Entry Widget & Grid Layout In Tkinter | Python Tkinter GUI Tutorial In Hindi #10

- **Entry() widgets** are the basic widgets of Tkinter, which is used to get input, i.e., text strings, from the user of an application. This widget allows the user to enter a single line of text.

Note: **If the user enters a string, which is longer than the available display space of the widget, the content will be scrolled.** This means that the string cannot be seen in its entirety. The arrow keys can be used to move to the invisible parts of the string. **If you want to enter multiple lines of text, you have to use the text widget. An entry widget is also limited to a single font.**

- The **Grid()** manager is the most flexible of the geometry managers in Tkinter. The Grid() geometry manager puts the widgets in a 2-dimensional table. The master widget is split into a number of rows and columns, and each "cell" in the resulting table can hold a widget.

Note: **We can also create layouts using Pack() manager,** but it takes a number of extra frame widgets and a lot of work to make things look good. If you use the grid manager instead, you only need one call per widget to get everything laid out properly.

Code is described below:

```
from tkinter import *

def getvals():
    print(f"The value of username is {uservalue.get()}")
    print(f"The value of password is {passvalue.get()}")


root = Tk()
root.geometry("655x333")

user = Label(root, text="Username")
password = Label(root, text="Password")
user.grid()
password.grid(row=1)

# Variable classes in tkinter
# BooleanVar, DoubleVar, IntVar, StringVar

uservalue = StringVar()
passvalue = StringVar()
```

```
userentry = Entry(root, textvariable = uservalue)
passentry = Entry(root, textvariable = passvalue)

userentry.grid(row=0, column=1)
passentry.grid(row=1, column=1)

Button(text="Submit", command=getvals).grid()

root.mainloop()
```

- Importing *tkinter* is the same as importing any other module in the Python code. Note that the name of the module 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 getvals() is defined) is used and use the get() method on the Entry objects (i.e. uservalue and passvalue).

```
def getvals():
    print(f"The value of username is {uservalue.get()}")
    print(f"The value of password is {passvalue.get()}")
```

- 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 655 pixels and height is 333 pixels, so we can write the function as *geometry(655x333).*

```
root.geometry("655x333")
```

- To call the Label widget, which is a child of the root widget. The keyword parameter "text" specifies the text "Username" for the variable *user* and "Password" for the variable *password* to be shown:

```
user = Label(root, text="Username")
password = Label(root, text="Password")
```

- To use the grid() method for placing the labels in the proper place by passing attributes row and column. If we don't pass anything in the grid() method, the default is 0 for both row and column (Here, for the *user,* we pass nothing in the grid() so row=0 and column=0, and for the *password,* we pass row=1 so row=1 and column=0).

```
user.grid()
password.grid(row=1)
```

- To take the variables and set them to the variable classes (i.e., StringVar()). The attribute *textvariable* of Entry() widget is used to allow the user to enter some text.

```
uservalue = StringVar()
```

```
passvalue = StringVar()

userentry = Entry(root, textvariable = uservalue)
passentry = Entry(root, textvariable = passvalue)
```

- To use the grid() method for placing the Entry() widgets in the proper place by passing attributes row and column. If we don't pass anything in the grid() method, the default is 0 for both row and column (Here, for *userentry* we pass row=0 and column=1, and for *passentry* we pass row=1 and column=1 in the grid()).

```
userentry.grid(row=0, column=1)
passentry.grid(row=1, column=1)
```

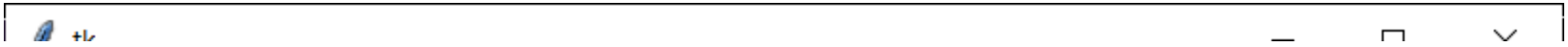
- We extend our little script by two buttons, "Submit". We bind the function getvals(), using the get() method on the Entry objects, to the *Submit* button. So, every time this button is clicked, the content of the Entry fields will be printed on the terminal from which we had called the script.

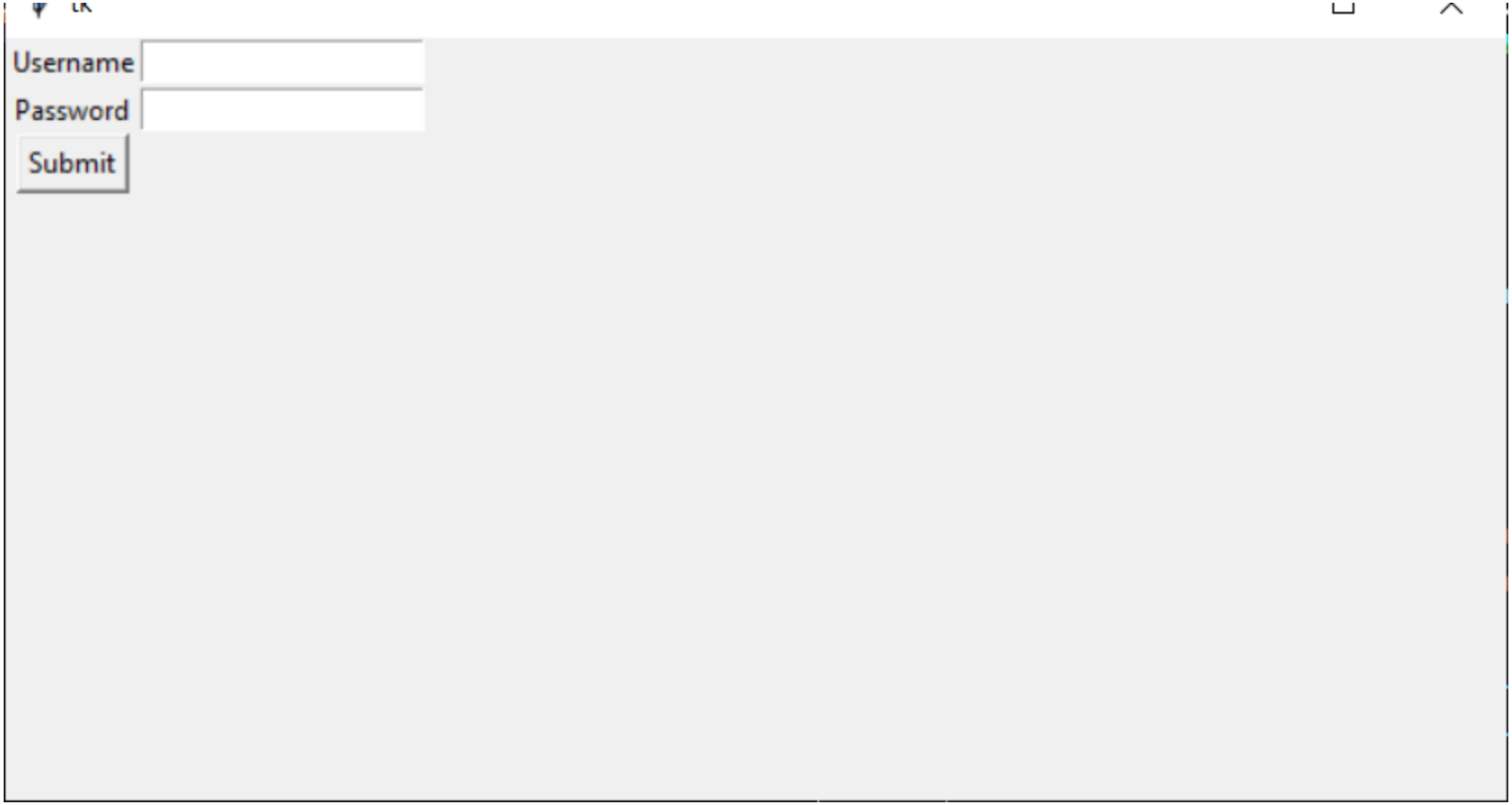
```
Button(text="Submit", command=getvals).grid()
```

- 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 *

def getvals():
    print(f"The value of username is {uservalue.get()}")
    print(f"The value of password is {passvalue.get()}")
```

```
root = Tk()
root.geometry("655x333")

user = Label(root, text="Username")
password = Label(root, text="Password")
user.grid()
password.grid(row=1)

# Variable classes in tkinter
# BooleanVar, DoubleVar, IntVar, StringVar

uservalue = StringVar()
passvalue = StringVar()

userentry = Entry(root, textvariable = uservalue)
passentry = Entry(root, textvariable = passvalue)

userentry.grid(row=0, column=1)
passentry.grid(row=1, column=1)

Button(text="Submit", command=getvals).grid()

root.mainloop()
```

Previous Next

CodeWithHarry