GLS UNIVERSITY

Faculty of Computer Applications & Information Technology

Integrated MCA Programme

Semester V

Unit 5: Python GUI Programming (Tkinter)

Introduction

- The tkinter package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit.
- Both Tk and tkinter are available on most Unix platforms, including macOS, as well as on Windows systems.
- Tkinter supports a range of Tcl/Tk versions, built either with or without thread support. The official Python binary release bundles Tcl/Tk 8.6 threaded.
- Tcl/Tk is not a single library but rather consists of a few distinct modules, each with separate functionality and its own official documentation.
- Python’s binary releases also ship an add-on module together with it.

Introduction

- Python provides various options for developing graphical user interfaces (GUIs).
- Most important are listed below.
  - Tkinter: It is the easiest among all to get started with. It is Python's standard GUI (Graphical User Interface) package. It is the most commonly used toolkit for GUI Programming in Python.
  - JPython: It is the Python platform for Java that is providing Python scripts seamless access o Java class Libraries for the local machine.
  - wxPython: It is open-source cross-platform GUI toolkit written in C++. It one of the alternatives to Tkinter, which is bundled with Python.
- There are many other interfaces available for GUI. But these are the most commonly used ones.

Tcl

- Tcl is a dynamic interpreted programming language, just like Python.
- Though it can be used on its own as a general-purpose programming language, it is most commonly embedded into C applications as a scripting engine or an interface to the Tk toolkit.
- The Tcl library has a C interface to create and manage one or more instances of a Tcl interpreter, run Tcl commands and scripts in those instances, and add custom commands implemented in either Tcl or C.
- Each interpreter has an event queue, and there are facilities to send events to it and process them.
- Unlike Python, Tcl's execution model is designed around cooperative multitasking, and Tkinter bridges this difference (see Threading model for details).

Python GUI Programming - Tkinter

- It is the standard GUI toolkit for Python.
- It was written by Fredrik Lundh.
- Python when combined with Tkinter provides a fast and easy way to create GUI applications.
- Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.
- Creating a GUI application using Tkinter is an easy task.
- Tkinter is an acronym for "Tk interface".
- Tkinter comes pre-installed with Python3

Tkinter Modules

- Most applications will need the main tkinter module, as well as the tkinter.ttk module, which provides the modern themed widget set and API.

  - from tkinter import *

  - from tkinter import ttk

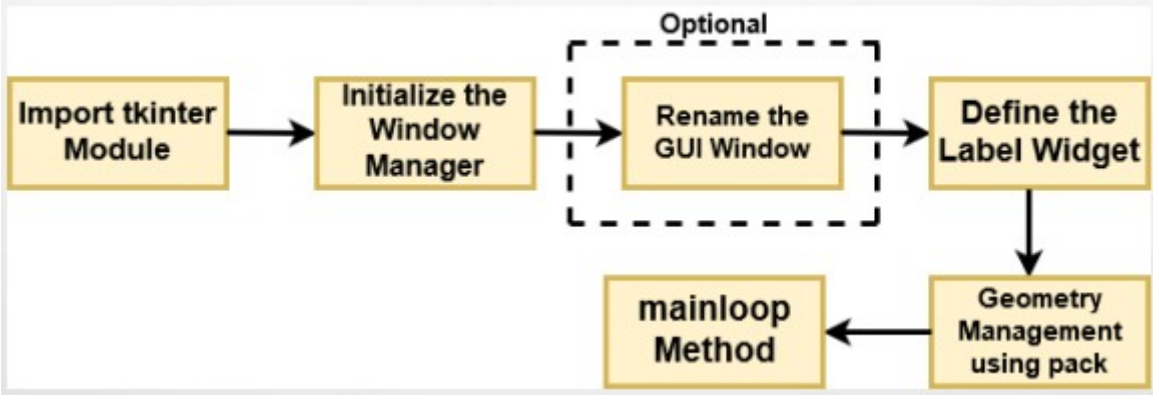
- Python offers multiple options for developing GUI (Graphical User Interface).
- Out of all the GUI methods, tkinter is the most commonly used method.
- It is a standard Python interface to the Tk GUI toolkit shipped with Python.
- Python Tkinter is the fastest and easiest way to create GUI applications.

Tkinter

- To create a Tkinter Python app, you follow these basic steps:
  - **Import the tkinter module:** This is done just like importing any other module in Python. Note that in Python 2.x, the module is named 'Tkinter', while in Python 3.x, it is named 'tkinter'.
  - **Create the main window (container):** The main window serves as the container for all the GUI elements you'll add later.
  - **Add widgets to the main window:** You can add any number of widgets like buttons, labels, entry fields, etc., to the main window to design the interface as desired.
  - **Apply event triggers to the widgets:** You can attach event triggers to the widgets to define how they respond to user interactions.

Tk

- Tk is a Tcl package implemented in C that adds custom commands to create and manipulate GUI widgets.
- Each Tk object embeds its own Tcl interpreter instance with Tk loaded into it.
- Tk's widgets are very customizable, though at the cost of a dated appearance.
- Tk uses Tcl's event queue to generate and process GUI events.



Tk()

- To create a main window, tkinter offers a method AS:

  ‘Tk(screenName=None, baseName=None, className=’Tk’, useTk=1)’.

- To change the name of the window, you can change the className to the desired one.

mainloop():

- It is used when application is ready to run.
- mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event till the window is not closed.
- Eg: m.mainloop()

pack() method

- tkinter also offers access to the geometric configuration of the widgets which can organize the widgets in the parent windows.
- There are mainly three geometry manager classes class.
  - pack() method: It organizes the widgets relatively in blocks before placing in the parent widget.
  - grid() method: It organizes the widgets in grid (table-like structure) before placing in the parent widget.
  - place() method: It organizes the widgets by placing them on specific positions directed by the programmer.

pack( )

- the pack() method is one of the geometry managers used to organize widgets within a parent widget (like a window or a frame).
- It arranges widgets by packing them in blocks before placing them in the parent widget, either from the top down, bottom up, left to right, or right to left.
- **widget.pack(option=value, ...)**
- **side**: Specifies which side of the parent widget to pack the widget against. Possible values are TOP, BOTTOM, LEFT, RIGHT.
- **fill**: Determines if the widget should expand to fill any extra space. Can take values:
  - NONE (default): The widget keeps its natural size.
  - X: The widget expands horizontally to fill the space.
  - Y: The widget expands vertically to fill the space.
  - BOTH: The widget expands in both directions.
- **expand**: If True, the widget expands to fill any extra space in the geometry manager's parent widget.
- **padx, pady**: Specifies padding along the X and Y axes, respectively, adding space around the widget.
- **anchor**: Specifies where the widget should be packed relative to the available space (e.g., N, S, E, W, CENTER).

place( )

- place() method is another geometry manager used to position widgets within a parent widget. Unlike pack() and grid(), the place() method allows you to specify the exact position of a widget using absolute or relative coordinates.
- **widget.place(option=value, ...)**
- x: The x-coordinate (horizontal position) of the widget, in pixels.
- y: The y-coordinate (vertical position) of the widget, in pixels.
- relx: Specifies the relative x-coordinate as a float (0.0 to 1.0) with respect to the width of the parent widget.
- rely: Specifies the relative y-coordinate as a float (0.0 to 1.0) with respect to the height of the parent widget.
- anchor: Specifies which part of the widget is placed at the x, y coordinate. Options are:
- N, S, E, W, NE, NW, SE, SW, CENTER.
- width: Specifies the width of the widget in pixels.
- height: Specifies the height of the widget in pixels.
- relwidth: Specifies the relative width of the widget as a float (0.0 to 1.0).
- relheight: Specifies the relative height of the widget as a float (0.0 to 1.0).

grid( )

- the grid() method is a geometry manager that organizes widgets in a table-like structure using a grid of rows and columns. Unlike pack() and place(), the grid() method allows widgets to be arranged in a more structured way, assigning them specific row and column locations.
- **widget.grid(option=value, ...)**
- row: Specifies the row in which the widget should be placed (default is 0).
- column: Specifies the column in which the widget should be placed (default is 0).
- rowspan: Specifies how many rows the widget should span (default is 1).
- columnspan: Specifies how many columns the widget should span (default is 1).
- sticky: Determines where the widget should "stick" within the cell. It can take values like N, S, E, W, or combinations like NE, SW to specify alignment (e.g., top-left, bottom-right).
- padx: Specifies horizontal padding (space) inside the cell.
- pady: Specifies vertical padding (space) inside the cell.
- ipadx: Specifies the internal horizontal padding (within the widget).
- ipady: Specifies the internal vertical padding (within the widget).

Grid Manager

- The Grid geometry manager places the widgets in a 2-dimensional table, which consists of a number of rows and columns.
- The position of a widget is defined by a row and a column number.
- Widgets with the same column number and different row numbers will be above or below each other.
- Correspondingly, widgets with the same row number but different column numbers will be on the same "line" and will be beside of each other, i.e. to the left or the right.
- Using the grid manager means that you create a widget, and use the grid method to tell the manager in which row and column to place them.
- The size of the grid doesn't have to be defined, because the manager automatically determines the best dimensions for the widgets used

Tkinter Variable Classes

- Variables can be used with most entry widgets to track changes to the entered value.
- The Checkbutton and Radiobutton widgets require variables to work properly.
- Some widgets (like text entry widgets, radio buttons and so on) can be connected directly to application variables by using special options:
- variable, textvariable, onvalue, offvalue, and value.
- Create a Tkinter variable:
- `x = StringVar() # Holds a string; default value ""`
- `x = IntVar() # Holds an integer; default value 0`
- `x = DoubleVar() # Holds a float; default value 0.0`
- `x = BooleanVar() # Holds a boolean, returns 0 for False and 1 for True`

Tkinter Widgets

- Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application.
- These controls are commonly called widgets.
- Common widgets of Tkinter are:

• Label	• menu
• Button	• list – box
• Canvas	• menu button
• combo-box	• Message
• frame	• tk_option Menu
• level	• progress-bar
• check-button	• radio button
• entry	• scroll bar
• level-frame	• Separator
	• tree-view and many more