

07/07/2025

Tkinter

Section 1: Tkinter Basics & Setup.

① What is Tkinter and how it is related to the TK GUI toolkit?

→ It is Python's standard library for creating GUI's.

→ It is a wrapper around the TK GUI toolkit, which is written in 'C' and developed by TCK programming language.

② Why is Tkinter considered a cross-platform GUI toolkit?

→ The underlying TK toolkit runs on Windows, Mac OS, Linux.

→ A Tkinter program written in Python will run unchanged on any supported OS.

③ What are the advantages of using Tkinter for Python GUI development?

→ Built-in

→ Easy to learn

→ Good documentation

→ Fast-prototyping

→ Active development

⑨ What is a Combobox and how is it different from a Listbox

A combobox from tk is a dropdown menu:

→ Allows selection from a drop-down list

→ Optionally editable.

List box

→ Shows all options at once

Combobox

collapses them into a single line

⑩ How do you create a Spinbox for selecting a numeric value and retrieve its value?

→ Spin box lets users pick a number using arrow buttons

spin = tk.Spinbox(root, from=-1, to=10)
Spin.pack()

value = spin.get()
print("selected:", value)

- ⑥ What is a Listbox used for Tkinter?
- Display a scrollable list of items.
 - Allow the user to select one or more items.
 - Item selection menus, file pickers, option lists.

⑦ How do you add a vertical scroll bar to a listbox?

```
listbox = tk.Listbox(root)
scrollbar = tk.Scrollbar(root, orient="vertical",
                        command=listbox.yview)
listbox.config(yscrollcommand=scrollbar.set)
```

```
listbox.pack(side="left", fill="y")
```

```
scrollbar.pack(side="right", fill="y")
```

⑧ How do you get the selected item from a listbox?

```
selected_index = listbox.curselection()
```

```
if selected_index:
```

```
    item = listbox.get(selected_index[0])
```

```
    print("Selected item:", item)
```

③ How do you animate a shape on a canvas using .after():

def move_shape():

Canvas.move(rect, 5, 0) # moves px right

Canvas.after(50, move_shape) # repeat after some time.

rect = Canvas.create_rectangle(10, 50, 60, 100, fill="red")

④ How do you move an object on canvas using .move()?

→ canvas.move(object_id, dx, dy)

↓
id returned
when shape
created

↓
distance to
move horizontally
and vertically

⑤ How do you handle mouse events on a canvas, such as clicking inside a shape?

→ Bind the canvas to mouse events.

→ check if the click is inside the shape using find_closest() or find_overlapping()

④ What method is used to set a fixed window size?
→ Set the size using geometry() and disable resizing.
→ root.resizable(False, False)

⑤ What is the correct format for setting window geometry with position
"Widthxheight+x+y"
root.geometry("500x300+100+100")

Section - 6 : Advanced Widgets .

① What is the purpose of the Canvas widget?
→ Draw graphics
→ Create custom widgets.
→ Build animations , games .
→ Handle mouse based drawing.

② How do you draw a rectangle or circle on a canvas?

→ canvas.create_rectangle(50,50,150,150,fill='blue')
→ canvas.create_oval(100,30,180,110) fill='red')

Section - 5: Window Management & Geometry in Tkinter,

① How do you make a Tkinter window non-resizable?

→ Use the .resizable() method on the root window:

→ root.resizable(False, False)

↓ ↓
Width height.

② What does the geometry() method do in Tkinter?

→ .geometry() method sets the size and position of the main window.

→ root.geometry("widthxheight+x+y")

x

window
dimensions

distance
from the
top-left
corner of
the screen

③ How do you position a window at specific screen coordinates.

→ Use the geometry() method with position.

→ root.geometry("400x300+200+100")

② What is the syntax for binding keyboard or mouse events to a widget?

→ Use the `.bind()` method with an event string and a call back function

→ `widget.bind("<Event Name>", callback)`

③ How do you capture the x and coordinates from a mouse click event?

→ In the event callback, use `event.x` and `event.y`.

`def on_click(event):`

`print("Mouse clicked at:", event.x, event.y)`

`widget.bind("<Button-1>", on_click)`

④ How do you bind the `<key>` event to track key presses?

→ Bind to the Widget(`Text, Entry, root`)

`def on_keypress(event):`

`print("key pressed:", event.char)`

`root.bind("<Key>", on_keypress)`

⑤ What does the `event` parameter represent in a callback function?

→ Parameter is an object automatically passed by Tkinter when an event is triggered.

⑨ How do you add widgets inside a frame?

```
frame = tk.Frame(root)
frame.pack()
label = tk.Label(frame, text="Inside frame")
label.pack()
```

⑩ Can pack() and grid() be used together in the same container?

- No, you should not mix pack() and grid() in the same container.
- We can use an empty frame.
- It can lead to layout conflicts and errors.

Section - 4: Event Handling & User interaction in Tkinter.

⑪ How do you bind a function to a Button click event?

→ Use the command parameter of the Button widget:

```
def say-hello():
```

```
    print("Hello")
```

```
button = tk.Button(root, text="click", command=say-hello)
```

```
button.pack()
```

⑥ What is the role of the `place()` method and when should it be used?

→ `Place()`: gives precise control over widget positioning.

Absolute (x, y)

Relative ($relx, rely$) co-ordinates.

Used when:

- for pixel-perfect positioning
- designing custom layouts or animations.

⑦ How can you center a widget using `place()`?

To center a widget: $\text{center of the parent}$
`widget.place(relx=0.5, rely=0.5, anchor="center")`

Aligns the
widget's centre
to that position

⑧ What is a frame in Tkinter and why is it useful?

→ contains widget used to group other widgets

useful for: splitting the window into sections.

③ What does side = "left" or fill do in the pack() method?

→ Side = "left": places the widget to the left side of the parent.

→ fill = "x": Expands the widget horizontal to fill the available width.

④ When should you prefer grid() over pack()

Use grid():

→ tabular or form-like layouts

→ fine control over positioning.

Use pack():

→ simple vertical or horizontal packing.

⑤ What are row, column and sticky options used for in grid(),

→ row and column:

Define where the widget appears in the grid.

→ sticky: Define how the widget sticks to cell sides:

"n" - top | "e" - right

"s" - bottom | "w" - left

"nsew" - to expand fully

11. What is the difference between
· get(), · insert() and · delete() ?

→ · get() → Retrieves the content

→ · insert() → Inserts Content at a
specific position

→ · delete() → Deletes content from a specific
range or completely.

Section 3: Layouts - pack(), grid(), place()
Frames in Tkinter

Q) What are the three layout managers?

→ Pack() - Simple and automatic layout
based on Side and space.

→ grid() - Table-like layout using row
and columns

→ place() - Absolute positioning using
x, y coordinates of relative
positions.

② Explain how the pack() geomet
manager works.

→ Pack() organizes widgets in blocks
before or after each other.

→ It adds widgets relative to par
widget's sides.

How do you insert and retrieve text from a Text widget?

Insert:

→ text_widget.insert("1.0", "Hello\\n Multi")

Retrieve:

→ Content = text_widget.get("1.0", "end-1c")

Which method is used to delete all text from an Entry or Text widget?

For Entry:

entry.delete(0, tk.END)

For Text:

text_widget.delete("1.0", tk.END)

How do you configure a Button to execute a function on click?

```
def say_hello():
```

```
    print("Hello")
```

```
button = tk.Button(root, text="click", command=say_hello)
```

```
button.pack()
```

④ How do you update the text of a label dynamically?
→ Use .config() or .configure():
→ label.config(text = "updated Text")

⑤ What is the use of the Entry Widget and how do you retrieve its value?
→ Entry Widget is used for single-line user input.
→ To retrieve the values
entry = tk.Entry(root)
entry.pack()
value = entry.get()

⑥ How do you validate an email?
import re
def validate_email():
 email = entry.get()
 pattern = r'^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}\$'
 if re.match(pattern, email):
 print("Valid Email")
 else:
 print("Invalid Email")

Section 2: Widgets - Label, Entry, Text, Button in Tkinter.

Q1) What is a Widget in Tkinter?

→ It is a GUI element such as button, Label, tex box or frame.

→ A visual component that allows users to interact with application.

Q2) Name any five commonly used widgets:

1.) Label - to display text or images.

2.) Entry - to enter a single line text.

3.) Text - to enter or display multi-line text.

4.) Button - to triggers action or commands.

5.) List box - to display a list of Selectable item

Q3) How do you create and display a Label widget?

```
label = tk.Label(root, text="Hi")  
label.pack()
```

7) What is the purpose of root.mainloop() in a Tkinter program?

- Starts the event loop of the GUI application
- Keeps the window responsive and open.
- Waits for user interactions

8) How do you set the window size in Tkinter?

- Use the .geometry() method
- root.geometry("500x300")
Width x height.

9) How do you change the title of a Tkinter window?

- Use the .title() method.
- root.title("My App")

10) How do you add an icon to a Tkinter window?

- Use .iconbitmap() for .ico files (Windows)
- .iconphoto() for cross-platform.
- root.iconbitmap("myicon.ico")

④ How do you check if Tkinter is installed in your system?

```
import tkinter  
print("Tkinter is installed")  
(OR)
```

```
python -m tkinter
```

Runs without any error.

⑤ What command is used to install Tkinter on Ubuntu/Debian Systems?

```
Sudo apt-get install python3-tk
```

⑥ What does the tk.Tk() function do in a Tkinter application?

→ Creates the main application window.

→ Serves as the root container for all other GUI widgets

→ It must be created before adding widgets.