**Word Count: 1031**

Video, Create a GUI with Python

In this video, we will learn about Tkinter library in Python and learn to use Tkinter widgets to create Graphical User Interface applications. Finally, we will learn to use the geometry managers available for Tkinter.

So what is a GUI?

Graphical User Interface is nothing but an application that helps you to interact with the computers visually. They are used to perform different tasks in desktops, laptops, and other electronic devices.

Python has a plethora of libraries, and these 4 stands out mainly when it comes to GUI. There are as follows:

* Kivy
* Python QT
* wxPython
* Tkinter

Among all of these, Tkinter is preferred by a lot of learners and developers just because of how simple and easy it is. I am sure you will have the same opinion in a while as well.

Tkinter is actually an inbuilt Python module used to create simple GUI apps. It is the most commonly used module for GUI apps in Python. You don’t need to worry about the installation of the Tkinter module as it comes with Python default. Consider the following diagram it shows how an application actually executes in Tkinter:

To start out with, we first import the Tkinter model. Followed by that, we create the main window. It is in this window that we are performing operations and displaying visuals and basically everything. Later, we add the widgets, and lastly, we enter the main event loop.

If you noticed, there are 2 keywords here that you might not know at this point. They are

* Widgets
* Main Event Loop

An event loop is basically telling the code to keep displaying the window until we manually close it. It runs in an infinite loop in the back-end. Check out the following hello world GUI code for better clarity:

As you can see, we are importing the Tkinter package and defining a window. Followed by that, we are giving a window title, which is shown on the title tab whenever you open an application.

For example, Microsoft Word is shown on the title tab when you open a word application, correct? Similarly, here we call it GUI. We can call it anything we want based on the requirement.

Lastly, we have a label. A label is nothing but what output needs to be shown on the window. In this case, as you can already see, it is hello world.

Now let's run this code in the IDE. You can now see your first GUI app.

The next question we need to ask is, what are widgets?

Widgets are something like elements in HTML. You will find different types of widgets to the different types of elements in Tkinter.

Let’s see a brief introduction to all of these widgets in Tkinter.

Check out this diagram for the list of the majorly used Tkinter widgets:

* Canvas – Canvas is used to draw shapes in your GUI.
* Button – Button widget is used to place the buttons in Tkinter.
* Checkbutton – Check button is used to create the check buttons in your application.
* Entry – Entry widget is used to create input fields in the GUI.
* Frame – Frame is used as containers in Tkinter.
* Label – Label is used to create single line widgets like text, images etc.
* Menu – Menu is used to create menus in the GUI.

Let us walk through all of these widgets individually for better understanding. As we don’t have time to go through an example program for each of the widgets, I will be showing a code snippet and output of that side by side. The syntax for each of the widgets are self-explanatory. Please feel free to pause the videos on each widget to have a good look. I have also provided the sample syntax in the resources section.

1. Label Widget
2. Button Widget
3. Check button widget
4. Entry Widget
5. Combobox widget
6. Radio button widget
7. Scrolled Text Widget
8. Message Box Widget
9. SpinBox Widget

Just creating a widget does not necessarily mean that it will appear on the screen.. To get the widget to appear, we need to tell the parent widget where to put it. To do that, we use one of Tkinter's three geometry managers. A geometry manager is some code that runs on the backend of Tkinter. We simply choose which geometry manager we want to use and give it some parameters to work with. The three geometry managers are: grid, pack, and place.

The grid manager places widgets in a table format with rows and columns. It will avoid overlapping widgets will resize rows/columns as necessary to fit the widgets.

The pack is often the easiest geometry manager to use, as it just puts widgets in a single row or column . It "packs" the widgets by putting them side-by-side or top-to-bottom.

The place geometry manager offers the most control but can be the most difficult to use. It allows you to specify the absolute or relative positions of the widgets in a window.

As we have covered the basics of GUI development, we ask you to go through the resources to learn more and practice with different examples. GUI creation is a vast topic, and it can’t be covered in a single video

Now I will give you an activity. Upgrade your command line simple calculator you built earlier with GUI Capabilities. The final GUI Simple Calculator application should look like this. If you are not able to successfully make this calculator, you can refer to the code I have provided in the resources section.

Summary

In this video, we have covered the following

* What is Tkinter
* How to work with Tkinter methods
* Develop Simple GUIs
* Create a Simple GUI Calculator

Section Summary.

In this section, we have covered the following

* Basics of Python Programming
* Typecasting & loops in Python
* Conditional Branching & Functions in Python
* Modules in Python
* Create a GUI with Python

In the next section, we will start working with Hardware interfacing in the Raspberry Pi 4