A Calculator With GUI Using Python

→ Graphical User Interface (GUI) is a form of user interface which allows users to interact with computers through visual indicators using items such as icons, menus, windows, etc.

→ Tkinter is a built-in python module that is used to create GUI applications. It gives an-object oriented interface to the Tk GUI toolkit

Some other python libraries available for creating GUI are
- Kivy
- Python Qt
- wxPython

→ Widgets

- Widgets in Tkinter are the elements of GUI application which provides various controls such as Labels, Buttons, ComboBoxes, Checkboxes, Menubars, etc. these are to enable users to interact with the app.

- Tkinter uses the win32 gui api's (for windows) to render these elements.

(•) methods :

→ pack - places widgets blockwise, also provides attributes to help with the structuring.

→ grid - Creates grid design to help in designing the gui. (similar to CSS grid).

→ place - places widgets to a specified position.

Structure of a Tkinter program

Importing Tkinter modules

Creating a main window for GUI

Adding widgets to the app

Enter the Main event loop

(•) What is the main event loop?

→ Root.mainloop() is a method in the main window that executes what we wish to execute in an application (the Gui application starts here). As the name implies it would loop forever until the user exits the window.

(•) label - a widget the the user is not supposed to interact with and labels should always be packed.

(•) .geometry - is a function that defines width and height.
- minsize()
- maxsize()

(•) .title - is a method used to give ~~tittle~~ title to the application we are working with.

* Important label Options :
  - text - adds text to display
  - bd - background
  - fg - foreground
  - font - used to set font
  - padx - x padding (x-axis)
  - pady - y padding (y-axis)
  - relief - border styling (SUNKEN, RAISED, GROOVE, RIDGE)

* Attributes of Pack : The Pack geometry manager packs widgets in rows or columns. We can use options like fill, expand and side to control this geometry manager.

(1) fill : Determines weather widget fills any extra space allocated to it by packer or keeps its own min dimentions.

(2) side : Determine the widgets position in relation to its parent. (TOP, BOTTOM, LEFT, RIGHT)

(3) anchor : Anchors are used to define where text is positioned relative to a reference point. (n, s, e, w, center, nw, sw, be and se)

(•) Frames - frames are like divs (in css) can be understood as containers.

(•) Button - similar to the label we have button class that is used to create buttons.

(•) Grid - .grid() is ~~same~~ an alternative packing where we can specify the element's position in the grid using rows and columns. eg - btn.grid(row=1, column=1)

(•) Entry Widget - It's similar to input area in js, or it is a text area. (class = .Entry(args))

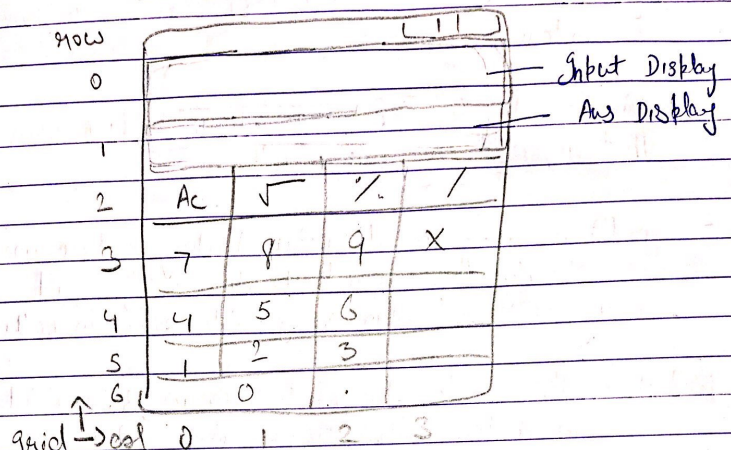
* Variable types in Tkinter:
  (i) BooleanVar
  (ii) DoubleVar
  (iii) IntVar
  (iv) StringVar

  { these are the variable classes that can be used in textvariable attr of Entry obj
  eg - user = Entry(root, textvariable = StringVar())

  * Holds the value inputted to the Entry obj

(•) Checkbox - to create a checkbox in ~~the~~ gui

Calculator Strategy



- Input Display = Entry Widget => text variable = Ans_Val
- Answer Display = Entry Widget => text variable = answerfinalLabel

- The buttons having similar pattern are created using create Button function
- Rest of the buttons are hard coded

- other functions :-
  - -> changeAns_val - Updates ans value to the Entry
  - -> ~~Answer~~ cleanAns_Val - ~~clear~~ clears ans val in Entry
  - -> evaluateSquareRoot - evaluates sq root using Math module
  - -> evaluateAnswer - evaluates answer.
  - -> allClear - Works on 'AC' button