may be necessary to multiply by more than one conversion ratio in more complex conversions.

Use tkinter to create GUI apps in Python

In [20]:
```
from tkinter import *
```

Main window

In [21]:
```
App = Tk()
App.title("Converter")
App.geometry('450x180')
```

Out[21]:
```
''
```

Set Menu

In [22]:
```
def update_menu(optmenu, optvar, options):
    menu = optmenu['menu']
    # clear current options
    menu.delete(0, 'end')
    # populate new options
    for item in options:
        menu.add_command(label = item, command = lambda value = item: optvar.set(value))
    # reset selection
    optvar.set('')

def update_conversion_menus(index):
    if index == 'Temperature' :
        scalesa = ['Fahrenheit','Celsius']
    elif index == 'Weight' :
        scalesa = ['Pound','Kilogram']
    else:
        scalesa = ['Mile','Kilometer']
    update_menu(from_menu, from_var, scalesa)
    update_menu(to_menu, to_var, scalesa)
```

Menu

In [23]:
```
kind = ['Temperature', 'Length', 'Weight']
kind_var = StringVar()
all_menu = OptionMenu(App, kind_var, *kind, command = update_conversion_menus) # used command option
all_menu.grid(row = 0, column = 1, pady = 5)
```

The scale of the length to be used for conversion

In [24]:
```
from_var = StringVar()
from_menu = OptionMenu(App, from_var, None)
from_menu.grid(row = 1, column = 1, pady = 5)
```

In between label

In [25]:
```
lbl = Label(App, text = ' convert to ')
lbl.grid(row = 1, column = 2, pady = 5)
```

The scale of the length to convert the value to

In [26]:
```
to_var = StringVar()
to_menu = OptionMenu(App, to_var, None)
to_menu.grid(row = 1, column = 3, pady = 5)
```

Entry pre-label

In [27]:
```
numL = Label(App, text = 'Enter: ')
numL.grid(row = 2, column = 0, columnspan = 1, pady = 5)
```

Entry field

In [28]:
```
numE = Entry(App)
numE.grid(row = 2, column = 1, columnspan = 1, pady = 5)
```

In between Entry field and Converter function

In [29]:
```
equal = Label(App, text=' = ')
equal.grid(row = 2, column = 2, pady = 5)
```

Result of conversion

In [30]:
```
conv_numL = Label(App, width = 10)
conv_numL.grid(row = 2, column = 3, pady = 5)
```

Converter temperature function

In [31]:
```
def Convert_Temperature(froM, tO, num):
    if froM == 'Fahrenheit' and tO == 'Celsius':
        num = (num - 32) * 5 / 9
    elif froM == 'Celsius' and tO == 'Fahrenheit':
        num = (num * 9 / 5) + 32
    return num
```

Converter length function

In [32]:
```
def Convert_Length(froM, tO, num):
    if froM == 'Mile' and tO == 'Kilometer':
        num *= 1.609
    elif froM == 'Kilometer' and tO == 'Mile':
        num /= 1.609
    return num
```

Converter weight function

In [33]:
```
def Convert_Weight(froM, tO, num):
    if froM == 'Pound' and tO == 'Kilogram':
        num /= 2.205
    elif froM == 'Kilogram' and tO == 'Pound':
        num *= 2.205
    return num
```

Do convertsion

In [34]:
```
def do_conversion():
    try:
        froM = from_var.get()
        tO = to_var.get()
        num = float(numE.get().strip()) # exception may be raised on invalid input

        # do corresponding conversion based on selections
        index = kind_var.get()
        if index == 'Temperature':
            converted_num = Convert_Temperature(froM, tO, num)
        elif index == 'Length':
            converted_num = Convert_Length(froM, tO, num)
        else:
            converted_num = Convert_Weight(froM, tO, num)

        # show the conversion result
        conv_numL.config(text=round(converted_num, 4))
    except Exception as e:
        print(e)

Button(App, text='Convert', command=do_conversion).grid(row=3, column=1, pady=5)
```

Clear text function

In [35]:
```
def clear_text():
    numE.delete(0, END)
    from_var.set("")
    to_var.set("")
    kind_var.set("")
    conv_numL.config(text="")
```

Clear text button

In [36]:
```
btn = Button(App, text="Restart", command=clear_text)
btn.grid(row=3, column=3, pady=5)
```

Loop

In [37]:
```
App.mainloop()
```

Author: Hà Đức Thành Viên and Vũ Anh Tuấn Kiệt

Date: 18/10/2021

Final Project: Convertor