Example 2 : Tkinter GUI Window / Adding A Button

This is a little more than just adding a button. The button has a FUNCTION assigned to it – a "quit" command to allow the user to exit cleanly. When making a GUI we would normally aim at making the program 'User-Friendly' and easy to use.

A button is known in Tkinter as a 'Widget'

Adding a button may seem a very simple feature but in this example we have learnt a few useful things:

- Add colour to the writing
- Add colour to the button background
- Assign a font and font size to the text within the button
- Give the button a function (to quit)
- Position the button

OK so here is the code, it is the same as lesson one with 2 extra lines: (REMEMBER if you are running any version of Python 2 you need to change `tkinter` to `Tkinter` on the import command)

```
from tkinter import *

top = Tk()
top.minsize(400,400)
top.title("Tkinter Basic GUI's")

exit_button = Button(top, text='Exit', fg='red', bg='black', font='freesansbold, 12', command=quit)
exit_button.pack(side=BOTTOM)

top.mainloop()
```

So, the majority of the above code is the same. There are two additional lines here and I will briefly explain them. Do not be put off by the simplicity of adding a button, there are a few usefull things involved that will help you later on, especially when adding other features like text-entry boxes and menus.

So in this example I added 2 lines of code – to put a button in the window. Here is the first extra line of code:

```
exit_button = Button(text='Exit', fg='red', bg='black', font='freesansbold, 12', command=quit)
```

So the first part of this line of code " `exit_button = Button` " is simply saying to Python that we want to add a Button, and we wish to call it "exit_button". You can call your features anything within reason – meaning that if you stick to lettered words (strings) you should be OK. As a general rule don't use numbers, and if you choose to assign a name which Python already knows as a function you will get errors too.
For Example: We are making a Quit / Exit button. So, if you call the button "quit" like this it won't work:

```
quit = Button()
```

You will find that Python won't accept it because it already knows the name “quit” as somethingelse – a command to quit. You should usualy be able to spot this because if you just type quit in a Python edit window, you will notice the colour difference when Python recognises it. Just try to keep it simple.

The rest of this additional line of code is all contained within () brackets. Here is what is going on:

```
(top, text='Exit', fg='red', bg='black', font='freesansbold, 12', command=quit)
```

- `top` : this tells Python to put the button in the top window (Root / Parent window) In general this is where they will go unless told otherwise (If you have sub-windows for example).
- `text='Exit'` : This assigns Text to the button – in our example we have made an exit button so the text should read 'Exit'.
- `fg='red'` : 'fg' is an abbreviation for 'Foreground'. In this case it represents the colour of the text.
- `bg='black'`: 'bg' is an abbreviation of 'Background'. In this case it represents the colour of the button.
- `font='freesansbold, 12'` : Fairly self-explanatory, we are saying to Python that we want the text to be a specific font, and size. (***SEE NOTE ON FONTS BELOW***)
- `command=quit` : This tells Python that we want the button to do something specific – by giving it a command. We are making an Exit button so luckily Python has a function readu-made for this. The quit-program command will now be assigned to this button.

The second additional line is : `exit_button.pack(side=BOTTOM)`
Withouth this line, Python will not add the button it to the GUI. There are a few ways to do this.

pack() is known as a “Layout Manager” or “Geometry Manager”

Pack is the simplest of the layout managers, and another popular one is **'grid'** and the third is **'place'**. You can use grid to be more specific with where you want to position things within your GUI. I am still getting to grips with grid() myself so for now I will just be using pack(). As we are keeping the GUI basic we don't need to use grid. If for example you wanted to make a calculator GUI with the numbers layed out nicely like buttons on a calculator you would NEED to use the grid layout manager.

With the pack() feature you can include some basic instructions for the position of your widget / button. You can specify to place the widget in any of the following positions:

TOP / BOTTOM / LEFT / RIGHT

S in our code we have positioned the Button at the bottom of the GUI with:

`exit_button.pack(side=BOTTOM)` – 'bottom' can be replaced with any one of the positions noted above. If you don't add a position and just use the pack() feature alone Python will normally place your widget towards the top-center.

Here is a list of available Widgets in Tkinter:

The Button Widget
The Canvas Widget
The Checkbutton Widget
The Entry Widget
The Frame Widget
The Label Widget
The LabelFrame Widget
The Listbox Widget
The Menu Widget
The Menubutton Widget
The Message Widget
The OptionMenu Widget
The PanedWindow Widget
The Radiobutton Widget
The Scale Widget
The Scrollbar Widget
The Spinbox Widget
The Text Widget
The Toplevel Widget

The list of Widgets was found on : http://effbot.org/tkinterbook/tkinter-index.htm – There are a number of guides and sources of documentation online, for Tkinter general use I found this site to be quite useful

NOTE ON FONTS FROM ABOVE

You can research the exact list of available fonts online, and I would imagine you can install a wider range if needed, BUT, I have a simple method I use, to see the general list of available fonts from within Python without having to look online or anything:

While you're in Python, move you're mouse pointer to the top-bar (with file, edit, format, run, options, windows and help menus) and click on the menu choice 'options' then click on the sub-menu 'configure IDLE' and you will see a list of available fonts. DON'T USE THIS TO SELECT THE FONT because this is actually for changing the font in which you type Python code.

You can use this list to see the fonts that Python has readily available and choose the name of one to add to your GUI widget.