# Beginners Python Cheat Sheet tkinter GUI module

## Basic Definitions:

**Tkinter:** Python library for creating graphical user interfaces (GUIs).
**Widget:** An element in a GUI (e.g., button, label, entry).
**Root Window**: The main window that contains all widgets.
**Parent/Child Relationship:**

- Every widget belongs to a parent container (root window or frame).
- Widgets inside a parent are called children.

**Frame:** Container used to organize widgets.
**Geometry Manager:** Methods to arrange widgets (pack, grid, place)

## Root Window

root = tk.Tk() Creates a window named "root"
root.title("Enter the Title for your window here")
root.geometry("**Width**x**Height**+**X**+**Y**") Sets size and position.
root.attributes("-attribute", value) Allows setting attributes
"-alpha", value (0.0 to 1.0) Controls transparency
"-topmost", True/False Makes it stay on top
"-fullscreen", True/False Toggles fullscreen mode
root.resizeable(width=True/False, height=True/False)
root.iconphoto(True, PhotoImage_Object) Sets the Window Icon
root.destroy() Closes the window
root.withdraw() Hides the window (not destroys)
root.deiconify() Restores a hidden or minimized window
root.lift() Bring the window to the front
root.update() Forces tkinter to process pending events
root.mainloop() Starts the eventloop, listening for events.

## Frames

widget = tk.Frame(*parent*, *attributes*) Creates a frame
width, height Creates dimensions of the frame
bg or background Background color
bd or borderwidth Width of the Border around frame
relief Style of the border
cursor changes mouse appearance when over frame
padx, pady creates padding around the frame

## tkinter Variables

var = tk.IntVar() creates an Integer Variable
var = tk.DoubleVar() creates a Float Variable
var = tk.StringVar() creates a String Variable
var = tk.BooleanVar() creates a Boolean Variable

var.get() returns the current value
var.set(x) sets the value to x
var.trace_add("mode", callback). calls function when var changes
var.trace_remove("mode", callback_name)
removes a trace callback
modes: write – when the variable is modified
read – when the variable is read
unset – when the variable is deleted

## Grid Method

widget.grid(*attributes*) sets the item into a grid
column the col number the widget occupies, start at 0.
columnspan number of columns the widget takes
in_ register widget as a child
ipadx, ipady internal padding
padx, pady external padding
row the row number the widget occupies, start at 0.
rowspan number of rows the widget takes
sticky determines how to stick in a cell
uses tk.N, tk.NE, tk.S, tk.SE method.
To make it stretch to fit, use tk.E+tk.W to stretch horizontally.
To make it stretch to fit, use tk.N+tk.S to stretch vertically.

## Pack Method

widget.pack(*attributes*) packs the item
expand (0, or 1) should expand to fill space
fill (NONE, X, Y, BOTH) How to resize as child
side (TOP, BOTTOM, RIGHT, LEFT)
Which side of the parent is used for child
in_ register widget as a child
ipadx, ipady internal padding
padx, pady external padding
anchor Specifies where it should be placed
used N,S,E,W,CENTER syntax

## Place Method

widget.place(*attributes*) places the item
anchor Specifies where it should be placed
used N,S,E,W,CENTER syntax
bordermode (INSIDE, OUTSIDE)
specifies if the border should be inside or outside
in_ register widget as a child
relwidth, relheight Float [0.0, 1.0]
size of the child widget related to the parent
relx, rely Float [0.0, 1.0]
position of the child widget related to the parent
width, height Absolute height/width of widget
x, y Absolute position of the widget

# Widgets

## Common Methods

config(attrubute) configure options after creation
cget(attribute) gets value of an option
destroy() removes the widget
bind(event, handler) binds events
after(ms, func) calls a function after a delay
update() manually refresh the widget

## Scale

widget = tk.Scale(*parent*, *attribute*) creates a Scale Slider

**Attributes**

from_, to, orient, length, tickinterval, resolution, variable, showvalue, troughcolor, sliderlength, fg, bg, font, width, height, relief, bd

## Attributes used in Widgets

**Text**

**text** → Display text for Label, Button, Radiobutton, Checkbutton
**image** → Display image (PhotoImage or BitmapImage)
**compound** → Combines text and image (top, bottom, left, right, center)
**justify** → Align multi-line text (left, center, right)
**wraplength** → Wrap text after X pixels

**Font**

**font** → Font family, size, style (e.g., ("Arial", 12, "bold"))
**fg** → Foreground/text color
**bg** → Background color
**highlightbackground** → Border color when not focused
**highlightcolor** → Border color when focused
**activeforeground** → Text color when active
**activebackground** → Background color when active

**Dimensions & Placement**

**width** → Width of widget (chars for Entry, pixels for Scale)
**height** → Height of widget (chars/lines for Label)
**padx, pady** → Padding inside geometry managers (pack, grid)
**relief** → Border style (flat, raised, sunken, groove, ridge)
**bd** → Border width
**anchor** → Position of content inside widget (n, s, e, w, center)

**State & Interaction**

**state** → Widget state: normal, disabled, active
**command** → Function executed on action
**variable** → Linked IntVar, StringVar, or FloatVar
**value** → Value of a Radiobutton when selected
**onvalue** / **offvalue** → Values for Checkbutton when checked/unchecked
**show** → Mask character for Entry (e.g., "*" for password)
**validate** → Input validation mode (focus, key, etc.)
**validatecommand** → Function executed for validation

## Label

widget = tk.Label(*parent*, *attribute*) creates a text label

**Attributes**

text, image, compound, justify, wraplength, fg, bg, font, width, height, anchor, relief, bd

## Button

widget = tk.Button(*parent*, *attribute*) creates a Button

**Attributes**

text, command, state, activeforeground, activebackground, fg, bg, font, width, height, relief, bd

## Entry

widget = tk.Entry(*parent*, *attribute*) creates a Text Entry Box

**Attributes**

width, show, textvariable, fg, bg, font, relief, bd, state

## Option Box

widget = ttk.Optionbox(*parent*, *attribute*) creates a Dropdown Menu

**Attributes**

variable, values, fg, bg, font, width, height, relief, bd

## Radio Button

widget = tk.Radiobutton(*parent*, *attribute*) creates a Radio Button

**Attributes**

text, variable, value, state, fg, bg, font, width, height, relief, bd

## Check Button

widget = tk.Checkbutton(*parent*, *attribute*) creates a Check Button

**Attributes**

text, variable, onvalue, offvalue, state, fg, bg, font, width, height, relief, bd

## Spinbox

widget = tk.Spinbox(*parent*, *attribute*) creates a Spinbox Selector

**Attributes**

from_, to, increment, width, state, fg, bg, font, relief, bd, textvariable

# Beginners Python Cheat Sheet tkinter GUI module - Examples

```
import tkinter as tk
from tkinter import ttk

root = tk.Tk()
root.title("Tkinter Widgets & Geometry Managers")
root.geometry("700x600+500+200")
root.configure(bg="#e0e0e0")
```

## FRAME 1 — Using PACK

```
packFrame = tk.LabelFrame(root, text="Pack Geometry", padx=10, pady=10,
bg="#f0f8ff")
packFrame.pack(fill="x", pady=10, padx=10)
```

### Label using pack

```
label = tk.Label(
    packFrame,
    text="This Label uses PACK",
    fg="white",
    bg="darkblue",
    font=("Arial", 12, "bold"),
    width=25,
    height=2,
    relief="ridge",
    bd=3)
label.pack(pady=5)
```

### Button using pack

```
def button_clicked():
    label.config(text="Button Clicked!")

button = tk.Button(
    packFrame,
    text="Click Me!",
    command=button_clicked,
    fg="white",
    bg="green",
    activeforeground="yellow",
    activebackground="darkgreen",
    font=("Arial", 11, "bold"),
    relief="raised",
    bd=4)
button.pack(pady=5)
```

### Entry using pack

```
entry_var = tk.StringVar()
entry = tk.Entry(
    packFrame,
    textvariable=entry_var,
    fg="blue",
    bg="lightyellow",
    font=("Courier", 12),
    width=20,
    relief="sunken",
    bd=3)
entry.insert(0, "Type here...")
entry.pack(pady=5)
```

## FRAME 2 — Using GRID

```
gridFrame = tk.LabelFrame(root, text="Grid Geometry", padx=10, pady=10,
bg="#fffaf0")
gridFrame.pack(fill="x", pady=10, padx=10)
```

### Option Menu

```
options = ["Apples", "Bananas", "Cherries"]
selected_option = tk.StringVar(value=options[0])
option_menu = ttk.Combobox(
    gridFrame,
    textvariable=selected_option,
    values=options,
    font=("Arial", 12),
    width=15,
    state="readonly"
)
option_menu.grid(row=0, column=0, padx=5, pady=5)
```

### Radio Buttons

```
radio_var = tk.StringVar(value="A")
radio1 = tk.Radiobutton(
    gridFrame,
    text="Option A",
    variable=radio_var,
    value="A",
    fg="black",
    bg="#fffaf0",
    font=("Arial", 11))
radio1.grid(row=0, column=1, padx=5, pady=5)

radio2 = tk.Radiobutton(
    gridFrame,
    text="Option B",
    variable=radio_var,
    value="B",
    fg="black",
    bg="#fffaf0",
    font=("Arial", 11))
radio2.grid(row=0, column=2, padx=5, pady=5)
```

### Check Button

```
check_var = tk.BooleanVar(value=False)
check_button = tk.Checkbutton(
    gridFrame,
    text="Enable Feature",
    variable=check_var,
    onvalue=True,
    offvalue=False,
    fg="black",
    bg="#fffaf0",
    font=("Arial", 11))
check_button.grid(row=1, column=0, columnspan=3, pady=5)
```

## FRAME 3 — Using PLACE

```
placeFrame = tk.LabelFrame(root, text="Place Geometry", padx=10, pady=10,
bg="#f5f5dc", height=200, width=660)
placeFrame.pack(pady=10, padx=10)
placeFrame.pack_propagate(False)  # keep frame from resizing
```

### Spinbox

```
spin_var = tk.StringVar(value="0")
spinbox = tk.Spinbox(
    placeFrame,
    from_=0,
    to=10,
    increment=1,
    textvariable=spin_var,
    fg="purple",
    bg="lavender",
    font=("Arial", 12),
    width=5,
    relief="ridge",
    bd=3)
spinbox.place(x=30, y=40)
```

### Scale

```
scale_var = tk.DoubleVar(value=50)
scale = tk.Scale(
    placeFrame,
    from_=0,
    to=100,
    orient="horizontal",
    length=200,
    tickinterval=25,
    resolution=5,
    variable=scale_var,
    showvalue=True,
    troughcolor="lightblue",
    sliderlength=20,
    fg="black",
    bg="#f5f5dc",
    font=("Arial", 10),
    relief="sunken",
    bd=2)
scale.place(x=120, y=30)
```

### Label showing live updates from Scale

```
value_label = tk.Label(
    placeFrame,
    text="Scale Value: 50",
    bg="#f5f5dc",
    font=("Arial", 11, "bold"))
value_label.place(x=350, y=50)

def update_value(*args):
    value_label.config(text=f"Scale Value: {int(scale_var.get())}")

scale_var.trace_add("write", update_value)
```

## MAIN LOOP

```
root.mainloop()
```