CodeWithHarry

HTML CSS JS C C++ JAVA PYTHON PHP

Handling Events In Tkinter GUI | Python Tkinter GUI Tutorial In Hindi #14

Overview Q&A Downloads Announcements

Handling Events In Tkinter GUI | Python Tkinter GUI Tutorial In Hindi #14

Tkinter is a GUI (Graphical User Interface) module that is widely used in desktop applications. It provides a variety of Widget classes and functions, with the help of which one can make our GUI more attractive and user-friendly in terms of both looks and functionality. The binding function is used to deal with the events. We can bind **Python's Functions** and methods to an event and bind these functions to any particular widget. Events can come from various sources, including key presses and mouse operations by the user and redraw events from the window manager (indirectly caused by the user, in many cases).
The most important part of an event specifier is the *type* field. It specifies the kind of event that we wish to bind and can be user actions like **Button** and **Key**, or window managers events like **Enter**, **Configure**, and others.

Event formats:

- **<Button-1>:** A mouse button is pressed over the widget. Button 1 is the leftmost button, Button 2 is the middle button (where available), and Button 3 is the rightmost button.
- **<B1-Motion>:** The mouse is moved, with mouse button 1 being held down (use B2 for the middle button, B3 for the right button).
- **<Double-Button-1>:** Button 1 was double-clicked. You can use **Double** or **Triple** as prefixes. **Note: Both bindings will be called if you bind to both a single click (<Button-1>) and a double click.**
- **<Enter>:** The mouse pointer entered the widget. **Note: This event doesn't mean that the user pressed the Enter key.**

Event attributes:

As Event attributes, we can use **width, height, char, num, widget,** etc.

Code is described below:

```
from tkinter import *

def harry(event):
    print(f"You clicked on the button at {event.x}, {event.y}")

root = Tk()
root.title("Events in Tkinter")
root.geometry("644x334")

widget = Button(root, text='Click me please')
widget.pack()

widget.bind('<Button-1>', harry)
widget.bind('<Double-1>', quit)
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 harry() is defined) is used, and within def, the event is called.
- It prints the X and Y coordinates of the event when the event has occurred (i.e., <Button-1>,<Double-1>).

```
def harry(event):
    print(f"You clicked on the button at {event.x}, {event.y}")
```

- 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 title of the GUI window title() function is used.

```
root.title("Events in Tkinter")
```

- 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 644 pixels, and height is 334 pixels so that we can write the function as *geometry(644x334).*

```
root.geometry("644x334")
```

- We extend our little script by the button "Click me please," passing the attribute "text" within the Button widget, and then the button is packed using the pack() method.

```
widget = Button(root, text='Click me please')
widget.pack()
```

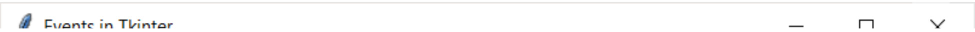
- The button "widget" is bound using the bind method, and we can pass any attribute (i.e., <Button-1>,<Double-1>, etc.). Whenever the button is clicked, the functions (i.e., harry) will be called, and the event will occur.

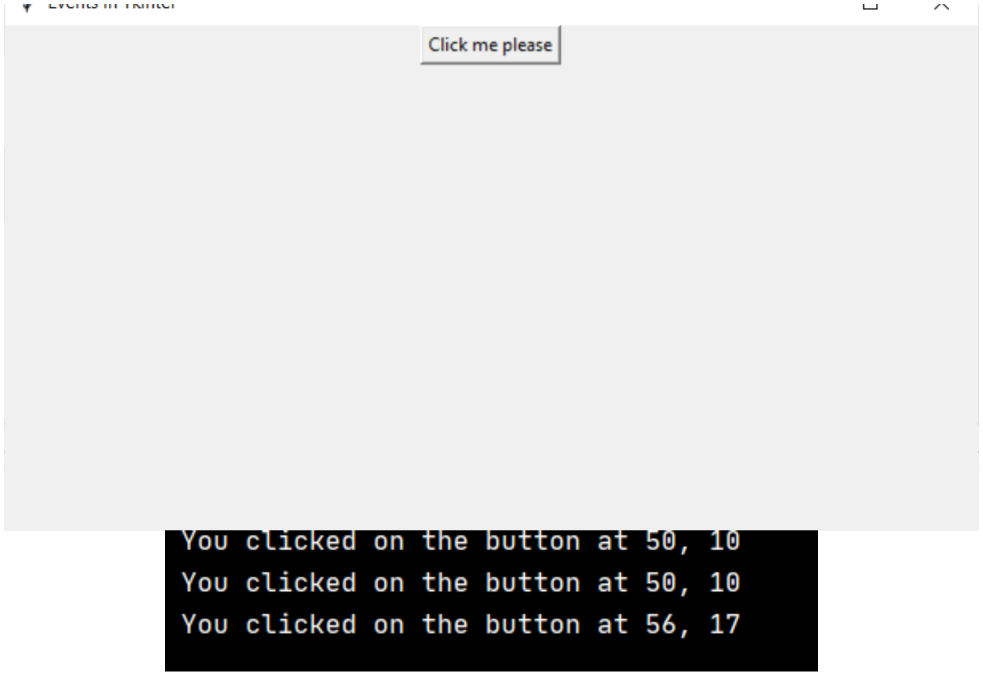
```
widget.bind('<Button-1>', harry)
widget.bind('<Double-1>', quit)
```

- 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. In this example, when the button is clicked once, it will call harry, and it will print the coordinates of x and y of the place where the button was clicked, and if the button is clicked twice, the program will quit.

```
root.mainloop()
```

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





```
You clicked on the button at 50, 10
You clicked on the button at 50, 10
You clicked on the button at 56, 17
```

Code as described/written in the video

Copy

```
from tkinter import *

def harry(event):
    print(f"You clicked on the button at {event.x}, {event.y}")

root = Tk()
root.title("Events in Tkinter")
root.geometry("644x334")

widget = Button(root, text='Click me please')
widget.pack()
widget.bind('', harry)
widget.bind('', quit)

root.mainloop()
```

Previous

Next