In ***Tkinter***, widgets are the basic building blocks of a Graphical User Interface (GUI) application. A widget can be anything from a simple text label, a button, or a text entry box to more complex elements like scroll bars, tabs, and menus. Each widget in Tkinter is created as an instance of a widget class, and they are used to display information or provide a way for the user to interact with the application.

Label Widget

A Label widget is used to display text or images on the screen. It is a simple way to provide information to the user without allowing them to edit it.

Example:

```
import tkinter as tk

root = tk.Tk()
label = tk.Label(root, text="Hello, Tkinter!")
label.pack()
root.mainloop()
```

This code creates a window with a label that displays the text "Hello, Tkinter!".

Button Widget

A Button widget is used to add clickable buttons to your application. When clicked, it can be configured to call a function or a method.

Example:

```
import tkinter as tk

def on_button_click():
    print("Button clicked")

root = tk.Tk()
button = tk.Button(root, text="Click Me", command=on_button_click)
button.pack()
root.mainloop()
```

This code creates a button that, when clicked, prints "Button clicked" to the console.

Entry Widget

An Entry widget is used for single-line text entry from the user. It's useful for forms where the user needs to input text data.

Example:

```
import tkinter as tk

root = tk.Tk()
entry = tk.Entry(root)
entry.pack()
root.mainloop()
```

This code creates a text entry box where users can type in text.

RadioButton Widget

A RadioButton widget is used to implement one-of-many selection as it allows only one option to be selected among a group.

Example:

```
import tkinter as tk

def selected():
    print(var.get())

root = tk.Tk()
var = tk.StringVar()
rb1 = tk.Radiobutton(root, text="Option 1", variable=var, value="1", command=selected)
rb1.pack()
rb2 = tk.Radiobutton(root, text="Option 2", variable=var, value="2", command=selected)
rb2.pack()
root.mainloop()
```

This code creates two radio buttons, and when one is selected, it prints the value of the selected option.

CheckBox Widget (CheckButton)

A CheckButton widget displays a number of options as toggle buttons from which the user can select any number of options.

Example:

```
import tkinter as tk

def display_selected():
    print(var1.get(), var2.get())

root = tk.Tk()
var1 = tk.IntVar()
cb1 = tk.Checkbutton(root, text="Option 1", variable=var1, command=display_selected)
cb1.pack()
var2 = tk.IntVar()
cb2 = tk.Checkbutton(root, text="Option 2", variable=var2, command=display_selected)
cb2.pack()
root.mainloop()
```

This code creates two checkboxes, and when either is toggled, it prints the state of both checkboxes (1 for checked, 0 for unchecked).

Frame

A Frame is a container widget that organizes and groups other widgets. It can be used to improve the layout of your application by dividing the interface into sections.

Example:

```
import tkinter as tk

root = tk.Tk()
frame = tk.Frame(root)
frame.pack()

button = tk.Button(frame, text="Inside Frame")
button.pack()

root.mainloop()
```

This code creates a Frame widget and places a Button inside it

Listbox

The Listbox widget is used to display a list of items from which a user can select one or more items.

Example:

```
import tkinter as tk

root = tk.Tk()
listbox = tk.Listbox(root)
listbox.pack()

listbox.insert(1, "Item 1")
listbox.insert(2, "Item 2")

root.mainloop()
```

This code creates a Listbox and inserts two items into it

Scrollbar

A Scrollbar widget provides a slide controller that is used to implement vertical or horizontal scrolling bars.

Example:

```
import tkinter as tk

root = tk.Tk()
scrollbar = tk.Scrollbar(root)
scrollbar.pack(side=tk.RIGHT, fill=tk.Y)

listbox = tk.Listbox(root, yscrollcommand=scrollbar.set)
for i in range(100):
    listbox.insert(tk.END, f"Item {i}")
listbox.pack(side=tk.LEFT, fill=tk.BOTH)
```

```
scrollbar.config(command=listbox.yview)

root.mainloop()
```

This code creates a vertical Scrollbar for a Listbox filled with items

Toplevel

The Toplevel widget is used to create additional windows that are separate from the main application window.

Example:

```
import tkinter as tk

def create_new_window():
    new_window = tk.Toplevel()
    new_window.title("New Window")
    tk.Label(new_window, text="This is a new window").pack()

root = tk.Tk()
tk.Button(root, text="Open New Window", command=create_new_window).pack()

root.mainloop()
```

This code creates a button that, when clicked, opens a new window

Scale

The Scale widget allows the user to select a numerical value by moving a slider knob along a scale.

Example:

```
import tkinter as tk

def show_value(val):
    print(val)

root = tk.Tk()
scale = tk.Scale(root, from_=0, to=100, orient=tk.HORIZONTAL,
command=show_value)
scale.pack()

root.mainloop()
```

This code creates a horizontal Scale widget that prints the current value when moved

Spinbox

The Spinbox widget allows the user to select a value from a given set of values by clicking the up and down arrows.

Example:

```
import tkinter as tk

root = tk.Tk()
spinbox = tk.Spinbox(root, from_=0, to=10)
spinbox.pack()

root.mainloop()
```

This code creates a Spinbox widget with values ranging from 0 to 10

Text

The Text widget is used for multi-line text input or display, allowing for text editing and formatting features.

Example:

```
import tkinter as tk

root = tk.Tk()
text = tk.Text(root, height=10, width=30)
text.pack()

root.mainloop()
```

This code creates a Text widget where users can input or display multiple lines of text