Python Programming

Unit 04 – Lecture 01: Tkinter Introduction and Widgets

Tofik Ali

School of Computer Science, UPES Dehradun

February 14, 2026

Repository: `https://github.com/tali7c/Python-Programming`

Quick Links

Core Concepts Demo Interactive Summary

Agenda

Learning Outcomes

- Create a basic Tkinter window and run the GUI loop

Learning Outcomes

- Create a basic Tkinter window and run the GUI loop
- Understand widgets and how they are placed in a window

Learning Outcomes

- Create a basic Tkinter window and run the GUI loop
- Understand widgets and how they are placed in a window
- Use common widgets: Label, Button, Entry, Text, Canvas, Listbox

Learning Outcomes

- Create a basic Tkinter window and run the GUI loop
- Understand widgets and how they are placed in a window
- Use common widgets: Label, Button, Entry, Text, Canvas, Listbox
- Use selection widgets: Checkbutton, Radiobutton, Combobox, Spinbox

What is Tkinter?

- Tkinter is Python's standard GUI library

What is Tkinter?

- Tkinter is Python's standard GUI library
- Allows building desktop apps (forms, tools, mini-projects)

What is Tkinter?

- Tkinter is Python's standard GUI library
- Allows building desktop apps (forms, tools, mini-projects)
- Works with widgets (buttons, text boxes, etc.)

Smallest Tkinter App

```
import tkinter as tk

root = tk.Tk()
root.title("My App")
root.geometry("400x200")
root.mainloop()
```

- `Tk()` creates the main window

Smallest Tkinter App

```
import tkinter as tk

root = tk.Tk()
root.title("My App")
root.geometry("400x200")
root.mainloop()
```

- `Tk()` creates the main window
- `mainloop()` keeps the window running

Widgets (GUI Building Blocks)

- Widgets are UI components: Label, Button, Entry, etc.

Widgets (GUI Building Blocks)

- Widgets are UI components: Label, Button, Entry, etc.
- Most widgets need:

Widgets (GUI Building Blocks)

- Widgets are UI components: Label, Button, Entry, etc.
- Most widgets need:
  - a parent (usually `root` or a Frame)

Widgets (GUI Building Blocks)

- Widgets are UI components: Label, Button, Entry, etc.
- Most widgets need:
  - a parent (usually `root` or a Frame)
  - configuration options (text, width, command, ...)

Label, Button, Entry (Example)

```
import tkinter as tk

root = tk.Tk()
name_var = tk.StringVar()

tk.Label(root, text="Name:").pack()
tk.Entry(root, textvariable=name_var).pack()

def greet():
    print("Hello,", name_var.get())

tk.Button(root, text="Greet", command=greet).pack()
root.mainloop()
```

More Widgets

- `Text`: multi-line text input/output

More Widgets

- `Text`: multi-line text input/output
- `Canvas`: drawing shapes

More Widgets

- `Text`: multi-line text input/output
- `Canvas`: drawing shapes
- `Scrollbar`: scrolling support (Text/Listbox)

More Widgets

- `Text`: multi-line text input/output
- `Canvas`: drawing shapes
- `Scrollbar`: scrolling support (Text/Listbox)
- `Listbox`: list selection

Selection Widgets

- `Checkbutton`: multiple independent selections

Selection Widgets

- `Checkbutton`: multiple independent selections
- `Radiobutton`: choose exactly one option

Selection Widgets

- `Checkbutton`: multiple independent selections
- `Radiobutton`: choose exactly one option
- ttk `Combobox`: dropdown list

Selection Widgets

- `Checkbutton`: multiple independent selections
- `Radiobutton`: choose exactly one option
- ttk `Combobox`: dropdown list
- `Spinbox`: numeric selection

Demo: Widget Showcase

- File: `demo/widgets_showcase.py`

Demo: Widget Showcase

- File: `demo/widgets_showcase.py`
- Shows:

Demo: Widget Showcase

- File: `demo/widgets_showcase.py`
- Shows:
  - Label, Button, Entry

Demo: Widget Showcase

- File: `demo/widgets_showcase.py`
- Shows:
  - Label, Button, Entry
  - Checkbutton and Radiobutton

Demo: Widget Showcase

- File: `demo/widgets_showcase.py`
- Shows:
  - Label, Button, Entry
  - Checkbutton and Radiobutton
  - Listbox and Text

Demo: Widget Showcase

- File: `demo/widgets_showcase.py`
- Shows:
  - Label, Button, Entry
  - Checkbutton and Radiobutton
  - Listbox and Text
  - ttk Combobox

Checkpoint 1

Question: Why do we need `root.mainloop()` in Tkinter?

Checkpoint 2

Question: What is the benefit of using `StringVar` with an `Entry` widget?

Think-Pair-Share

Discuss:

- What makes a GUI *easy to use*?
- Give 3 design rules (labels, spacing, validation, feedback, etc.).

Key Takeaways

- Tkinter builds GUI apps using widgets

Key Takeaways

- Tkinter builds GUI apps using widgets
- `Tk()` creates a window; `mainloop()` runs the event loop

Key Takeaways

- Tkinter builds GUI apps using widgets
- `Tk()` creates a window; `mainloop()` runs the event loop
- Widgets like Label/Button/Entry handle input and actions

Key Takeaways

- Tkinter builds GUI apps using widgets
- `Tk()` creates a window; `mainloop()` runs the event loop
- Widgets like Label/Button/Entry handle input and actions
- Selection widgets (check/radio/combobox) capture choices

Exit Question

Write the 3 key lines needed to create a Tkinter window titled `"Hello GUI"` and keep it running.