Example 3 : Tkinter GUI Window / Buttons & Functions

In example 2 I showed how to add a button that allows the user to cleanly exit the GUI program. A command was assigned to the button so it had a purpose and actually did something.

In this example I will demonstrate how to add another button, with a new function that will print a message to the Shell window when clicked by the user.

Here's the code in full:

```
from tkinter import *

top = Tk()
top.title('Tkinter GUI - Example 3')
top.minsize(800, 800)
top.configure(bg='orange')

lbl = Label(top, text='A Basic Window With 2 Buttons')
lbl.pack()

def hello_callback(): print ('Button Clicked')

btn = Button(top, text='Click Here', font='freesansbold, 14', command=hello_callback)
btn.pack()

exit_button = Button(top, text='Exit', fg='red', bg='black', font='freesansbold, 14', command=quit)
exit_button.pack(side=BOTTOM)

top.mainloop()
```

When you type this code out in Python 3 and run it you should see a few main differences.

1. The window size is larger (simple stuff – you can make this any size you want)
2. The background colour is orange (top.configure)
3. We have a label inserted to the window (like a heading but not the same as a Title)
4. Theres another button saying 'Click Me' (Click it and see what happens in the Shell window)

I will run through the code quickly to try and explain what is happening. Although we have covered most of it in the first two examples. We have changed a few visual details, but the most substancial addition is the new Button. This example makes a function called 'hello_callback' and assigns it to our new button to make it print a message.

- `from tkinter import *`

We have imported some extra things from Tkinter but with out current import comand/script we have access to the things we need. If we wanted to, we could just specifically import the things we will be using, but as our GUI's get more complex we will find we have to import numerous things – where we can neatly do it in one line.

If we were to import things individually, it would look something like this:

```
from tkinter import Label, Button, Tk
```

Next we have our main set-up commands for our root (top-level) window :

```
top = Tk()
top.title('Tkinter GUI - Example 3')
top.minsize(800, 800)
top.configure(bg='orange')
```

The first three are nothing new to us as we wrote them in example 2'. The title is slightly different – but you can write anything you want in between the quote marks. The last line: `[top.configure(bg='orange')]` Sets up the configuration of the root/top window (Tk) and we have assigned a background colour – orange.

Now, if you have used Pygame you may know that colors in Python usually have a numerical value (R, G, B) Red Green and Blue are the primary colors and the amount of each (when the three are mixed) determines the end colour. Luckily Tkinter has some pre-built colours ready made so you can type in most traditional colours in the place of 'orange' and change it yourself. You can even add 'light' to some ("lightBlue" for example) but I do not have a complete list of available colours to hand.

Next:

```
lbl = Label(top, text='A Basic Window With 2 Buttons')
lbl.pack()
```

Nothing too fancy here, we are adding a 'Label' which is very similar to a 'Header' in a text document, only you can choose to position a label anywhere. Tkinter will by default pack it at the top-center which is fine for our example. A label can be used to give an instruction, or a piece of information to the user.

Now we move on to our new button. We have written our own function for this button, a very basic one by Python's standards, and although it is simple it is still very effective. Our new function tells Python to print a message to the Shell. When we made the Exit button we used a 'quit' command assigned to the button to exit cleanly.

This time our command will be the new function:

```
def hello_callback(): print ('Button Clicked')

btn = Button(top, text='Click Here', font='freesansbold, 14', command=hello_callback)
btn.pack()
```

The first line tells Python we want to 'define' a function called 'hello_callback'. The definition is to print the message 'Button Clicked'. Python will automatically print things to the Shell unless told otherwise. Later on we will see how to print feedback to a widget within our GUI window instead.

You can change the printed message to anything you want, the main thing is that you understand how to assign functions to buttons / widgets.

The rest of the code is the same as before – `exit_button` and `top.mainloop()`