



Sliders In Tkinter Using Scale() | Python Tkinter GUI Tutorial In Hindi #19



[Overview](#) [Q&A](#) [Downloads](#) [Announcements](#)

Sliders In Tkinter Using Scale() | Python Tkinter GUI Tutorial In Hindi #19

The **Scale()** widget provides a graphical slider object that allows the user to select numerical values by moving the “slider” knob along the specific scale. Users can control the minimum and maximum values as well as the resolution. In Slider() widget, we use some new attributes, i.e., *to*, *from*, *orient*, etc.

Attributes:

- **from_:** This should be a float or integer value that defines one end or the starting value of the scale's range.
- **to:** This should also be a float or integer value that defines the other end or the finishing value of the scale's range. **Note: The to value can be either greater than or less than the from_ value. For vertical scales, the to value defines the bottom of the scale; for horizontal scales, the right end.**
- **orient:** It sets the orientation of the scale. If we set orient=HORIZONTAL, the scale runs along the x dimension, or we set orient=VERTICAL, it runs parallel to the y-axis. **Default is horizontal.**
- **length:** It sets the length of the Scale() widget. It defines the x dimension if the scale is horizontal or the y dimension if vertical. **The default value is 100 pixels.**
- **resolution:** Normally, the user is only able to change the scale in whole units. But this attribute sets this option to some other value to change the smallest increment of the scale's value. For example, if from_=-10 and to=10, and you set resolution=5, the scale will have 5 possible values: -10, -5, 0, +5, and +10.

- **tickinterval:** It displays periodic scale values. If we set this option to a number, ticks will be displayed on multiples of that value. For example, if from_=0, to=10, and tickinterval=2, labels will be displayed along the scale at values 0, 2, 4, 6, 8, 10. These labels appear below the scale if horizontal, to its left if vertical. **Default is 0, which suppresses the display of ticks.**

Methods:

- **get():** This method returns the current value of the scale.
- **set(value):** It sets the scale's value. For example, if we give set(30) the initial scale value will show 30 (the scale will start from 30).

Code is described below:

```
from tkinter import *
import tkinter.messagebox as tmsg

root = Tk()
root.geometry("455x233")
root.title("Slider tutorial")

def getdollar():
    print(f"We have credited {myslider2.get()} dollars to your bank account")
    tmsg.showinfo("Amount Credited!", f"We have credited {myslider2.get()} dollars to your bank account")

# myslider = Scale(root, from_=0, to=100)
# myslider.pack()
Label(root, text="How many dollars do you want?").pack()
myslider2 = Scale(root, from_=0, to=100, orient=HORIZONTAL, tickinterval=50)
# myslider2.set(34)
myslider2.pack()

Button(root, text="Get dollars!", pady=10, command=getdollar).pack()

root.mainloop()
```

- Importing *tkinter* is the same as importing any other module in the Python code. Note that the module's name in Python 2.x is '*Tkinter*', and

in Python 3.x it is '*tkinter*'.

```
from tkinter import *
```

- Tkinter-MessageBox (tkinter.messagebox) is imported as '*tmsg*'.

```
import tkinter.messagebox as tmsg
```

- To create the main window, Tkinter offers a method 'Tk'. To change the name of the window, you can change the className to the desired one.

```
root = Tk()
```

- To set the dimensions of the Tkinter window and to set the position of the main window on the user's desktop, the *geometry()* function is used. As in the example: the width is 455 pixels, and height is 233 pixels, so we can write the function as *geometry(455x233)*.

```
root.geometry("455x233")
```

- The title of the GUI window is set using the *title()* function. Here we have taken the title as "Slider tutorial".

```
root.title("Slider tutorial")
```

- A function "getdollar" is defined. Within this function, the print statement is written from where the user will get the myslider2 value (myslider2 is discussed below). A message box named "Amount Credited!" is also created, and within that, also *get()* method is written so that the user can see the myslider2 value in the message box.

```
def getdollar():  
    print(f"We have credited {myslider2.get()} dollars to your bank account")  
    tmsg.showinfo("Amount Credited!", f"We have credited {myslider2.get()} dollars to your bank account")
```

- A label is taken, and text is passed as an attribute and then the label is packed using the *pack()* method.

```
Label(root, text="How many dollars do you want?").pack()
```

- Slider is created using Scale() widget. Its beginning range is set to 0 using **from_** attribute and its ending range is set to 100 using **to** attribute. The orientation is taken as **HORIZONTAL** using the **orient** attribute. Also, **tickinterval** attribute is taken as 50 so that the scale only shows the multiple values of 50. At last, the slider is packed using the pack() method.

```
myslider2 = Scale(root, from_=0, to=100, orient=HORIZONTAL, tickinterval=50)  
myslider2.pack()
```

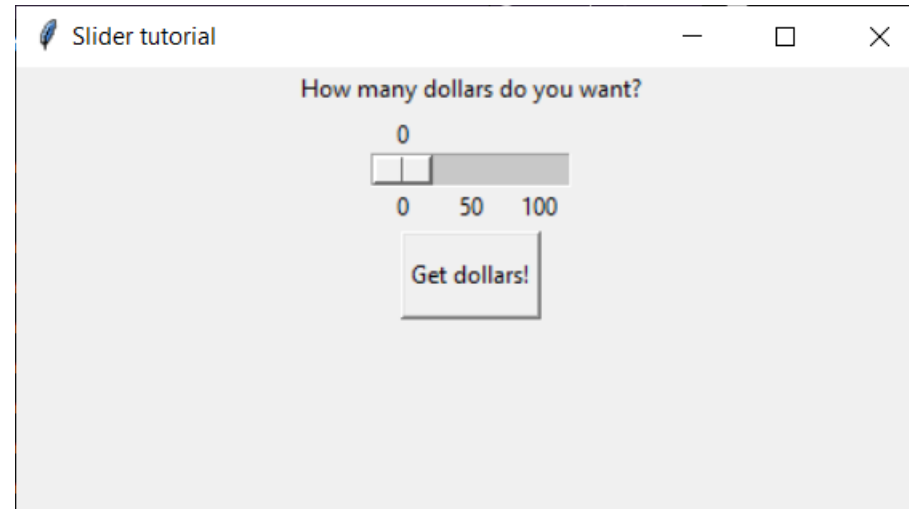
- A button is created using the Button() widget and getdollars() function is called from this button using the **command** attribute. The button is packed using pack() method.

```
Button(root, text="Get dollars!", pady=10, command=getdollar).pack()
```

- There is a method known by the name *mainloop()* which is used when your application is ready to run. This is an infinite loop used to run the application, wait for an event to occur and process the event as long as the window is not closed.

```
root.mainloop()
```

Output: The output of the code (or the GUI window) is given below:





Code as described/written in the video

```
from tkinter import *
import tkinter.messagebox as tmsg

root = Tk()
root.geometry("455x233")
root.title("Slider tutorial")

def getdollar():
    print(f"We have credited {myslider2.get()} dollars to your bank account")
    tmsg.showinfo("Amount Credited!", f"We have credited {myslider2.get()} dollars to your bank account")

# myslider = Scale(root, from_=0, to=100)
# myslider.pack()
Label(root, text="How many dollars do you want?").pack()
myslider2 = Scale(root, from_=0, to=100, orient=HORIZONTAL, tickinterval=50)
# myslider2.set(34)
myslider2.pack()

Button(root, text="Get dollars!", pady=10, command=getdollar).pack()

root.mainloop()
```

[Previous](#)[Next](#)

CodeWithHarry

Copyright © 2022 CodeWithHarry.com

