

Dhirubhai Ambani University

(Formerly known as DA-IICT)

Topic: GUI in Python-Tkinter

Course: Programming Lab

Course Code- PC503

Dr. Ankit Vijayvargiya

Assistant Professor

Room No. 4205, Faculty Block 4

Email: ankit_Vijayvargiya[at]dau.ac.in

Phone: 079-68261628(O), 7877709590(M)

GUI

What is GUI?

- A Graphical user interface (GUI) is an application that features buttons, windows, and various other widgets that the user can use to interact with the application.
- A good example would be a web browser. It features buttons, tabs, and a main window where all the content is displayed.

Why is a GUI Needed?

- **User-Friendly:** Intuitive for non-technical users.
- **Efficiency:** Faster interactions via point-and-click.
- **Visual Feedback:** Immediate response to actions.
- **Multitasking:** Supports multiple tasks via windows/menus.
- **Accessibility:** Easier for diverse users, including those with disabilities.

GUI

Python Libraries for GUI Development:

- Tkinter: Built-in, lightweight, cross-platform, beginner-friendly.
- PyQt: Qt-based, ideal for complex, professional apps.
- Kivy: Open-source, multi-touch apps for mobile/desktop.
- wxPython: Native look-and-feel, cross-platform.
- PyGTK: GTK-based, suited for GNOME environments.
- PySide: Qt-based, free for commercial use.
- BeeWare: Native apps for multiple platforms.

Choosing a library depends on the project's complexity, platform, and the features required.

GUI

Why Tkinter

Tkinter: Python's standard GUI library, included with Python.

Key Benefits:

- Built-in, no extra installation.
- Simple and intuitive for beginners.
- Cross-platform (Windows, macOS, Linux).
- Event-driven programming for responsive apps.
- Wide range of widgets for calculators, forms, etc.

Now, let us move further with basic tkinter setup and creation of labels, widgets, etc.

GUI

First Tkinter Application

Steps to Create a Tkinter App:

- Import the tkinter module.
- Create the main application window with Tk().
- After creating the main window, add the widgets (e.g., buttons, labels, frames).
- Call the main event loop with mainloop(), so that the action can take place on the user's computer screen.

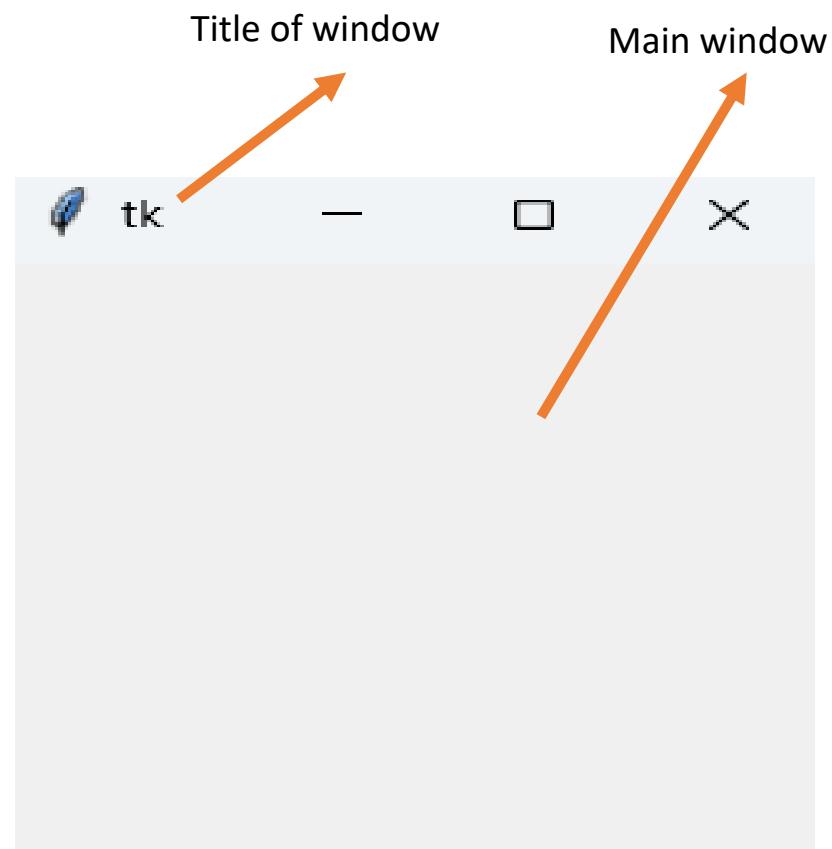
Code Exam

```
import tkinter as tk # Logical operator

window = tk.Tk()

#Add widgets button over here

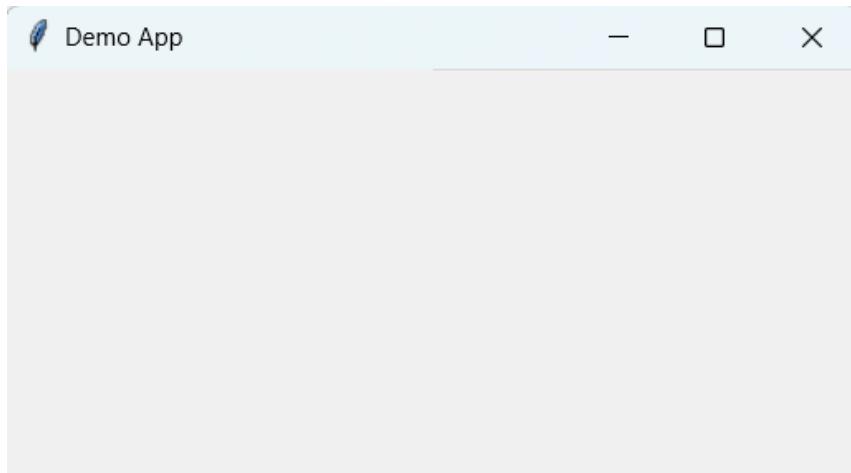
window.mainloop()
```



GUI

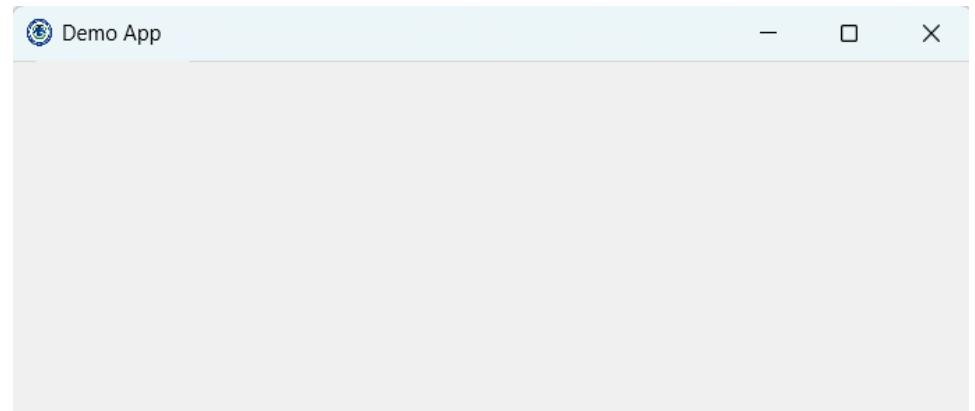
Add Title

```
window = tk.Tk()  
  
window.title("Demo App")  
  
window.mainloop()
```



Change Icon

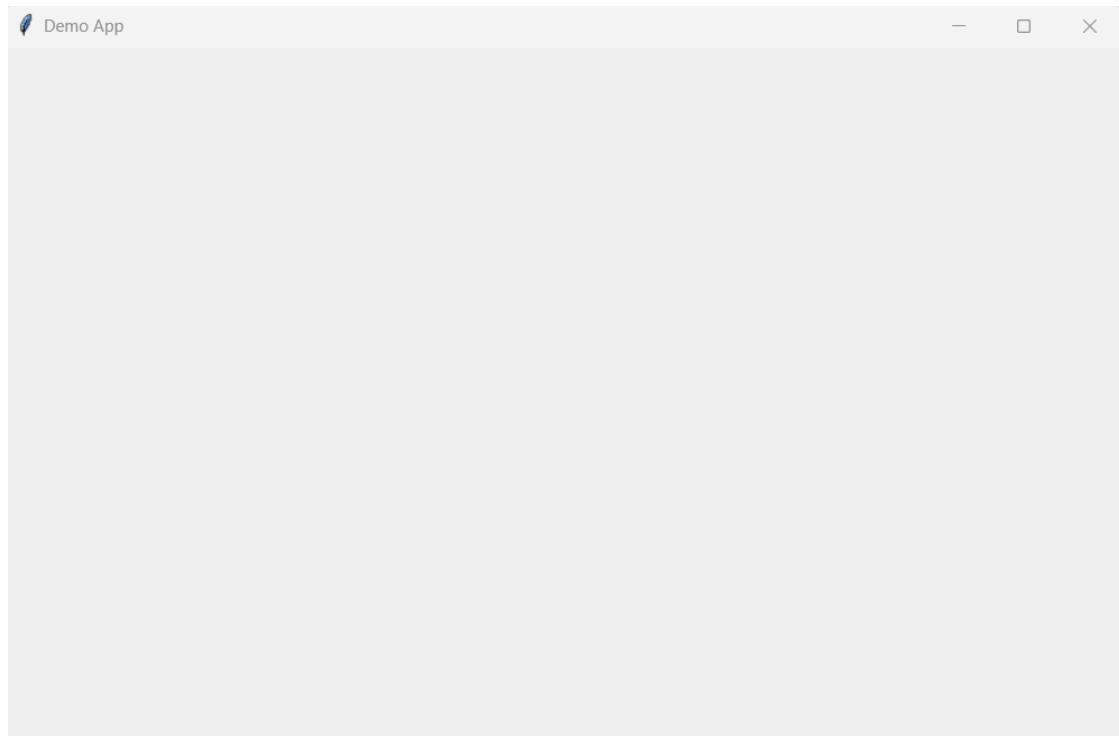
```
window = tk.Tk()  
window.title("Demo App")  
  
icon = tk.PhotoImage(file=r"D:/1111.png")  
window.iconphoto(False, icon)  
  
window.mainloop()
```



GUI

Size of UI

```
window = tk.Tk()  
  
window.title("Demo App")  
  
window.geometry("800x500") # size (Pixel)  
  
window.mainloop()
```



GUI

Tkinter widgets or components

Widget	Description
Button	Creates various buttons in Python Application.
Checkbutton	Select one or more options from multiple options.(Checkbox)
Entry	Allows the user to enter single line of text(Textbox)
Frame	Acts like a container which can be used to hold the other widgets
Label	Used to display non editable text on window
Listbox	Display the list items, The user can choose one or more items.
Radiobutton	Select one option from multiple options.
Text	Allows the user to enter single or multiple line of text(Textarea)
Scale	Creates the graphical slider, the user can slide through the range of values
Toplevel	Used to create and display the top-level windows(Open a new window)

Label

GUI

Tkinter - Widget- Label

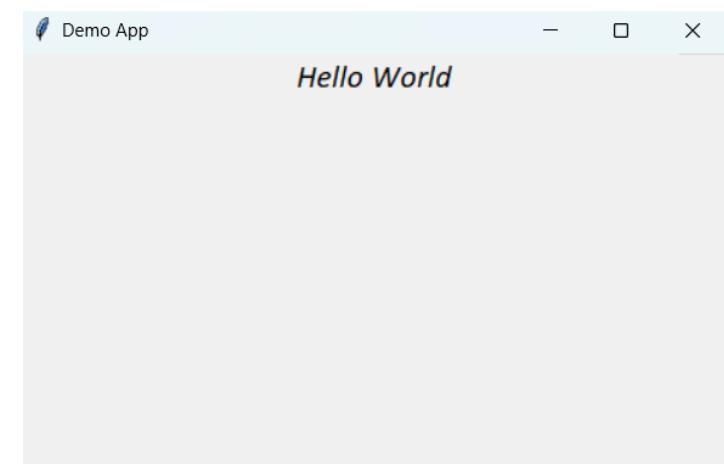
- A **label** is a widget used to **display text or images** on the screen.
- It is a *non-interactive* element, meaning the user **cannot type or click** inside it (it's only for showing information).

```
from tkinter import ttk #widget

window = tk.Tk()
window.title("Demo App")
window.geometry("800x500") # size (Pixel)

heading=ttk.Label(window,text="Hello World")
heading.pack()

window.mainloop()
```



GUI

Layout: Pack/Grid/Place

Pack Method:

- The `pack()` method is used to organise components or widgets in the main window.

Syntax:

`widget.pack(options)`

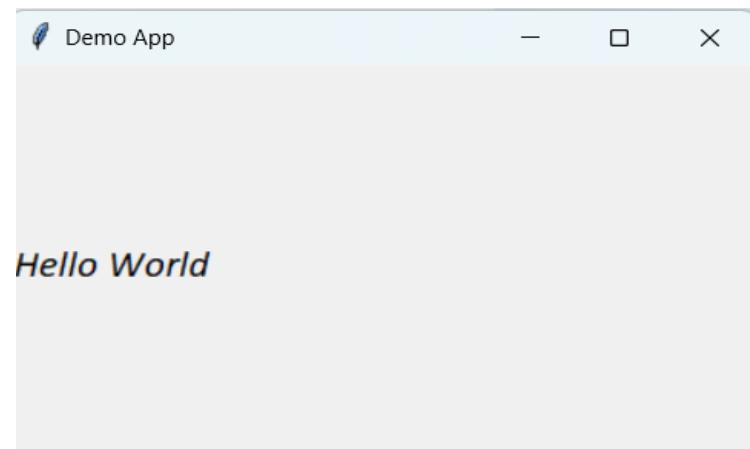
The possible options are:

- **side:** it represents the side to which the widget is to be placed on the window. Side may be Left, right, top (default) or bottom.
- **Padx, pady:** represent the number of pixels to pad the widget outside the widget's border

```
: window = tk.Tk()
window.title("Demo App")
window.geometry("800x500") # size (Pixel)

heading=ttk.Label(master=window,text="Hello World",
                  font="calibri 16 italic")
heading.pack(side="left")

window.mainloop()
```



GUI

Layout

Grid Method:

- The `grid()` method organizes the widgets in a tabular form. We can specify the rows and columns in the method call.

Syntax:

`widget.grid(options)`

The possible options are:

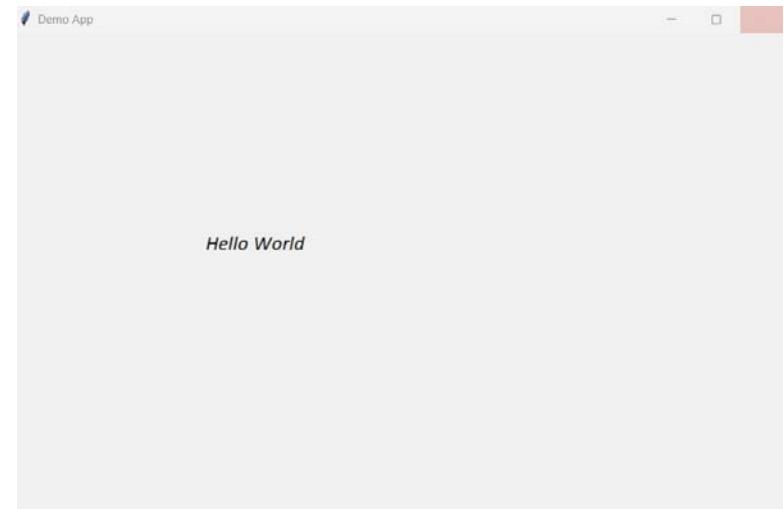
- Column:** Column number in which the widget to be placed
- Padx, pady:** represent the number of pixels to pad the widget outside the widget's border
- Row:** row number in which the widget is to be placed

```
: import tkinter as tk
from tkinter import ttk

window = tk.Tk()
window.title("Demo App")
window.geometry("800x500") # size (Pixel)

heading = ttk.Label(master=window, text="Hello World",
                     font="calibri 16 italic")
heading.grid(row=20, column=0, padx=20, pady=20)

window.mainloop()
```



GUI

Layout

Place Method:

- The **place()** method organizes the widgets to the specific x and y coordinates.

Syntax:

widget.place(x,y)

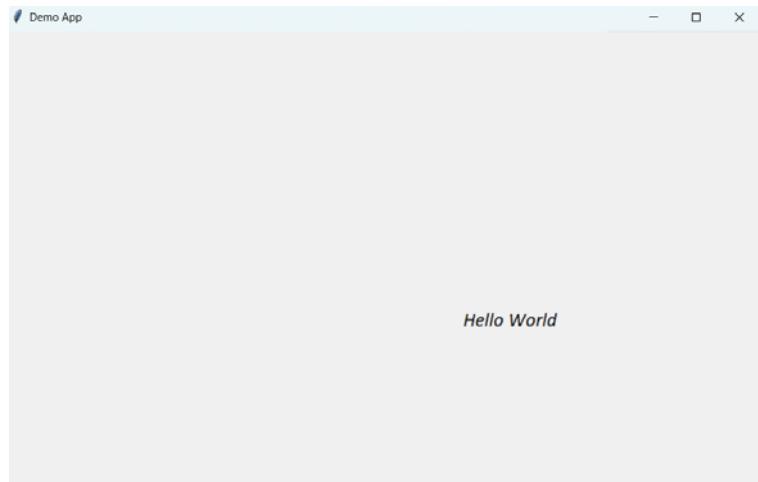
- x,y: refer to the horizontal and vertical offset in pixels.

```
import tkinter as tk
from tkinter import ttk

window = tk.Tk()
window.title("Demo App")
window.geometry("800x500") # size (Pixel)

heading = ttk.Label(master=window, text="Hello World",
                     font="calibri 16 italic")
heading.place(x=500, y=300)

window.mainloop()
```



Hello World

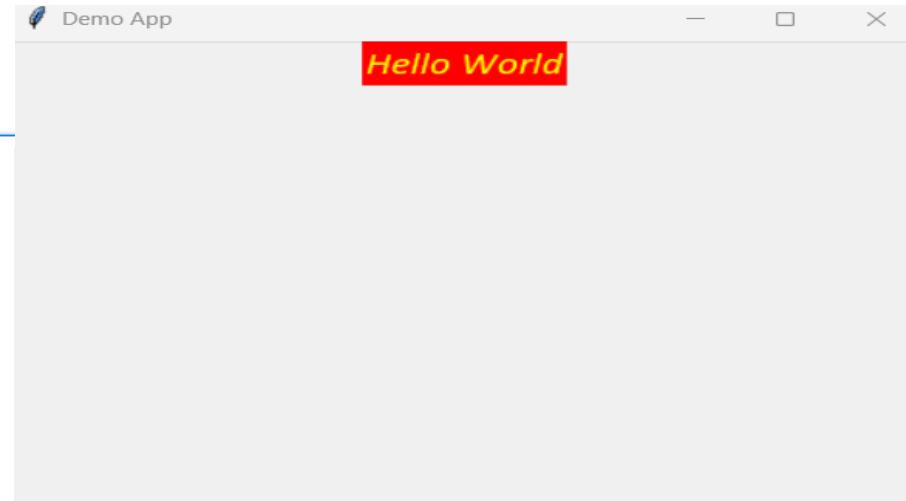
GUI

Tkinter - Widget- Label

```
window = tk.Tk()
window.title("Demo App")
window.geometry("800x500") # size (Pixel)

heading=ttk.Label(master=window,text="Hello World",
                  font="calibri 16 italic",background="red", foreground="yellow")
heading.pack()

window.mainloop()
```



GUI

Tkinter - Widget- Label- Image

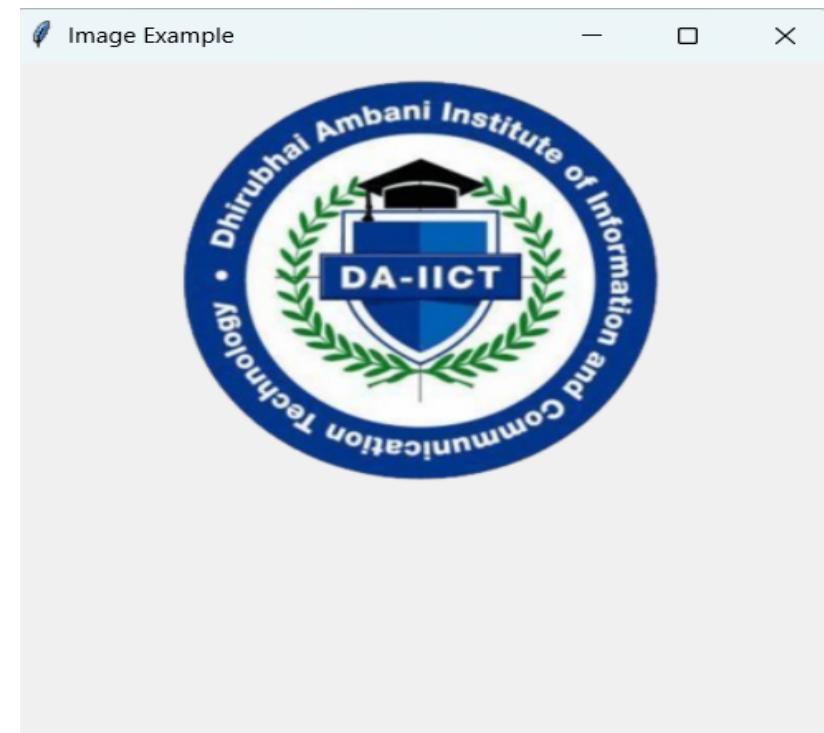
```
import tkinter as tk
from tkinter import ttk

window = tk.Tk()
window.title("Image Example")
window.geometry("400x400")

# Load image (PNG/GIF only)
img = tk.PhotoImage(file=r"D:/1111.png")

# Make a Label to show image
label = ttk.Label(window, image=img)
label.pack()

window.mainloop()
```



GUI

Tkinter - Widget- Label- Image- Size

```
: import tkinter as tk
from tkinter import ttk

window = tk.Tk()
window.title("Image Example")
window.geometry("400x400")

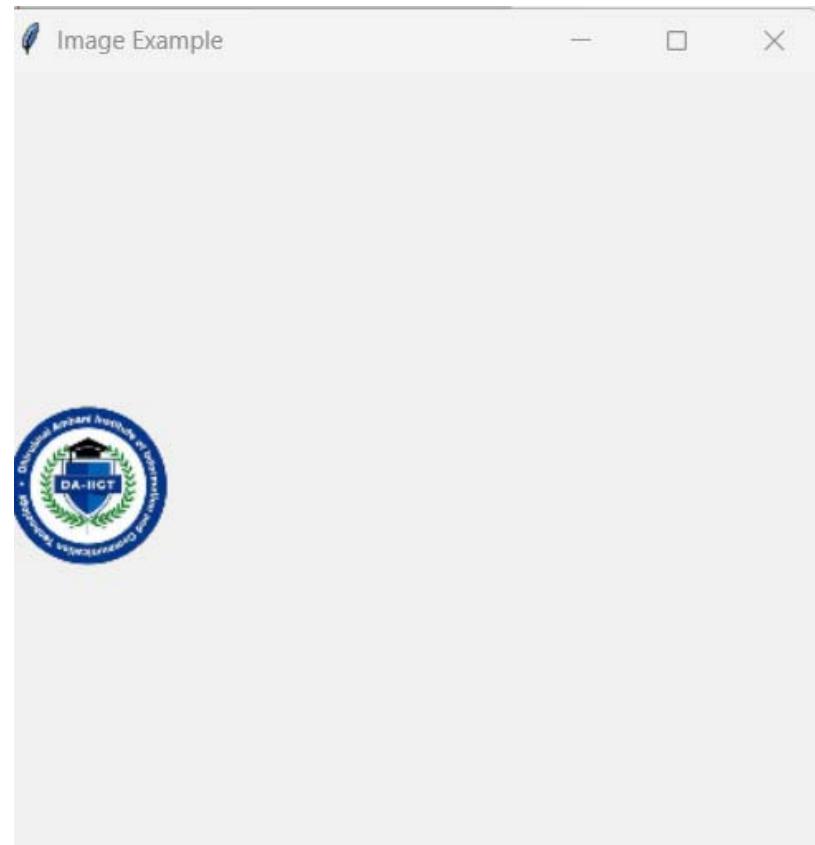
# Load image (PNG/GIF only)
img = tk.PhotoImage(file=r"D:/1111.png")

# to make it smaller
img = img.subsample(3, 3)

#to make it Larger
#img = img.zoom(1, 1)

# Make a Label to show image
label = ttk.Label(window, image=img)
label.pack(side="left")

window.mainloop()
```



Entry

GUI

Tkinter - Widget- Entry

- The entry widget is used to provide a single-line text box to the user to accept a value from them.
- We can use an entry widget to accept a text string from the user.

Syntax:

```
name= Entry(parent, options)
```

GUI

Tkinter - Widget- Entry

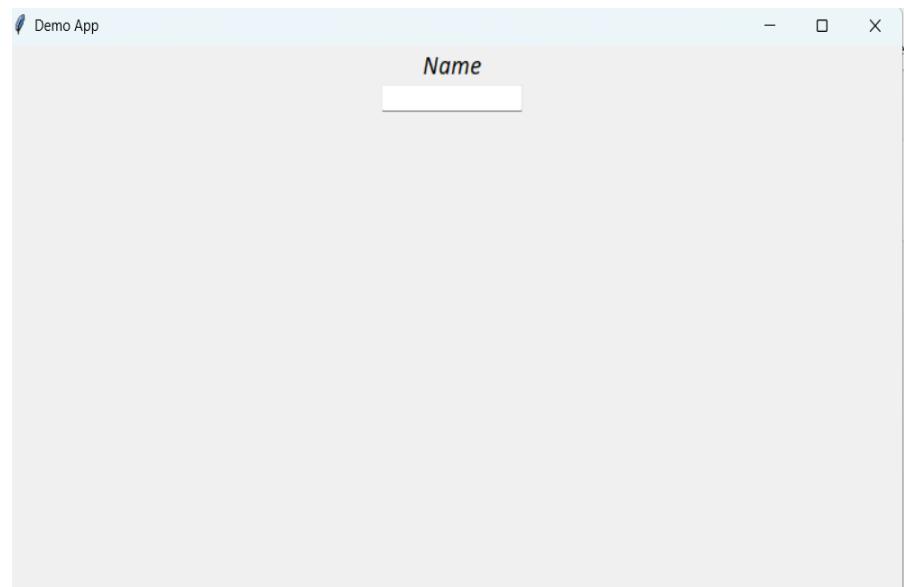
```
window = tk.Tk()
window.title("Demo App")
window.geometry("800x500") # size (Pixel)

heading_1=ttk.Label(window,text="Name",font="calibri 16 italic")

inputBox=ttk.Entry(window)

heading_1.pack()
inputBox.pack()

window.mainloop()
```



GUI

Tkinter - Widget- Entry

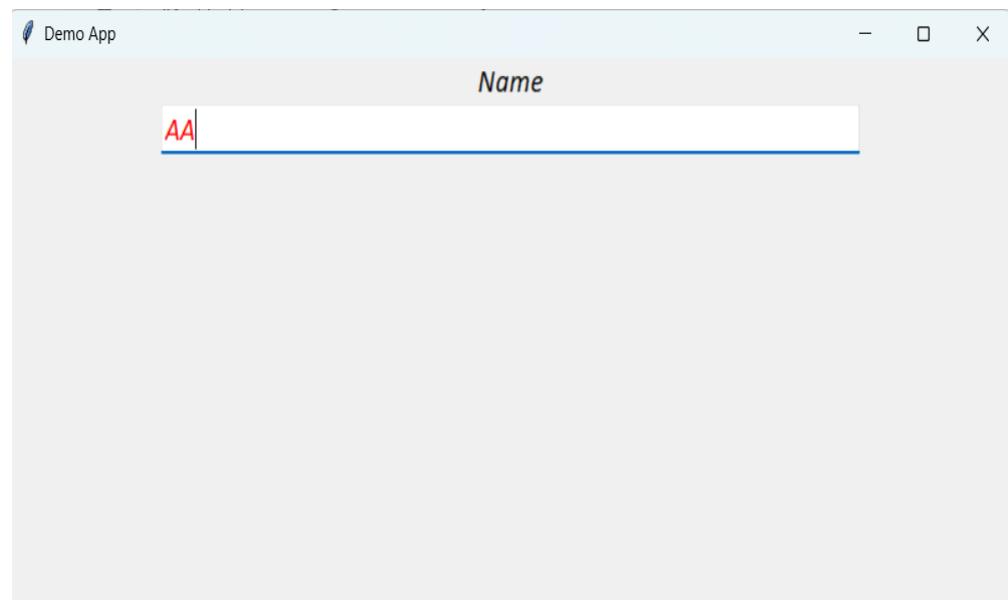
```
window = tk.Tk()
window.title("Demo App")
window.geometry("800x500") # size (Pixel)

heading_1=ttk.Label(window,text="Name",font="calibri 16 italic")

inputBox=ttk.Entry(window, width=50, foreground="red", font="calibri 16 italic")

heading_1.pack()
inputBox.pack()

window.mainloop()
```



GUI

Tkinter - Widget- Entry

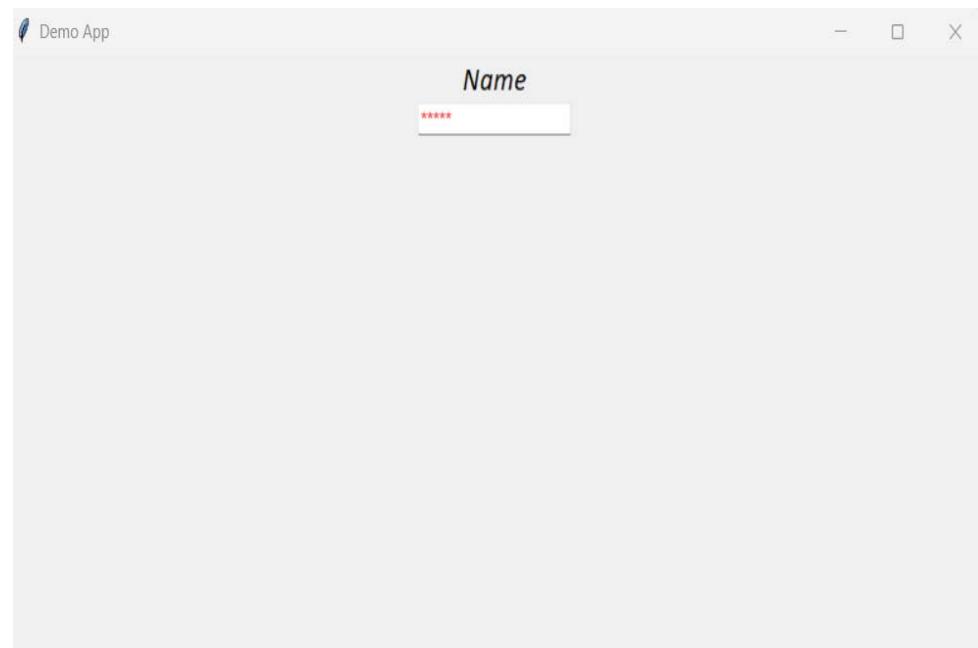
```
window = tk.Tk()
window.title("Demo App")
window.geometry("800x500") # size (Pixel)

heading_1=ttk.Label(window,text="Name",font="calibri 16 italic")

inputBox=ttk.Entry(window, show="*", foreground="red")

heading_1.pack()
inputBox.pack()

window.mainloop()
```



Frames

GUI

Tkinter - Widget- Frame

- The frame widget is used to organise a group of widgets. It acts like a container that can be used to hold the other widgets.
- The rectangular area of the screen is used to organise the widgets in the Python applications.

Syntax:

```
name= Frame(parent, options)
```

GUI

Tkinter - Widget- Frame

```
: import tkinter as tk
from tkinter import ttk

window = tk.Tk()
window.title("Demo App")
window.geometry("500x300") # size (Pixel)

style = ttk.Style()
style.configure("Red1.TFrame", background="red")
Frame_1 = ttk.Frame(window, style="Red1.TFrame")

style.configure("Red.TFrame", background="yellow")
Frame_2 = ttk.Frame(window, style="Red.TFrame")

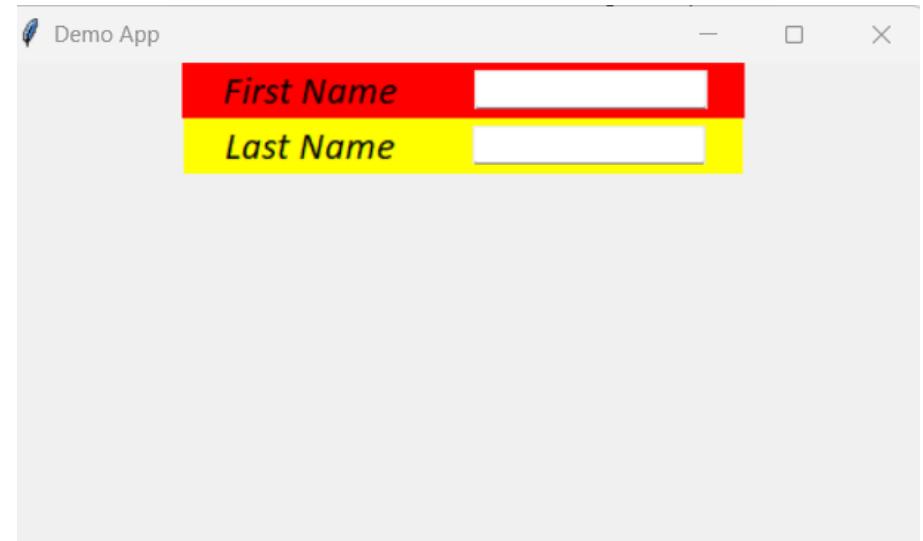
heading_1=ttk.Label(Frame_1,text="First Name",font="calibri 16 italic")
inputBox_1=ttk.Entry(Frame_1)

heading_1.pack(side="left", padx=20)
inputBox_1.pack(side="left", padx=20)

heading_2=ttk.Label(Frame_2,text="Last Name",font="calibri 16 italic")
inputBox_2=ttk.Entry(Frame_2)

heading_2.pack(side="left", padx=20)
inputBox_2.pack(side="left", padx=20)
Frame_1.pack()
Frame_2.pack()

window.mainloop()
```



Buttons

GUI

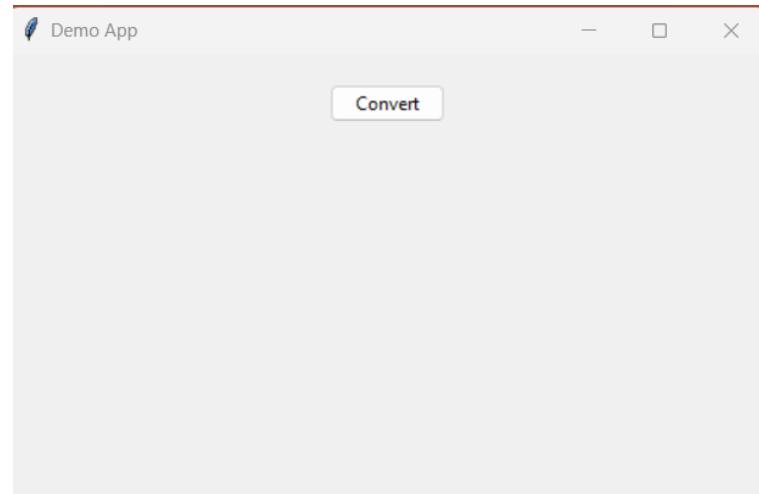
Tkinter - Widget- Button

- The button is used to add various kinds of buttons to the python application. We can also associate a method or a function with a button which is called when the button is pressed.
- Syntax: name = Button(parent, options)

```
: window = tk.Tk()
window.title("Demo App")
window.geometry("500x300") # size (Pixel)

button=ttk.Button(master=window, text="Convert")
button.pack(pady=20)

window.mainloop()
```



GUI

Tkinter - Widget- Button Normal state style

```
window = tk.Tk()
window.title("Demo App")
window.geometry("500x300") # size (Pixel)

style = ttk.Style()

# Normal state style
style.configure("TButton",background="red",
               foreground="green",
               font=("Calibri", 12, "bold"),
               padding=5,
               borderwidth=2)

button=ttk.Button(master=window, text="Convert",style="TButton")
button.pack(pady=20)

window.mainloop()
```

Hover/active style

```
window = tk.Tk()
window.title("Demo App")
window.geometry("500x300") # size (Pixel)

style = ttk.Style()

# Hover / active style
style.map(
    "TButton",
    background=[("active", "blue")],
    foreground=[("active", "yellow")]
)
button=ttk.Button(master=window, text="Convert",style="TButton")
button.pack(pady=20)

window.mainloop()
```



GUI

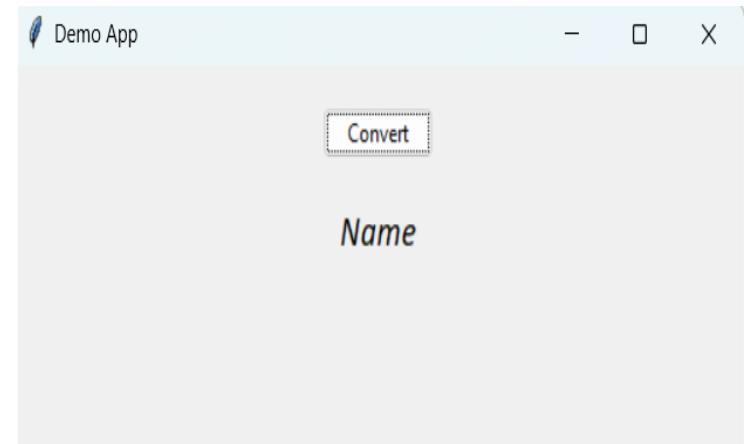
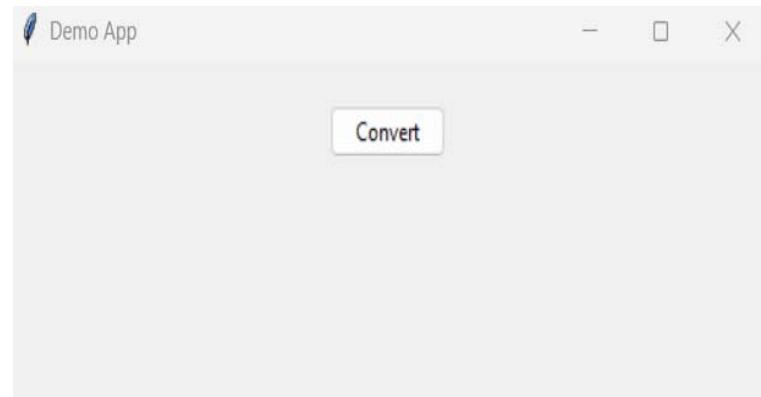
Tkinter - Widget- Button- Call Function

```
window = tk.Tk()
window.title("Demo App")
window.geometry("500x300") # size (Pixel)

def function():
    heading_1=ttk.Label(window,text="Name",font="calibri 16 italic")
    heading_1.pack()

button=ttk.Button(master=window, text="Convert", command=function)
button.pack(pady=20)

window.mainloop()
```



GUI

Tkinter - Widget- Checkbutton

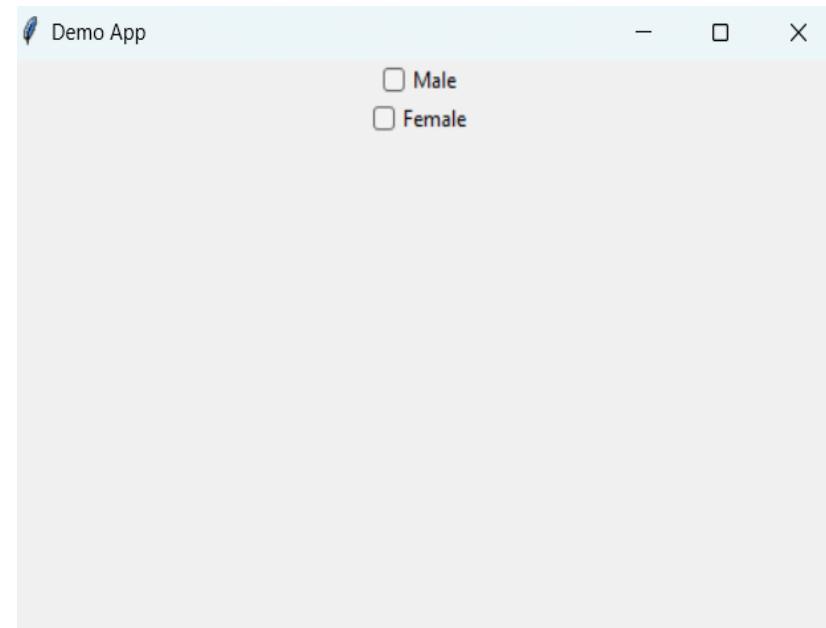
Checkbutton Widget: Toggles on/off states, linked to a variable.

```
: window = tk.Tk()
window.title("Demo App")
window.geometry("800x500") # size (Pixel)

gender_var_1 = tk.BooleanVar()
gender_var_2 = tk.BooleanVar()
male_radio = ttk.Checkbutton(window, text="Male", variable=gender_var_1)
female_radio = ttk.Checkbutton(window, text="Female", variable=gender_var_2)

male_radio.pack()
female_radio.pack()

window.mainloop()
```



GUI

Tkinter - Widget- Checkbutton

```
window = tk.Tk()
window.title("Demo App")
window.geometry("800x500") # size (Pixel)

heading_1 = ttk.Label(master=window, text="Name", font="calibri 16 italic")

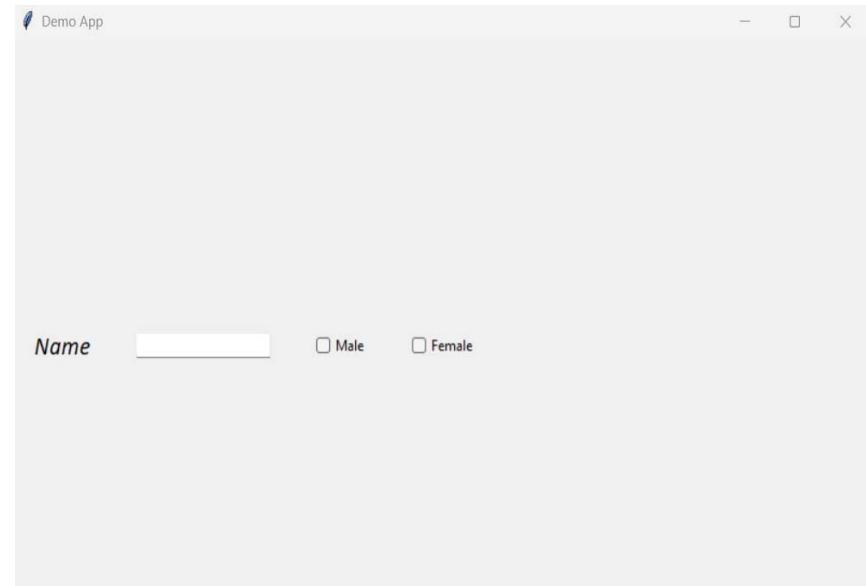
inputBox = ttk.Entry(master=window)

gender_var_1 = tk.BooleanVar()
gender_var_2 = tk.BooleanVar()
male_radio = ttk.Checkbutton(window, text="Male", variable=gender_var_1)
female_radio = ttk.Checkbutton(window, text="Female", variable=gender_var_2)

heading_1.pack(side="left", padx=20)
inputBox.pack(side="left", padx=20)

male_radio.pack(side="left", padx=20)
female_radio.pack(side="left", padx=20)

window.mainloop()
```



GUI

Tkinter - Widget- Radiobutton

Radiobutton Widget: Allows selecting one option from a group, sharing a variable.

```
window = tk.Tk()
window.title("Demo App")
window.geometry("800x500") # size (Pixel)

inputFrame = ttk.Frame(master=window)

heading_1 = ttk.Label(master=inputFrame, text="Name", font="calibri 16 italic")

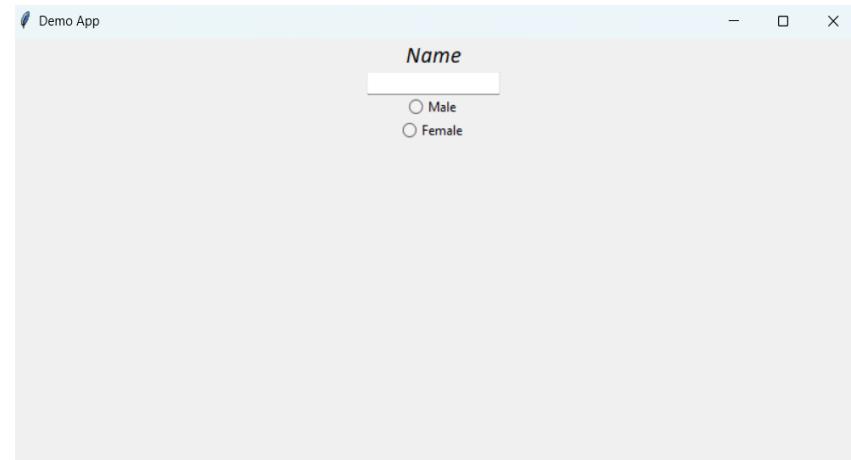
inputBox = ttk.Entry(master=inputFrame)

#  Use ONE variable for both options
gender_var = tk.StringVar(value "") # default empty

male_radio = ttk.Radiobutton(inputFrame, text="Male", variable=gender_var, value="Male")
female_radio = ttk.Radiobutton(inputFrame, text="Female", variable=gender_var, value="Female")
heading_1.pack(padx=20)
inputBox.pack()

male_radio.pack()
female_radio.pack()

inputFrame.pack()
window.mainloop()
```



GUI

Tkinter - Widget- Dropdown

```
import tkinter as tk
from tkinter import ttk

window = tk.Tk()
window.title("Demo App")
window.geometry("500x300") # size (Pixel)

inputFrame = ttk.Frame(master=window)

heading_1 = ttk.Label(master=inputFrame, text="Name", font="calibri 16 italic")
inputBox = ttk.Entry(master=inputFrame)

#  Dropdown menu for Gender
gender_var = tk.StringVar()

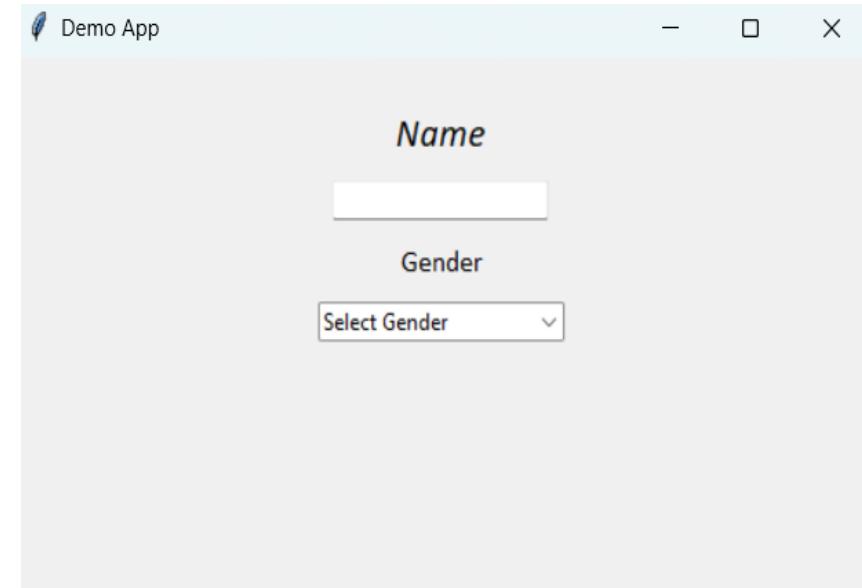
gender_label = ttk.Label(inputFrame, text="Gender", font="calibri 12")
gender_dropdown = ttk.Combobox(inputFrame, textvariable=gender_var, state="readonly")
gender_dropdown['values'] = ("Male", "Female", "Other") # dropdown options
gender_dropdown.set("Select Gender") # default display text

heading_1.pack(padx=20, pady=5)
inputBox.pack(pady=5)

gender_label.pack(pady=5)
gender_dropdown.pack(pady=5)

inputFrame.pack(pady=20)

window.mainloop()
```



GUI

Tkinter - Widget- Listbox

```
import tkinter as tk
from tkinter import ttk

window = tk.Tk()
window.title("Demo App")
window.geometry("500x300") # size (Pixel)

inputFrame = ttk.Frame(master=window)

heading_1 = ttk.Label(master=inputFrame, text="Name", font="calibri 16 italic")
inputBox = ttk.Entry(master=inputFrame)

#  Listbox for Gender
gender_label = ttk.Label(inputFrame, text="Gender", font="calibri 12")

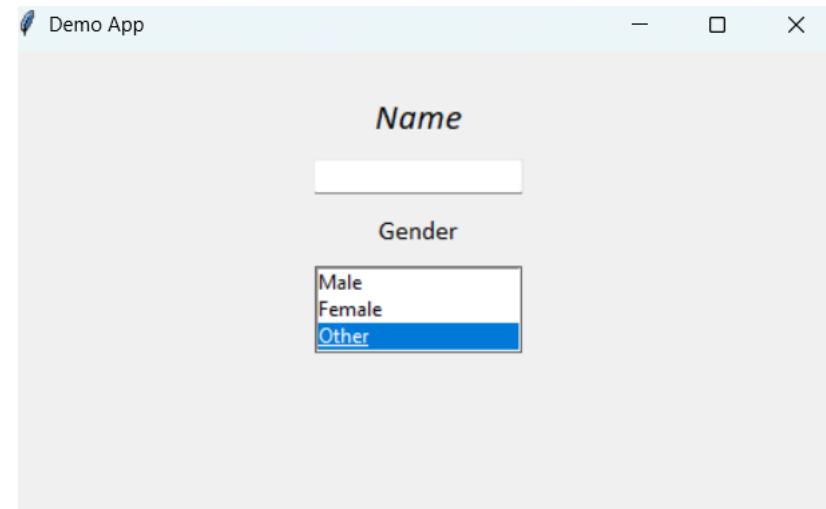
gender_listbox = tk.Listbox(inputFrame, height=3) # height=number of visible items
gender_listbox.insert(1, "Male")
gender_listbox.insert(2, "Female")
gender_listbox.insert(3, "Other")

heading_1.pack(padx=20, pady=5)
inputBox.pack(pady=5)

gender_label.pack(pady=5)
gender_listbox.pack(pady=5)

inputFrame.pack(pady=20)

window.mainloop()
```



GUI

Tkinter - Widget- Slider

```
import tkinter as tk
from tkinter import ttk

window = tk.Tk()
window.title("Demo App")
window.geometry("500x300") # size (Pixel)

inputFrame = ttk.Frame(master=window)

heading_1 = ttk.Label(master=inputFrame, text="Name", font="calibri 16 italic")
inputBox = ttk.Entry(master=inputFrame)

# Scale (slider)
scale_label = ttk.Label(inputFrame, text="Select Age", font="calibri 12")

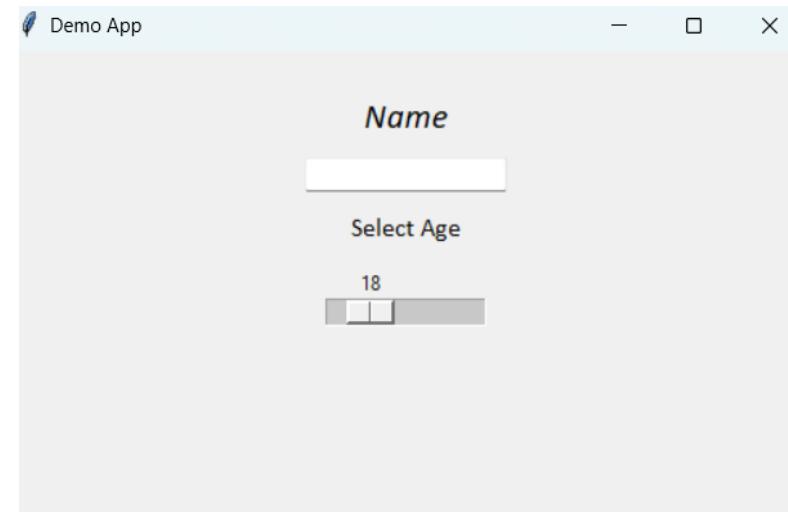
age_var = tk.IntVar(value=18) # default value

age_scale = tk.Scale(
    inputFrame,
    from_=1,
    to=100,
    orient="horizontal", # slider direction
    variable=age_var
)

heading_1.pack(padx=20, pady=5)
inputBox.pack(pady=5)

scale_label.pack(pady=5)
age_scale.pack(pady=5)

inputFrame.pack(pady=20)
window.mainloop()
```



GUI

Tkinter - Widget- Multiline Text Box

```
import tkinter as tk
from tkinter import ttk

window = tk.Tk()
window.title("Demo App")
window.geometry("500x300") # size (Pixel)

inputFrame = ttk.Frame(master=window)

heading_1 = ttk.Label(master=inputFrame, text="Name / Description", font="calibri 16 italic")

# Multiline Text Box
inputBox = tk.Text(inputFrame, width=40, height=5) # width=chars, height=lines

# Scale example from before (still included)
scale_label = ttk.Label(inputFrame, text="Select Age", font="calibri 12")
age_var = tk.IntVar(value=18)
age_scale = tk.Scale(inputFrame, from_=1, to=100, orient="horizontal", variable=age_var)

heading_1.pack(padx=20, pady=5)
inputBox.pack(pady=5)

scale_label.pack(pady=5)
age_scale.pack(pady=5)

inputFrame.pack(pady=20)

window.mainloop()
```

