**Objective**

* The objective of the lesson is to introduce students to use the Entry widget and process input from the user [MUST SAY]

**Lesson Description**

* In this node, learners will be working with Entry widget that will help them get input from the user in order to process text from the user.
* Student Link: <https://repl.it/@coding2/PYT2C3G1Node3#main.py>

**At the start**

**Before the Tasks**

* *Let’s discuss all the widgets we have learnt so far.*[For a quick recap, we have learnt about Label and button widget so far]
* *We use the ‘command’ function to assign a function to the button.*
* *We use the ‘configure’ function to print an output in the label.*

**Task 1**

**Task 1: Quiz Me**

* *We were using the input function to get any data from the user.*
* *In Tkinter we will use the ‘Entry’ widget to get input from the user*
* *‘Entry’ widget creates a textbox on the screen that allows us to enter data into it.*txt=Entry(window, width=10)
* *Here the txt is a variable that holds the data, Entry is the name of the widget.*
* *Widget will appear in our window with a width of 10 units*
* *Now let’s push the widget to the grid*txt.grid(column=0, row=2)
* *Great, we are pushing it to the second row, so that we can add widgets before the text box*
* *Now let’s run the code and observe the output*[Run the code and ask the learner if he/she can see a text box or not.  
  Ask them to type something in it]
* *Awesome!*

**Task 1: Solution**

from tkinter import \*

window = Tk()

window.title("Hello world")

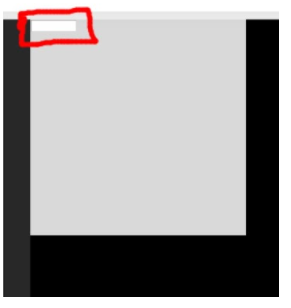
txt=Entry(window, width=10)

txt.grid(column=0, row=2)

window.geometry('400x400')

window.mainloop()

**Output Displayed:**

****

**ENTRY WIDGET:**

* The Entry Widget is used to enter or display a single line of text.
* **Syntax :** entry = tk.Entry(parent, options)
* **Parameters:**

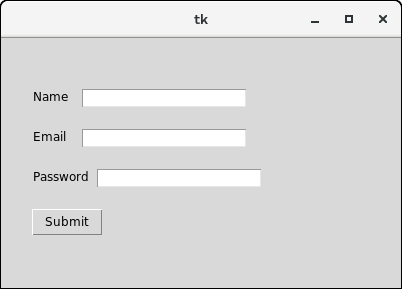
**1) Parent:** The Parent window or frame in which the widget to display.

**2) Options:** There are various options that can be passed along with the entry widget.

* **Methods:**

1. **get() :** Returns the entry’s current text as a string.
2. **delete() :** Deletes characters from the widget.
3. **insert ( index, ‘name’) :** Inserts string ‘name’ before the character at the given index.

<https://replit.com/@udhayakumar311/Session-34-Entryex#main.py>



**Task 2**

**Task 2: Adding more Widgets**

* *Now let’s add a ‘Label’ widget just above the ‘Entry’ widget.*
* *This will be like an instruction for the user to enter something in the textbox.*
* *Let’s set the text of the label as “Enter your name”*lbl=Label(text="Enter your name")
* *Now let’s push it to the grid*lbl.grid(column=0,row=1)
* *Nice. Now let’s run the program and observe the output.*

**Task 2: Solution**

from tkinter import \*

window = Tk()

window.title("Hello world")

lbl=Label(text="Enter your name")

lbl.grid(column=0,row=1)

txt=Entry(window, width=10)

txt.grid(column=0, row=2)

window.geometry('400x400')

window.mainloop()

**Output Displayed:**

****

**Task 3**

**Task 3: Let’s print it out**

* *Let’s create a function called as clicked that will print the message with the name.*
* *def clicked():*
* *This function will print a message with the name*
* *Now let’s add a configure statement and print the message with the name*def clicked():  
   lbl.configure(text="Happy coding,"+txt.get())  
  *Here txt is the variable that will store the input from the user and get() function will get the input from the user.*
* *Now let’s add a button that will print your name and a message below it.*
* *Also add a command of the ‘clicked’ function to it.*btn=Button(text="Click me", command=clicked)  
  btn.grid(column=0, row=3)
* *Great, now let’s run the program and observe the output*

**Task 3: Solution**

from tkinter import \*

window = Tk()

window.title("Hello world")

lbl=Label(text="Enter your name")

lbl.grid(column=0,row=1)

txt=Entry(window, width=10)

txt.grid(column=0, row=2)

def clicked():

lbl.configure(text="Happy coding,"+txt.get())

btn=Button(text="Click me", command=clicked)

btn.grid(column=0, row=3)

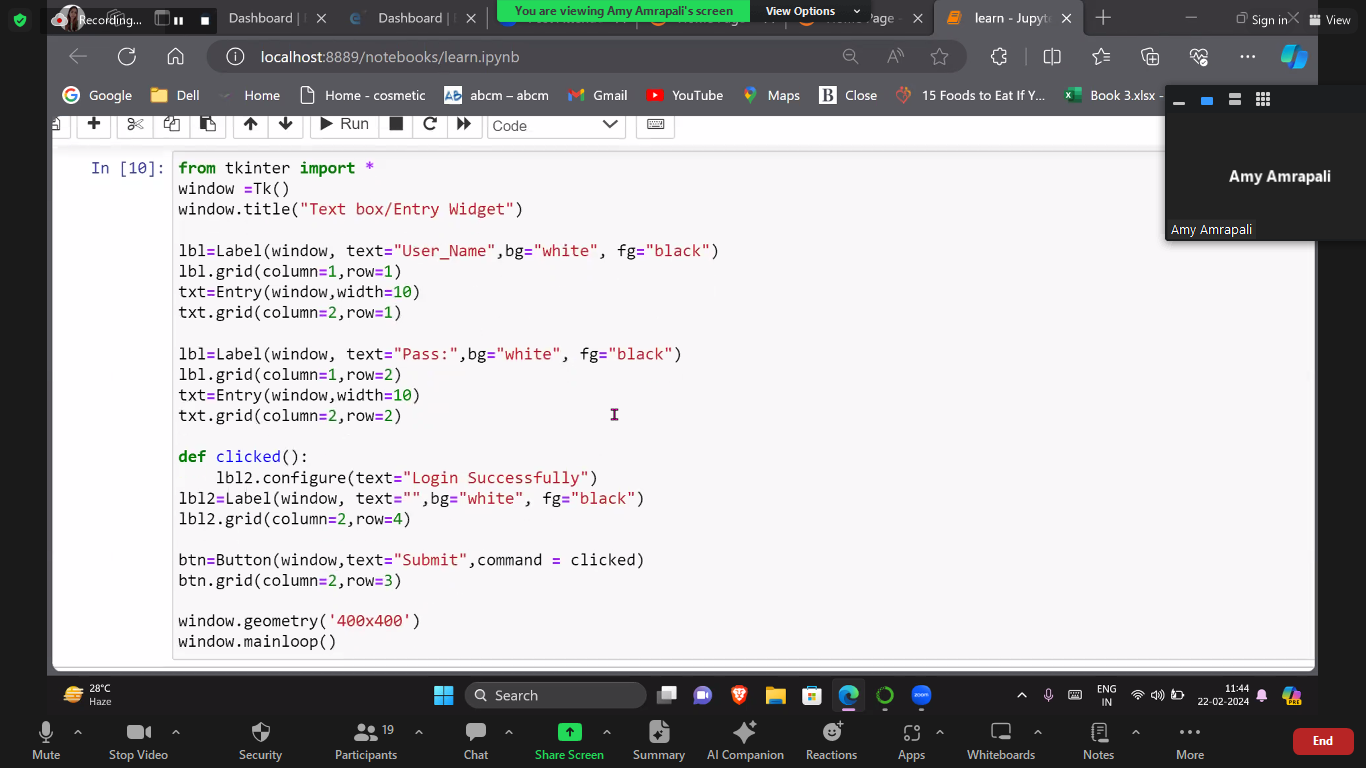
window.geometry('400x400')

window.mainloop()

**Output Display:**

****

**Login page code:**

****

**In the end**

* *You learnt to use the entry widget, get input from the user and work with variables to store/print them*.

Calculator:<https://replit.com/@NikankshaJain/additin-using-tkinter-17#main.py>

from tkinter import \*

window = Tk()

window.title("Login page")

lbl=Label(window,text="Enter first number",bg="blue",fg="white")

lbl.grid(column=1,row=1)

txt=Entry(window)

txt.grid(column=3, row=1)

lbl2=Label(window,text="Enter second number",bg="blue",fg="white")

lbl2.grid(column=1,row=2)

txt1=Entry(window)

txt1.grid(column=3, row=2)

def add():

a=int(txt.get())

b=int(txt1.get())

ans=a+b

lbl1.configure(text=ans)

btn=Button(window,text="+",bg="blue",fg="white",command =add)

btn.grid(column=2,row=3)

lbl1=Label(window,text=" ",bg="red",fg="white")

lbl1.grid(column=1,row=4)

window.geometry('400x400')

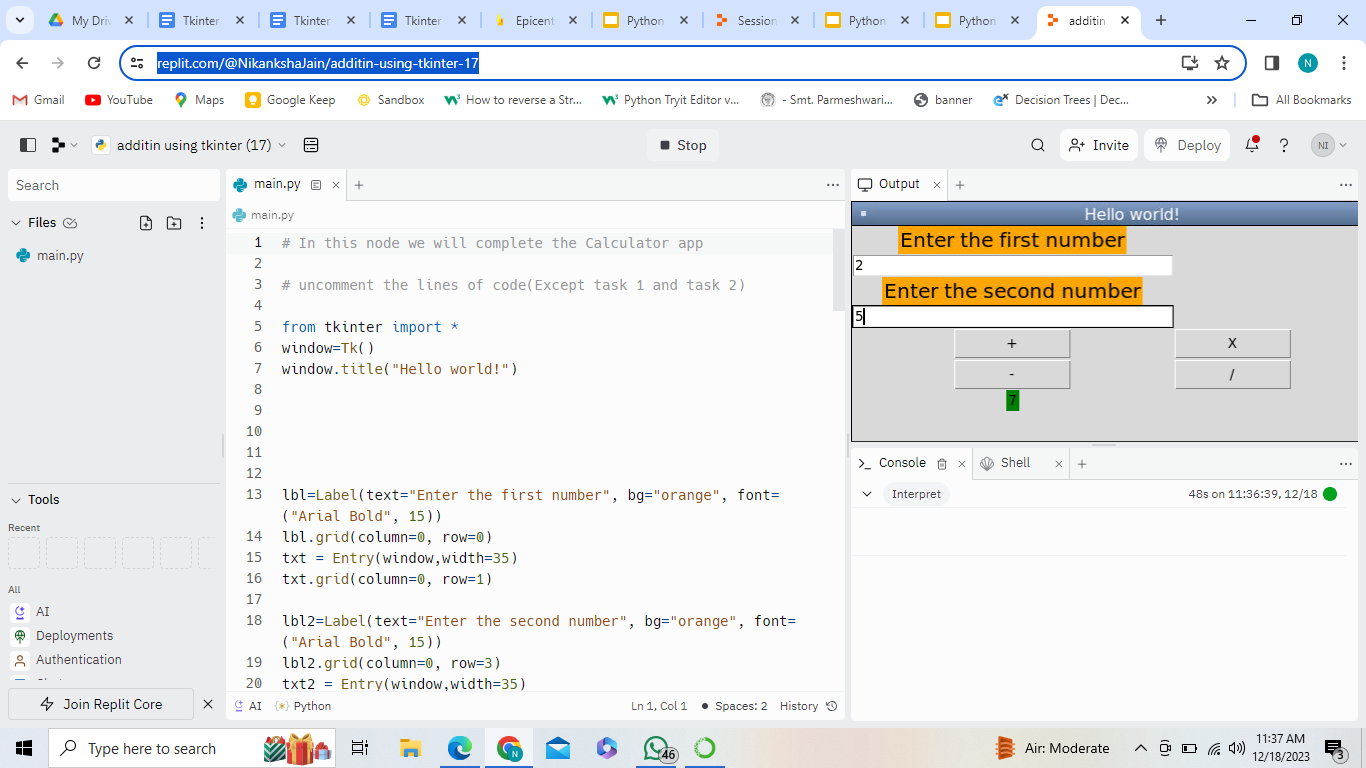
window.mainloop()

**To add image:**[**https://replit.com/@NikankshaJain/image-label-tkinter-1**](https://replit.com/@NikankshaJain/image-label-tkinter-1)

**Solution**Link to Solution: [https://repl.it/@SurajSharma12/Getting-the-Data-Solution  
](https://repl.it/@SurajSharma12/Getting-the-Data-Solution)

Example: Calculator

<https://replit.com/@NikankshaJain/additin-using-tkinter-17>



<https://replit.com/@jnnikanksha89/Session-43-Ex1#main.py>

<https://replit.com/@jnnikanksha89/Session-39-Ex1#main.py>

<https://replit.com/@jnnikanksha89/Session-43-Ex4#main.py>

Scroll bar:<https://replit.com/@udhayakumar311/Session-39-Class-2#main.py>

from tkinter import \*

from datetime import date

root = Tk()

root.geometry("500x500")

root.title("BMI Manager")

# white is default theme

theme = "white"

root.config(bg=theme)

# this function will save the bmi report whenever save option will be clicked!

def saveReport():

file = open("bmiReport.txt" , "a+")

today = date.today()

record = "BMI:" + str(finalResultVariable.get()) + "=" + str(calculatedBMI.get()) + " | Height :" + str(givenHeight.get()) + " | Weight :" + str(givenWeight.get()) + " | Date : " + str(today) + "\n"

file.write(record)

file.close()

def darkTheme():

global theme

theme = "black"

root.config(bg=theme)

def lightTheme():

global theme

theme = "white"

root.config(bg=theme)

# creating menu section for BMI program

# create a menubar

menubar = Menu(root)

root.config(menu=menubar)

# create the file\_menu

file\_menu = Menu(

menubar,

tearoff=0 # this will not disconnect menu from window

)

# add menu items to the File menu

file\_menu.add\_command(label='Save' , command=saveReport)

file\_menu.add\_separator()

# add a submenu

sub\_menu = Menu(file\_menu, tearoff=0)

sub\_menu.add\_command(label='Light theme' , command=lightTheme)

sub\_menu.add\_command(label='Dark theme' , command=darkTheme)

# add the File menu to the menubar

file\_menu.add\_cascade(

label="Change Theme",

menu=sub\_menu

)

# add Exit menu item

file\_menu.add\_separator()

file\_menu.add\_command(

label='Exit',

command=root.destroy

)

menubar.add\_cascade(

label="File",

menu=file\_menu,

underline=0

)

titleLabel = Label(text="BMI Calculator")

titleLabel.grid(row=0 , column=0)

givenHeight = DoubleVar()

givenWeight = IntVar()

calculatedBMI = IntVar()

selectHeight = Label(text="Select Height in meters")

selectHeight.grid(row=1 , column= 0 )

# considering range of the height from 1 meters to 2 meters

heightScale = Scale(from\_=1 , to=2.0 , resolution=0.05 , orient=HORIZONTAL , length=200, variable=givenHeight)

heightScale.grid(row=2 , column=0)

selectHeight = Label(text="Select Weight in KGs")

selectHeight.grid(row=3 , column= 0 )

# considering range of the weight from 2 kgs to 200kgs

heightScale = Scale(from\_=2 , to=200 , resolution=1 , orient=HORIZONTAL , length=200 , variable=givenWeight)

heightScale.grid(row=4 , column=0)

showBMI = Label(text="Your BMI")

showBMI.grid(row=5 , column= 0 )

calculatedBmiScale = Scale(from\_=0.0 , to=200 , resolution=0.05 , orient=HORIZONTAL , length=200 , variable=calculatedBMI)

calculatedBmiScale.grid(row=6 , column=0)

# creating a Entry widget to show the answer in statement with colors

finalResultVariable = StringVar()

finalResultVariable.set("Your BMI Result")

finalResultEntry = Entry(textvariable=finalResultVariable , bg="white")

finalResultEntry.grid(row=10 , column=2)

def getBMI():

if ( givenHeight.get() != 0):

print(givenHeight.get())

print(givenWeight.get())

calculatedBMI.set( givenWeight.get() / ( givenHeight.get() \* givenHeight.get() ) )

if calculatedBMI.get() < 18 :

finalResultVariable.set("You are underweight")

finalResultEntry['bg'] = "red"

elif calculatedBMI.get() > 24.9 :

finalResultVariable.set("You are overweight")

finalResultEntry['bg'] = "red"

else :

finalResultVariable.set("You are healthy")

finalResultEntry['bg'] = "light green"

getBmiButton = Button(text= "Get BMI" , command=getBMI)

getBmiButton.grid(row=5 , column=0)

root.mainloop()

Menu Bar :<https://replit.com/@udhayakumar311/Session-39-Ex1#main.py>

from tkinter import Toplevel, Button, Tk, Menu

top = Tk()

menubar = Menu(top)

file = Menu(menubar, tearoff=0)

file.add\_command(label="New")

file.add\_command(label="Open")

file.add\_command(label="Save")

file.add\_command(label="Save as...")

file.add\_command(label="Close")

file.add\_separator()

file.add\_command(label="Exit", command=top.quit)

menubar.add\_cascade(label="File", menu=file)

edit = Menu(menubar, tearoff=0)

edit.add\_command(label="Undo")

edit.add\_separator()

edit.add\_command(label="Cut")

edit.add\_command(label="Copy")

edit.add\_command(label="Paste")

edit.add\_command(label="Delete")

edit.add\_command(label="Select All")

menubar.add\_cascade(label="Edit", menu=edit)

help = Menu(menubar, tearoff=0)

help.add\_command(label="About")

menubar.add\_cascade(label="Help", menu=help)

top.config(menu=menubar)

top.mainloop()