Tkinter

1. Introduction to Tkinter

Tkinter is Python’s standard GUI (Graphical User Interface) library, which provides tools to create simple to complex desktop applications. It is easy to use, and because it’s part of the Python standard library, it doesn’t require any additional installation.

---

2. Setting Up Tkinter

- **Importing Tkinter:**

```
import tkinter as tk                  # Standard convention
```

- **Creating the Main Window:**

```
root = tk.Tk()                        # Initializes the main application window
root.title(”My Tkinter App”)          # Sets the window title
root.geometry(”600x400”)              # Sets the window size (width x height)
```

- **Running the Application:**

```
root.mainloop()              # Starts the Tkinter event loop, keeps the window open
```

---

3. Basic Widgets

1. **Label:** Displays text or an image.

```
label = tk.Label(root, text=”Hello, Tkinter!”, font=(”Arial”, 14))
```

```
label.pack(pady=10)  # Adds the label to the window
```

2. Button: Triggers a function when clicked.

```
def on_click():
    print(”Button Clicked!”)
button = tk.Button(root, text=”Click Me”, command=on_click)
button.pack(pady=10)
```

3. Entry: A single-line text entry widget.

```
entry = tk.Entry(root)
entry.pack(pady=10)

def get_entry_value():
    print(”Entry Value:”, entry.get())
```

4. Text: A multi-line text entry widget.

```
text = tk.Text(root, height=5, width=30)
text.pack(pady=10)
```

5. Frame: A container for organizing other widgets.

```
frame = tk.Frame(root)
frame.pack(pady=10)

label_in_frame = tk.Label(frame, text=”Inside Frame”)
label_in_frame.pack()
```

6. Canvas: For drawing shapes, lines, or placing images.

```
canvas = tk.Canvas(root, width=300, height=200, bg=”white”)
canvas.pack(pady=10)
```

```
# Drawing a line
canvas.create_line(0, 0, 300, 200, fill=”blue”, width=2)
```

7. **Menu:** Creating a dropdown menu.

```
menu = tk.Menu(root)
root.config(menu=menu)

file_menu = tk.Menu(menu, tearoff=0)
menu.add_cascade(label=”File”, menu=file_menu)
file_menu.add_command(label=”New”)
file_menu.add_command(label=”Open”)
file_menu.add_separator()
file_menu.add_command(label=”Exit”, command=root.quit)
```

4. Layout Management

1. **pack():** Arranges widgets in blocks before placing them in the parent widget.

```
label.pack(side=”top”, fill=”x”, pady=5)
```

2. **grid():** Places widgets in a grid of rows and columns.

```
label.grid(row=0, column=0, padx=10, pady=10)
entry.grid(row=0, column=1, padx=10, pady=10)
```

3. **place():** Places widgets at an absolute position within the window.

```
label.place(x=50, y=100)
```

5. Event Handling

Tkinter allows you to bind events (like mouse clicks, key presses) to widgets.

```
def on_button_click(event):
    print(f”Clicked at {event.x}, {event.y}”)
```

```
button.bind(”<Button-1>”, on_button_click)  # Binds left-click event to button
```

Common Events:

- <Button-1>: Left mouse button click
- <KeyPress>: Key press event
- <Motion>: Mouse movement over the widget

---

6. Advanced Widgets

1. **Listbox:** Displays a list of options.

```
listbox = tk.Listbox(root)

listbox.pack()



listbox.insert(1, ”Option 1”)

listbox.insert(2, ”Option 2”)

listbox.insert(3, ”Option 3”)
```

---

2. **Checkbox (Checkbutton):** Allows the user to make multiple selections.

```
var1 = tk.IntVar()

checkbox = tk.Checkbutton(root, text=”Check me!”, variable=var1)

checkbox.pack(pady=10)
```

---

3. **Radiobutton:** Allows the user to select one option from a set.

```
var2 = tk.StringVar(value=”Option 1”)

radio1 = tk.Radiobutton(root, text=”Option 1”, variable=var2, value=”Option 1”)

radio2 = tk.Radiobutton(root, text=”Option 2”, variable=var2, value=”Option 2”)



radio1.pack()

radio2.pack()
```

---

4. **Scale:** A slider to select a numeric value.

```
scale = tk.Scale(root, from_=0, to=100, orient=”horizontal”)
```

```
scale.pack(pady=10)
```

---

5. **Combobox (Using ttk module):** A dropdown menu for selecting an option.

```
from tkinter import ttk


combobox = ttk.Combobox(root, values=[”Option 1”, ”Option 2”, ”Option 3”])
combobox.pack(pady=10)
combobox.current(0)  # Set the default selection
```

---

7. Messagebox

A simple dialog box to display messages.

```
from tkinter import messagebox


def show_message():
    messagebox.showinfo(”Information”, ”This is an info message.”)


button = tk.Button(root, text=”Show Message”, command=show_message)
button.pack(pady=10)
```

---

8. File Dialogs

To open or save files using a standard dialog box.

```
from tkinter import filedialog


def open_file():
    file_path = filedialog.askopenfilename()
    print(f”File selected: {file_path}”)


button = tk.Button(root, text=”Open File”, command=open_file)
```

```
button.pack(pady=10)
```

9. Organizing Widgets with PanedWindow

A container widget to stack child widgets either vertically or horizontally.

```
pane = tk.PanedWindow(root)

pane.pack(fill="both", expand=1)


left = tk.Label(pane, text="Left Pane")

pane.add(left)


right = tk.Label(pane, text="Right Pane")

pane.add(right)
```

10. Status Bar

A simple label typically displayed at the bottom of the window to show status messages.

```
status = tk.Label(root, text="Status: Ready", bd=1, relief="sunken", anchor="w")

status.pack(side="bottom", fill="x")
```

prepared by chhavi-rohilla
https://www.linkedin.com/in/chhavi-rohilla-607996251