

## Tkinter

1) what is Tkinter & how is related to the TK GUI toolkit?

Tkinter is a standard python library for creating GUIs. It acts as a wrapper around the TK GUI toolkit, allowing python developers to build windows, dialogs, buttons, menus & other GUI elements using Tk's functionality through python syntax.

2) why Tkinter considered a cross - platform GUI Toolkit?

Tkinter works on multiple operating systems such as windows, macOS & Linux. The same python Tkinter code can run on these platforms without modification, making it is a convenient choice for building portable GUI applications.

3) what are advantages of using Tkinter for python GUI Development?

- \* Tkinter is built into python, so no extra installation is needed. It easy to use, lightweight well-documented, and supports cross platform development.

- \* It is ideal for small to medium-sized GUI applications.

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

To run this command:

```
import tkinter
```

If there is no error, Tkinter is installed.

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

```
sudo apt-get install python3-tk
```

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

tk.Tk() creates the main application window.

It initializes the Tkinter framework & serves as the root window where all GUI elements are placed.

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

root.mainloop() starts the Tkinter event loop, which waits for user interaction & keeps the window open. It ensures the program runs & responds to events like button clicks.

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

```
root =
```

```
root.geometry("300x300")
```

! !  
width height  
Pixels pixels

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

`tk root.title("Tkinter window")`

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

`root.iconbitmap("myicon.ico")`

11) what is widgets in Tkinter? Name five commonly used widgets?

A widget in Tkinter is a GUI elements that performs a specific functions, such as displaying text or collecting user input.

\* label

\* button

\* entry

\* Text

\* checkbutton

12) How do you create and display a Label widget in a Tkinter window.

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

`label.pack()`

13) How do you update the text of a Label dynamically?

`label.config(text = "updated text")`

14) what is the use of the entry widget & how do you retrieve its value?

The entry widget allows user to enter a single line of text. Retrieve its value with .get().

Value = entry.get()

15) How do you validate an email address entered into an entry widget using regex?

```
import re
```

```
email = entry.get()
```

```
pattern = r"^[a-zA-Z0-9]+@[a-zA-Z]+\.[a-zA-Z]+"
```

```
if re.match(pattern, email):
```

```
    print("valid email")
```

```
else:
```

```
    print("invalid email")
```

16) what is the purpose of text widget & how it's differ from entry widget?

The Text widget is used for multi-line text input, whereas entry supports only a single line. Text is suitable for larger inputs like comments or messages.

14) How do you insert & retrieve text from a text widget?

Insert text:

```
text.insert("1.0", "Hello")
```

Retrieve text:

```
Content = text.get("1.0", tk.END)
```

18) Which method is used to delete all text from an entry or Text widget?

For entry:

```
entry.delete(0, tk.END)
```

For Text:

```
text.delete("1.0", tk.END)
```

19) How do you configure a button to execute function on click?

Use the command parameter.

E.g.

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

command = my-function)

20) What is the difference b/w .get(), .insert() & .delete() methods?

\* .get() → retrieves current content of the widget.

\* .insert() → adds content at a specified position.

• delete() → Removes content b/w specified position.

Q1) what are the three layout (geometry) managers in Tkinter?

\* pack()

\* grid()

\* place()

Q2) Explain how the pack() geometry manager works.

pack() organizes widgets in blocks before placing parent widget. It places widgets relative to each other (top, bottom, left, right) & automatically resizes them in specified.

Q3) what does \*side = "left" or fill = "x" do in the pack method?

\* side = "left" places the widgets to the left side of the parent container.

\* fill = "x" makes the widget expand horizontally to fill the available spaces.

Q4) when should you prefer grid() over pack()?

use grid() when need to place widgets in a structured, tabular layout using rows & columns. Its ideal for form or complex arrangement where precise placement is needed.

Q5) what are row, column & sticky positions used in grid()?

row & column → specify the grid position of the widget.

sticky → controls how the widget expands within the cell.

Q6) what is the role of place() method & when should it be used?

place() allows precise positioning of widget using absolute or relative coordinates. It's used when you need full control over the widget location, such as custom-designed layouts.

Q7) How can you center a widget using place()?

use relx=0.5, rely=0.5, anchor="center"

Q8) what is a frame in Tkinter & why is it useful?

A Frame is a container widget used to group other widgets. It helps organize layouts, especially when combining different geometry managers, or separating logical sections of the UI.

Q9) How do you add widgets inside a frame?

Create a Frame, then create pack (or grid or place) widgets inside it like this:

Eg. `frame = Tk.Frame(root)`

`frame.pack()`

`label = Tk.Label(frame, text = "frame e.g.")`

`label.pack()`

Q10) Can pack() and grid() be used together in the same container?

No, pack() & grid() cannot be used in the same container. Mixing them will result in Tcl Errors.

Q11) How do you bind a function to a button click event?

Bind a function in two ways.

1. using command:

`btn = Tk.Button(root, text = "click", command = "my_fn")`

2. using bind() method:

`button.bind("<Button-1>", my_fn)`

32) what is the syntax for binding keyboard or mouse events to a widget?

widget.bind ("", key\_handler) → key press

widget.bind ("", click\_handler)

↳ Left Mouse

33) How do you capture the x & y coordinates from a mouse click event?

```
def click_handler(event):
```

```
    print("x:", event.x, "y:", event.y)
```

34) How do you bind the <key> event to track key presses?

```
root.bind ("", key_handler)
```

```
def key_handler(event):
```

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

35) what does the event parameter represent in a callback function?

The event parameter is an object that contains information about the event that triggered the callback. It includes attributes like event.x, event.y, event.char, event.keysym, and more.

36) How do you make a window non-resizable?

use resizable() method.

root.resizable(False, False)

37) what does the geometry() method do in Tkinter?

It is used to sets the size & optionally the position of the window.

E.g. root.geometry("300x300")

38) How do you position a window at specific screen coordinates?

Add +x+y to the geometry.

E.g. root.geometry("400x250+100+150")

39) what method is used to set a fixed window?

use both geometry & resizable(),

E.g. root.geometry("300x300") → set size

root.resizable(False, False) → lock size

40) what is the correct format for setting window geometry with position (e.g. "500x300+100+100")?

Correct format is:

"<width>x<height>+<x-offset>+<y-offset>"

41) What is the purpose of the canvas widgets in Tkinter?

Canvas widget is used for drawing shapes, displaying images, or creating custom graphics. It's ideal for games, charts or other visual requiring pixel-level control.

42) How do you draw a rectangle or circle on a canvas?

Rectangle:

canvas.create\_rectangle(50, 50, 150, 100)

Circle (Oval):

canvas.create\_oval(50, 50, 100, 100)

43) How can you animate a shape on a canvas using .after()?

Use .after() to schedule repeated movement:

def move\_shape():

    canvas.move(shape\_id, 5, 0)

    canvas.after(50, move\_shape)

move\_shape()

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

canvas.move(object\_id, dx, dy)

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

Bind a click event to the canvas.

Check coordinates.

```
Canvas.bind("<Button-1>", handle_click)
```

```
def handle_click(event):
```

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

46) What is the listbox used for in Tkinter?

A listbox displays a list of selected items. User can choose one or multiple items from the list.

47) How do you add vertical scrollbar to a listbox?

scrollbar

```
Scrollbar = tk.Scrollbar(root)
```

```
listbox = tk.Listbox(root, yscrollcommand=Scrollbar.set)
```

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

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

```
listbox.pack(side="left", fill="both", expand=True)
```

48) How do get the selected item from listbox?

```
selection = listbox.get(listbox.curselection())
```

49) what is combobox & how is it different from a listbox?

A combobox (Tk) is a drop down list that lets user choose one item. Unlike, listbox, it saves space & allows optional text entry.

50) How do you create a Spinbox for selecting a numeric value & retrieve its value?

Spin = Tk.Spinbox (root, from\_=0, to=100)

Value = spin.get() → Retrieve value.