



## Status Bar In Tkinter | Python Tkinter GUI Tutorial In Hindi #24



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

## Status Bar In Tkinter | Python Tkinter GUI Tutorial In Hindi #24

A **status bar** is usually a narrow bar at the bottom of the GUI to indicate extra information like the file's word counts or anything that could add extra value to the user interface. Tkinter doesn't have a dedicated status bar widget but uses Label **L** widget with appropriate configuration to work as the status bar in the GUI.

**Code is described below:**

```
from tkinter import *

def upload():
    statusvar.set("Busy..")
    sbar.update()
    import time
    time.sleep(2)
    statusvar.set("Ready")

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

```
root.title("Status bar tutorial")

statusvar = StringVar()
statusvar.set("Ready")
sbar = Label(root, textvariable=statusvar, relief=SUNKEN, anchor="w")
sbar.pack(side=BOTTOM, fill=X)
Button(root, text="Upload", command=upload).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 *
```

- To define a function 'def' (i.e. here the function `upload()` is defined) is used. Within this function "statusvar" variable (`StringVar()`) is set to "Busy.." and the `update()` function is used so that it updates the string "Busy.." on the `Label ()` widget. `Time` module is imported, and `time.sleep(2)` is used to wait 2 sec. In the previous state ("Busy..") and then it sets to "Ready" state.

```
def upload():
    statusvar.set("Busy..")
    sbar.update()
    import time
    time.sleep(2)
    statusvar.set("Ready")
```

- 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 created using the “title()” function. Here the title is “Status bar tutorial”.

```
root.title("Status bar tutorial")
```

- A StringVar() is taken as “statusvar” (a string variable), and it is set to “Ready” initially.

```
statusvar = StringVar()  
statusvar.set("Ready")
```

- A Label() is created (this will actually show as a status bar), and “statusvar” (StringVar()) is taken as “textvariable” where the text is shown after each update. The attribute “relief” determines how the Label appears. We prefer the Label to appear “SUNKEN” so that the status bar looks seamlessly one part of the window. The alignment of the text inside the Label is set to “w” (west) using the “anchor” attribute.
- The Label() variable “sbar” is packed at the BOTTOM of the window, and the Label is filled along X dimension.

```
sbar = Label(root, textvariable=statusvar, relief=SUNKEN, anchor="w")  
sbar.pack(side=BOTTOM, fill=X)
```

- We extend our little script by the button “Upload”. We bind the function upload() to the *Upload* button. So, every time this button is clicked, it will call the upload() function. Then we need to use the pack() method to pack the button.

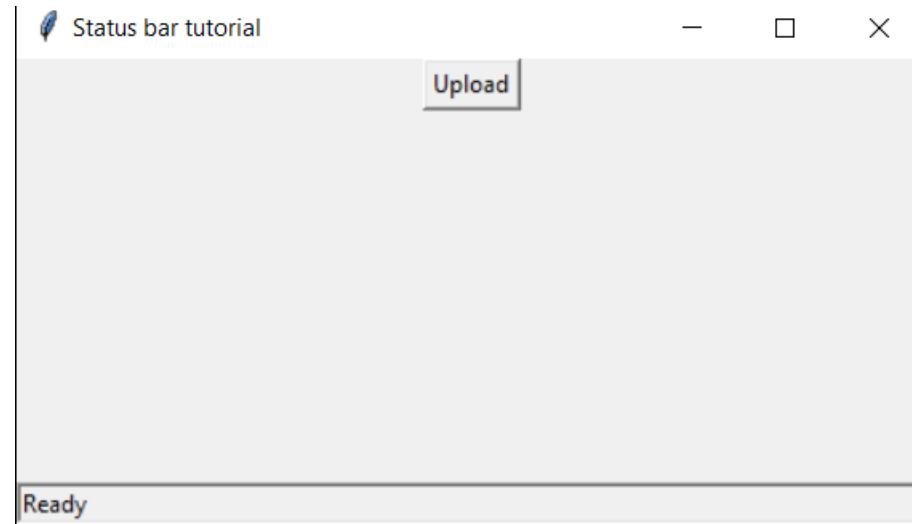
```
Button(root, text="Upload", command=upload).pack()
```

- 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:

Copy

```
from tkinter import *

def upload():
    statusvar.set("Busy..")
    sbar.update()
    import time
    time.sleep(2)
    statusvar.set("Ready")

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

statusvar = StringVar()
statusvar.set("Ready")
sbar = Label(root, textvariable=statusvar, relief=SUNKEN, anchor="w")
```

```
sbar.pack(side=BOTTOM, fill=X)
Button(root, text="Upload", command=upload).pack()
root.mainloop()
```

[Previous](#)[Next](#)**CodeWithHarry**

Copyright © 2022 CodeWithHarry.com

