Example 5 – Tkinter Feet to Meters Conversion Calculator

In the previous lessons we made a basic Tkinter GUI program in Python – and assigned specific functions to buttons – to make them do things. In this example we are going to start from scratch and build a new program.

We will make a Tkinter GUI for calculating the conversion of Feet to Meters. We will be designing a new program, similar to the previous lessons and we will be making buttons with specific functions.

This time we will have an entry box and button the same as before, but we will have a different job for our widgets. Instead of just printing text to the Shell Window – we will make Python take our Entry-Box data (a length in Meters) calculate the conversion to Feet – and display the answer...ON the GUI window – not in the Shell window.

I will display the code in full first – then go through it line by line:

```python
#!/usr/bin/env/python
from Tkinter import *

top = Tk()
top.minsize(400, 200)
top.title('((Ben Woodfield))')
raser = Label(top, text='FEET to METERS Conversion Calculator', fg='purple', font='freesansBold, 14')
raser.pack()

q = Button(top, text='Exit', command=quit, fg='red')
q.pack(side=BOTTOM, fill=X)

cvt_from = StringVar()
cvt_to = StringVar()

def do_convert():
    feet_val = float(cvt_from.get())
    meters_val = feet_val * 0.3048
    cvt_to.set('%s meters' % meters_val)

from_lbl = Label(top, text='Enter FEET (numbers only):', fg='blue', font='12')
from_lbl.pack()

from_entry = Entry(top, textvariable=cvt_from)
from_entry.pack()

to_lbl = Label(top, text='Result:', fg='purple', font='12')
to_lbl.pack()

result_lbl = Label(top, textvariable=cvt_to, bg='yellow', font='freesansBold, 12')
result_lbl.pack()

convert_btn = Button(top, text='Convert to Meters', fg='blue', command=do_convert)
convert_btn.pack()


top.mainloop()
```

If you are running this directly using the code above, remember to edit line 2 appropriately. Type Tkinter for Python 2 and tkinter for Python 3 (capitol T on the import tkinter line)

Running this code should present you with a GUI window – asking you to enter your value in FEET. If you enter anything other than a number the code won't run and you will see an error message – Python is looking for a NUMBER. Enter any number in the Text Entry Box and click Calculate

You should see the result in yellow – displayed in a Label.

Below I shall explain the program in detail:

Lines 1 and 2:

```
#!/usr/bin/env/python
from Tkinter import *
```

So far we have not needed to use this first line of code – our programs have been very simple so have not needed it – and this program would work without it – I will explain what it's purpose is:

```
#!/usr/bin/env/python
```

In programming people refer to this line as “Shebang” You may not ever neeed to use it – but what it does is tell the computer about what sort of Python file it is - before you run it. So in a terminal you would have to tell the computer you are about to open a Python document with a certain version of Python (if you have more than one version) – by typing python3 program.py but if you have more than one version of Python and you just execute the code normally - your computer might automatically open files up in Python 3 or your newest version – when you might want to use Python 2.

It just tells the computer “Hey, I was written in Python 2, so run me with Python 2 please!”

If you have only **one** version of Python you will not need to bother with this line of code – but if – like me you have Python 2 and Python 3, it gets really annoying after a few times – when you click to open a code...and it opens up in the wrong version of Python and it wont run.

Before I knew how to use this line of Shebang code I used to save EVERY Python file with py2 or py3 in their file-name and also put a comment in the code to remind me which version I wrote it on.

By adding this line of code your computer will automatically know what version or Python it was written in.

Next line:

```
from Tkinter import *
```

You should be familiar with this import code by now – we are basically importing the Tkinter module.

Next we set up the basic structure of our program:

```
top = Tk()
top.minsize(400, 200)
top.title('((Ben Woodfield))')
raser = Label(top, text='FEET to METERS Conversion Calculator', fg='purple', font='freesansBold, 14')
raser.pack()
```

We are configuring the basic layout – the size of the window, the title to be shown, the main Label showing information for the end user, and pack it all in. This is a basic window – but it wouldn't run as yet – if you added the very end line to this it would: `top.mainloop()`

So nothing different to what we have covered already.

Next is the Quit Button:

```
q = Button(top, text='Exit', command=quit, fg='red')
q.pack(side=BOTTOM, fill=X)
```

We assign a name to it, add the properties / customise it and pack it in the window using `pack()`. Here I added something new. I used the `fill=x` command to expand the Quit button across the whole window.

Because a Quit button is such a common thing, Python has a built in function for it ready-made. So we can assign a `command=quit` function to the Button – without having to write an exit function ourselves.

Next we have 2 lines of code we have not come across before:

```
cvt_from = StringVar()
cvt_to = StringVar()
```

The best way I can explain is as follows:

- cvt_from is to retrieve data from the program's Entry box
- cvt_to is to send the data back to the programs Text Label

In Python these are referred to as State Variables

You will see parts of the above code in the function below:

```
def do_convert():
    feet_val = float(cvt_from.get())
    meters_val = feet_val * 0.3048
    cvt_to.set('%s meters' % meters_val)
```

Here we are writing the `do_convert():` function ready to use later. We will write a function over a few lines, then we can use this function at any time by just using it's name.

We covered functions briefly before. In our GUI from lesson 4 we had to write 2 simple functions:
They allowed us to have activity from our buttons when they are clicked. One printed a "Button Clicked" message at the Shell window – the other one printed the text entered into the Entry-box by the user, to the Shell window. The point of having a function is more beneficial when you have longer programs that may need to repeat a process numerous times. If the process is say 10 lines long it can be very time consuming to repeat those 10 lines every time you want to do a specific task. You can command python to do 10 lines of code with just 1 line of code - by defining a section of code as a function. I mentioned the Quit button didn't need a function to be written because Python already has one built-in.

We are using our new function in Lesson 5 to do the main calcultaion for our Feet to Meters conversion.

Take another look at our function def do_convert():

```
def do_convert():
    feet_val = float(cvt_from.get())
    meters_val = feet_val * 0.3048
    cvt_to.set('%s meters' % meters_val)
```

```
def do_convert():
```

Is the name of the Function

```
 feet_val = float(cvt_from.get())
```

This is saying `feet_val` (value in feet – entered by the user) – is a float number.

In Python a FLOAT or a Floating Point Number is a Decimal point.

It means a person can enter a number like 2.5 instead of just 2 or 3 – allowing more accuracy.
So what the line of code is saying is that: `feet_val` (value in feet) is a floating point number, and `cvt_from.get()` as we saw above – is a Variable - the Data from the entry box.

It basically grabs the data from the Entry Box in the program – ready to perform the calculation.

After that line we have:

```
meters_val = feet_val * 0.3048
```

In this line of code we perform the conversion calculation.

meters_val (value in meters) = feet_val (value in feet)   x   0.3048

The answer to this calculation is the answer the User wants, we have called it `meters_val` and we will display it back onto the GUI window by entering the the next line of code:

```
cvt_to.set('%s meters' % meters_val)
```

Everything after these lines of code we have covered previously. It is the layout of the Buttons and Labels showing the features like colour, text and layout position:

```
from_lbl = Label(top, text='Enter FEET (numbers only):', fg='blue', font='12')
from_lbl.pack()

from_entry = Entry(top, textvariable=cvt_from)
from_entry.pack()

to_lbl = Label(top, text='Result:', fg='purple', font='12')
to_lbl.pack()

result_lbl = Label(top, textvariable=cvt_to, bg='yellow', font='freesansBold, 12')
result_lbl.pack()

convert_btn = Button(top, text='Convert to Meters', fg='blue', command=do_convert)
convert_btn.pack()
```

NOTICE: you will see how the function gets used above – to perform the operations in the program. Our `def do_convert():` function doesn't actually get used until we call it in to our program with our Calculate Button (called `convert_btn`).

If you have any further questions on this lesson please don't hesitate to e-mail or me using the email in the Header

Or message me on Facebook: @PythonCommunity

Well done for getting this far.