

OCTOBER 2025  
Day - 44

09/07

## Managing Widgets and Events in Tkinter.

26 SUNDAY

Widget states:

- state
  - ↳ Normal : → Default .
  - enabled
  - user interactive
- ↳ Disabled : → Non-Interactive  
→ can not be used .

↳ active : <sup>27 MONDAY</sup> being interacted

- When clicked
- button pressed
- or
- Hovered ,

Syntax :

Widget.config(state = 'disabled')

Keyboard and Mouse events :

OCTOBER 2025

TUESDAY 28

## Creating Custom Widgets :

- can create custom widgets combining multiple existing widgets.
- Suits your need.
- Interactive and reusable.

## \*\*kwargs

EDNESDAY 29

↳ Multiple keyword arguments.

## Super() in python :

- To reuse parent class
- ~~Extend~~ inheritance, not replace it
- both method from parent and child class are used
- Without Super(), we need to call parent class manually.

OCTOBER 2025

Widgets functionalities.

→ customize to triggered

30 THURSDAY

Bind - Handling events  
(mouse, keyboard)

<Button-1> - left mouse btn

fg - fore ground  
(text) color

<Button-2> - Middle mouse  
btn

31 FRIDAY

<Button-3> - Right mouse btn.

NOVEMBER 2025  
Day - 45.

- Advanced Features in Tkinter 01 SATURDAY
- Multi-threaded app.
    - ↳ multiple task at same time
  - Runs independently.
  - Share the same memory.
  - Simultaneously.

Benefits: Fast.

Better responsive

Efficient I/O

02 SUNDAY

Shared memory

Issue

→ Complexity

→ Data Corruption

→ Need synchronization

① use after() to handle threads

safely → schedules a function

widget.after(time, function)  
→ Ensure safe updates

## NOVEMBER 2025

MONDAY 03

Tkinter and SQLite Integration:

→ connecting Tkinter with SQLite

for storing and retrieving data

→ Python In-build <sup>module</sup> SQLite

→ To create DB-driven application

→ SQLite store, retrieve and manipulate data.

TUESDAY 04

import sqlite3

Conn = sqlite3 . connect ("mydb.db")

cursor = conn . cursor ()

cursor . execute (

" CREATE TABLE IF NOT EXISTS

users (Id INTEGER PRIMARY

name TEXT,

age INTEGER )"

Non/Install  
SQLite viewer