**Tkinter : Introduction**

Tkinteris the standard GUI (Graphical User Interface) toolkit for the Python programming language. It provides a set of tools for creating graphical user interfaces and is commonly used for developing desktop applications with graphical interfaces.

**1.Import Tkinter:**

import tkinter as tk

Import the Tkinter module and alias it as tk for convenience.

**2.Create the main window**

window = tk.Tk()

Create the main window for your Tkinter application.

**3. Create a label widget:**

label = tk.Label(window, text="Hello, Tkinter!")

Create a Label widget with the specified text and associate it with the main window.

**4.Pack the label into the window:**

label.pack()

Use the pack() method to organize and display the label within the window.

**5.Start the Tkinter event loop:**

window.mainloop()

Initiate the Tkinter event loop, which handles user inputs and keeps the GUI responsive.

**Note:from tkinter import \*is a common way to import all classes and functions from the Tkinter module.**

**Example:**

****from** tkinter **import** \***

top=Tk()

top.mainloop()

**Tkinter Widget**

|  |  |  |
| --- | --- | --- |
| SN | Widget | Description |
| 1 | [Button](https://www.javatpoint.com/python-tkinter-button) | The Button is used to add various kinds of buttons to the python application. |
| 2 | [Canvas](https://www.javatpoint.com/python-tkinter-canvas) | The canvas widget is used to draw the canvas on the window. |
| 3 | [Checkbutton](https://www.javatpoint.com/python-tkinter-checkbutton) | The Checkbutton is used to display the CheckButton on the window. |
| 4 | [Entry](https://www.javatpoint.com/python-tkinter-entry) | The entry widget is used to display the single-line text field to the user. It is commonly used to accept user values. |
| 5 | [Frame](https://www.javatpoint.com/python-tkinter-frame) | It can be defined as a container to which, another widget can be added and organized. |
| 6 | [Label](https://www.javatpoint.com/python-tkinter-label) | A label is a text used to display some message or information about the other widgets. |
| 7 | [ListBox](https://www.javatpoint.com/python-tkinter-listbox) | The ListBox widget is used to display a list of options to the user. |
| 8 | [Menubutton](https://www.javatpoint.com/python-tkinter-menubutton) | The Menubutton is used to display the menu items to the user. |
| 9 | [Menu](https://www.javatpoint.com/python-tkinter-menu) | It is used to add menu items to the user. |
| 10 | [Message](https://www.javatpoint.com/python-tkinter-message) | The Message widget is used to display the message-box to the user. |
| 11 | [Radiobutton](https://www.javatpoint.com/python-tkinter-radiobutton) | The Radiobutton is different from a checkbutton. Here, the user is provided with various options and the user can select only one option among them. |
| 12 | [Scale](https://www.javatpoint.com/python-tkinter-scale) | It is used to provide the slider to the user. |
| 13 | [Scrollbar](https://www.javatpoint.com/python-tkinter-scrollbar) | It provides the scrollbar to the user so that the user can scroll the window up and down. |
| 14 | [Text](https://www.javatpoint.com/python-tkinter-text) | It is different from Entry because it provides a multi-line text field to the user so that the user can write the text and edit the text inside it. |
| 14 | [Toplevel](https://www.javatpoint.com/python-tkinter-toplevel) | It is used to create a separate window container. |
| 15 | [Spinbox](https://www.javatpoint.com/python-tkinter-spinbox) | It is an entry widget used to select from options of values. |
| 16 | [PanedWindow](https://www.javatpoint.com/python-tkinter-panedwindow) | It is like a container widget that contains horizontal or vertical panes. |
| 17 | [LabelFrame](https://www.javatpoint.com/python-tkinter-labelframe) | A LabelFrame is a container widget that acts as the container |
| 18 | [MessageBox](https://www.javatpoint.com/python-tkinter-messagebox) | This module is used to display the message-box in the desktop based applications. |

**1. Label Widget:**

Tkinter Label is a widget that is used to implement display boxes where you can place text or images.

****Syntax:****

*w = Label ( master, option, … )*

Example:

from tkinter import \*

root= Tk()

label=Lable(root,text=”Message to display”)

label.pack()

root.mainloop()

**2.Button Widget**

import tkinter as tk

# Create the main window

window = .Tk()

window.title("Simple Tkinter Example")

# Create a frame

frame =Frame(window)

frame.pack()

# Create a button widget inside the frame

button = Button(frame, text="Click me!")

button.pack()

# Create a label widget

label = tk.Label(window, text="Hello, Tkinter!")

label.pack()

# Start the Tkinter event loop

window.mainloop()

**3.Entry and Grid Widget**

from tkinter import \*

frame=Tk()

l1=Label(frame,text=”username”)

l2=Label(frame,text=”passowrd”)

entry=Entry(frame)

entry2=Entry(frame)

l1.grid(row=0,column=0)

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

l2.grid(row=1,column=0)

entry2.grid(row=1,column=1)

btn=Button(frame,text=”login”,fg=”red”,bg=”black”

btn.grid(row=2,column=0)

frame.mainloop()

**4.Message alert**

The messagebox module is used to display the message boxes in the python applications. There are the various functions which are used to display the relevant messages depending upon the application requirements.

### **1**. **showinfo()**

### **2. showwarning()**

### **3. showerror()**

### **4. askquestion()**

### **5. askokcancel()**

### **6. askyesno()**

### **7. askretrycancel()**

**Example:**

**from** tkinter **import** **\***

**from** tkinter **import** messagebox

root **=** Tk()

w **=** Label(root, text **=**'GeeksForGeeks')

w.pack()

messagebox.showinfo("showinfo", "Information")

messagebox.showwarning("showwarning", "Warning")

messagebox.showerror("showerror", "Error")

messagebox.askquestion("askquestion", "Are you sure?")

messagebox.askokcancel("askokcancel", "Want to continue?")

messagebox.askyesno("askyesno", "Find the value?")

messagebox.askretrycancel("askretrycancel", "Try again?")

root.mainloop()

**5.CheckButtton**

**Eg:**

from tkinter import \*

root =Tk()

# Create a BooleanVar to store the check state

check\_var = BooleanVar()

check\_button = Checkbutton(root, text="Check me", )

check\_button.pack()

root.mainloop()

**6.OptionMenu**

OptionMenu is basically a dropdown or popup menu that displays a group of objects on a click or keyboard event and lets the user select one option at a time.

**Eg:**

import tkinter as tk

root = tk.Tk()

root.title("Simple OptionMenu Example")

# Create a StringVar to store the selected option

selected\_option = tk.StringVar()

# Set default value and available options

selected\_option.set("Option 1")

options = ["Option 1", "Option 2", "Option 3"]

# Create OptionMenu widget

option\_menu = tk.OptionMenu(root, selected\_option, \*options)

option\_menu.pack(pady=10)

# Start the Tkinter event loop

root.mainloop()

**7.Radio Button**

import tkinter as tk

root = tk.Tk()

root.title("RadioButton Example")

# Create a StringVar to store the selected option

selected\_option = tk.StringVar()

# Set default value and available options for radio buttons

selected\_option.set("Option 1")

options = ["Option 1", "Option 2", "Option 3"]

# Create and place radio buttons

for option in options:

radio\_button = tk.Radiobutton(root, text=option, variable=selected\_option, value=option)

radio\_button.pack(pady=5)

# Create a Label to display the selected option

label = tk.Label(root, textvariable=selected\_option)

label.pack(pady=10)

# Start the Tkinter event loop

root.mainloop()

**8.ListBox**

import tkinter as tk

root = tk.Tk()

root.title("Listbox Example")

# Create a Listbox widget

listbox = tk.Listbox(root, selectmode=tk.SINGLE)

listbox.pack(pady=10)

# Add items to the Listbox

items = ["Item 1", "Item 2", "Item 3", "Item 4", "Item 5"]

for item in items:

listbox.insert(tk.END, item)

# Start the Tkinter event loop

root.mainloop()

**9.Menu**

import tkinter as tk

root = tk.Tk()

root.title("Menu Example")

# Create a menu bar

menu\_bar = tk.Menu(root)

root.config(menu=menu\_bar)

# Create a File menu

file\_menu = tk.Menu(menu\_bar, tearoff=0)

menu\_bar.add\_cascade(label="File", menu=file\_menu)

# Add items to the 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.destroy)

# Create an Edit menu

edit\_menu = tk.Menu(menu\_bar, tearoff=0)

menu\_bar.add\_cascade(label="Edit", menu=edit\_menu)

# Add items to the Edit menu

edit\_menu.add\_command(label="Cut")

edit\_menu.add\_command(label="Copy")

edit\_menu.add\_command(label="Paste")

# Start the Tkinter event loop

root.mainloop()

**10.Panel**

import tkinter as tk

root = tk.Tk()

root.title("Panel Example")

# Create a frame to resemble a panel

panel\_frame = tk.Frame(root, bg="lightgray", padx=20, pady=20)

panel\_frame.pack(padx=10, pady=10)

# Add widgets to the panel frame

label = tk.Label(panel\_frame, text="This is a panel", font=("Helvetica", 16), bg="lightgray")

label.pack(pady=10)

button = tk.Button(panel\_frame, text="Click me", command=root.destroy)

button.pack()

# Start the Tkinter event loop

root.mainloop()