

TKinter - GUI (Day 39) 01/07/2025

✓ TKinter is a standard Python interface to the
TK GUI toolkit.

✓ TKinter is a Python library that allows you to
create window-based applications.

✓ It provides a set of tools to build interface
such as buttons, labels, text

Why use

- ✓ Ease of use
- ✓ Cross-platform
- ✓ Lightweight
- ✓ Standard library

New folder → TKinter → file name → Day39.py
import tkinter run in terminal,

Creating a basic Tkinter window.

import tkinter as TK

root = tk.TK()

root.mainloop()

root.title("simple GUI")

root.geometry("300x200")

→

widgets!

widgets are the building blocks of a Tkinter application.

Some common widgets are:

Label: Displays text or images

Button: A clickable button that can trigger actions

Entry: A single-line text box for user input

Text: A multi-line text box for input and display.

```
label = tk.Label(root, text="Hello Tkinter", font=("Arial", 24))
```

```
label.pack()
```

```
root.mainloop()
```

```
tk.Label
```

```
entry = tk.Entry(root)
```

```
entry.pack()
```

```
Button = tk.Button(root, text="Click me", command=welcome)
```

```
entry.pack()
```

```
def welcome():
```

```
    print("welcome to Tkinter GUI") (terminal output)
```

```
label.config(text="welcome to GUI") (window output)
```

```
user_input = entry.get()
```

```
label.config(text=f"welcome to {user_input}")
```


Basic layout in Tkinter

- ✓ pack()
- ✓ grid()
- ✓ place()

1) pack()

- ✓ The pack method arranges widgets in blocks and places them in the available space.
- ✓ widgets are packed in the order they are created (top to bottom, left to right)

2) grid()

The grid() method places widgets in a grid system, similar to a table.

- ✓ you can specify the row and column for each widget.

```
label = tk.Label(root, text="Name:")
```

```
label.grid(row=0, column=0)
```

```
entry = tk.Entry(root)
```

```
entry.grid(row=0, column=1)
```

3) place()

The place() method allows for more precise control over the placement of widgets by specifying coordinates.

```
label.place(x=50, y=50)
```


Working with frames.

- ✓ Frames are useful for organizing and grouping widgets in a window.
- ✓ A frame is essentially a container for widgets, allowing for better organization and layout control

```
root = tk.Tk()
```

```
frame = tk.Frame(root)
```

```
frame.pack()
```

Add widgets inside the frame.

```
label = tk.Label(frame, text = "This is inside frame")
```

```
label.pack()
```

```
root.mainloop()
```