- Moatasem Elsayed

Bio

- Embedded Linux Software Engineer
- Embedded Software Engineer
- Embedded Software Engineer
- Founder & CEO
- Mentoring For Graduation Project +40
- Instructor at Embedded Systems 75+ G

Eng.moatasem.9@gmail.com

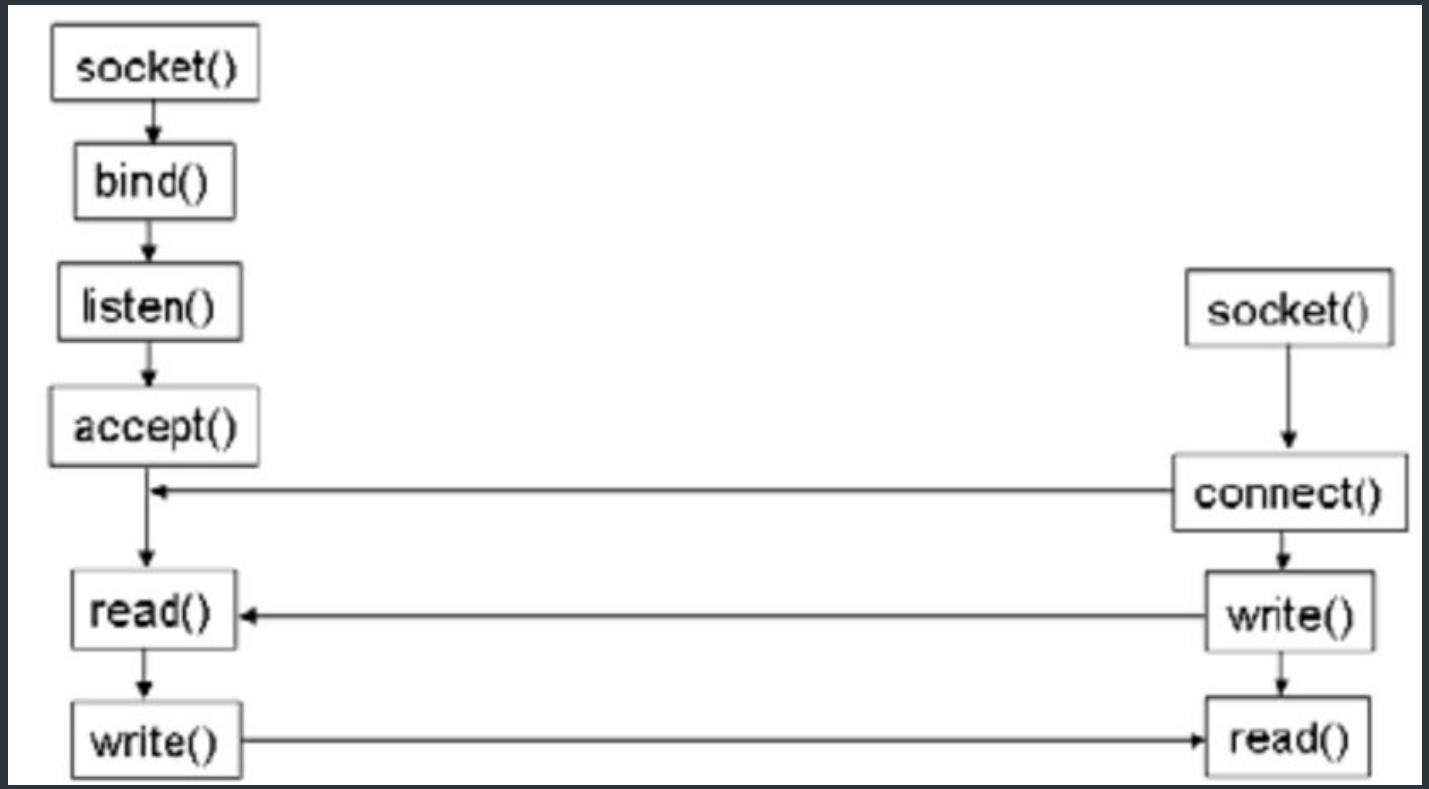
01112932885

Content

Containers

- Socket
- Tkinter
- Label
- Button
- Entry
- Radio Button
- Check Box
- Slider
- messagebox
- New form

Socket programming



server

server.py > ...

```python
import socket

s=socket.socket(socket.AF_INET,socket.SOCK_STREAM)#IPv4,TCP
ip=socket.gethostbyname(socket.gethostname())
print("your ip is : "+ip)
print("============================")
s.bind((ip,5000))
s.listen(5)
while True :
    client,address=s.accept()#waiting
    rodata=client.recv(1024)
    print(f"{address} is sending to you this message {rodata.decode('UTF-8')}"
    print("============================")
    msg=str(input("please enter the message that you want to send: "))
    msg_encoded=msg.encode('UTF-8')
    client.send(msg_encoded)
    client.close()
```

client

```python
import socket

client=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
ip=socket.gethostbyname(socket.gethostname())
print("your ip is : "+ip)
client.connect((ip,5000))
print("============================")
#while True:
msg=str(input("please enter the message that you want to send: "))
msg_encoded=msg.encode('UTF-8')
client.send(msg_encoded)

print("============================")
rodata=client.recv(1024)
print(f"{ip} is sending to you this message {rodata.decode('UTF-8')}")
client.close()
```

C++ and python

```cpp
int main() {
  int socketDesc, newSocket, readSize;
  struct sockaddr_in server, client;
  char message[2000] = {[0]=0};

  // Create socket
  socketDesc = socket(domain: AF_INET, type: SOCK_STREAM, protocol: 0);
  if (socketDesc == -1) {
    std::cerr << "Could not create socket\n";
    return 1;
  }

  // Prepare the sockaddr_in structure
  server.sin_family = AF_INET;
  server.sin_addr.s_addr = INADDR_ANY;
  server.sin_port = htons(hostshort: 8888);

  // Bind
  if (bind(fd: socketDesc, addr: reinterpret_cast<struct sockaddr *>(&server),
           len: sizeof(server)) < 0) {
    std::cerr << "Bind failed\n";
    return 1;
  }

  // Listen
  listen(fd: socketDesc, n: 3);

  std::cout << "Waiting for incoming connections...\n";

  // Accept incoming connection
  int c = sizeof(struct sockaddr_in);
  newSocket = accept(fd: socketDesc, addr: reinterpret_cast<struct sockaddr *>(&client),
                     addr_len: reinterpret_cast<socklen_t *>(&c));
  if (newSocket < 0) {
    std::cerr << "Accept failed\n";
    return 1;
  }
  std::cout << "Connection accepted\n";

  // Receive data from client
```

```python
import socket

def main():
    # Create a socket object
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Server address and port
    server_address = ('localhost', 8888)

    # Connect to the server
    client_socket.connect(server_address)

    # Send data to server
    message = "hello"
    client_socket.sendall(message.encode())

    # Receive response from server
    response = client_socket.recv(1024).decode()
    print(f"Received response from server: {response}")

    # Close the socket
    client_socket.close()

if __name__ == "__main__":
    main()
```

Output

```
D:\Embedded System\Embedded Linux\My presentation\01python\Session 5>python s
erver.py
your ip is : 169.254.56.37
============================
('169.254.56.37', 56887) is sending to you this message hi server iam moatase
m
============================
please enter the message that you want to send: hi client iam server
```

```
PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL                powershell
on 5> python .\client.py
your ip is : 169.254.56.37
============================
please enter the message that you want to send: hi server iam moata
sem
============================
169.254.56.37 is sending to you this message hi client iam server
PS D:\Embedded System\Embedded Linux\My presentation\01python\Sessi
on 5>
```

Nmap

```python
import socket
try:
    sock=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
    portlist=[22,80,443]
    #sock.settimeout(10)
    ip=socket.gethostbyname("www.google.com")
    for i in portlist:
        scan=sock.connect_ex((ip,i))
        if scan == 0:
            print("{} is openned : Service {}".format(i,socket.getservbyport(i)))
        else:
            print("{} is Closed : Service {}".format(i,socket.getservbyport(i)))
except socket.error as e:
    print(e)
```

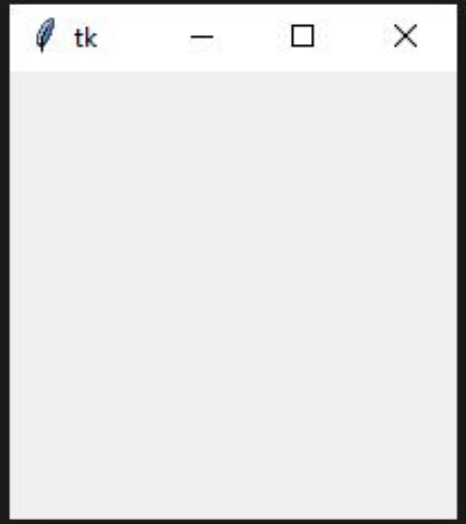
```
PS D:\Embedded System\Embedded Linux\My presentation\01python\Session 6> python .\nmap.py
22 is Closed : Service ssh
80 is openned : Service http
443 is Closed : Service https
PS D:\Embedded System\Embedded Linux\My presentation\01python\Session 6>
```

GUI

- Tkinter
- Label
- Button
- Entry
- Radio Button
- Check Box
- Slider
- messagebox
- New form

GUI (tkinter)

```python
import tkinter
m=tkinter.Tk() #where m is the name of the main window object
m.mainloop()
```

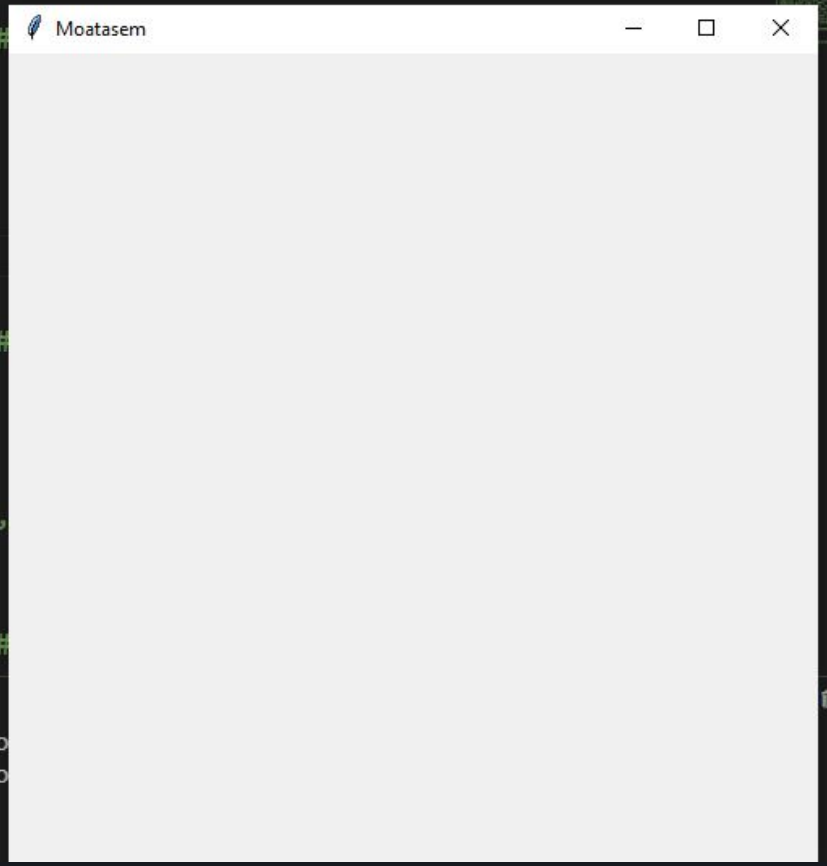


Title , geometry

```
# m.mainloop()
#####################################
from tkinter import *
window=Tk()
window.title("Moatasem")
window.geometry("500x500+150+200")
window.mainloop()

#####################################
# import tkinter as tk
# m = tk.Tk()
# m.title('Counting Seconds')
# button = tk.Button(m, text='Stop',
# button.pack()
# m.mainloop()
#####################################
```

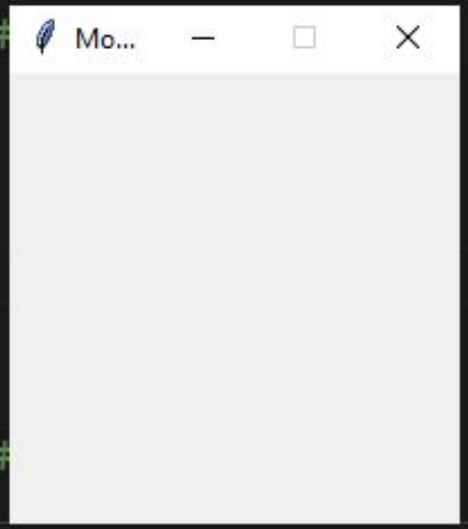
OBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
S D:\Embedded System\Embedded Linux\My presentatio
S D:\Embedded System\Embedded Linux\My presentatio
```



resizable

```
# window.mainloop()
##########################################
from tkinter import *
window=Tk()
window.title("Moatasem")
window.geometry("200x200+150+200")
window.resizable(False,False)
window.mainloop()
##########################################
# from tkinter import *
```

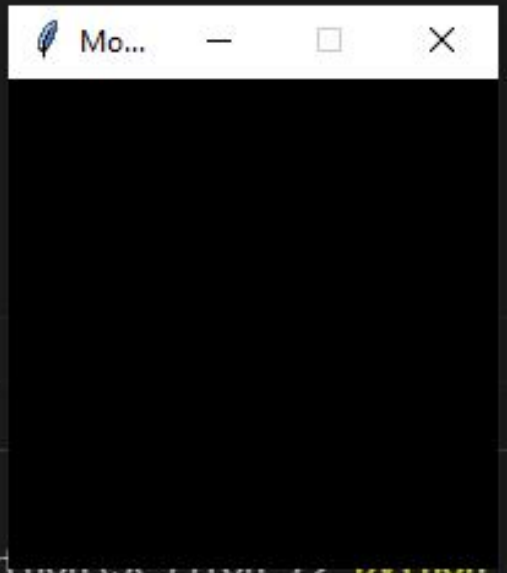


OBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
D:\Embedded System\Embedded Linux\My presentation\01python\Session 7> python .\lab1.py
```

configure

```
17  ##################################3
18  from tkinter import *
19  window=Tk()
20  window.title("Moatasem")
21  window.geometry("200x200+150+200")
22  window.resizable(False,False)
23  window.configure(background="black")
24  window.mainloop()
```



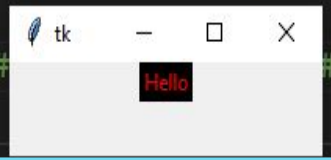
BLEMS OUTPUT DEBUG CONSOLE TERMINAL

D:\Embedded System\Embedded Linux\My presentation\01py

Widget_function

Obj = widget_function(window , options=)

```
from tkinter import *
window=Tk()
window.geometry("200x50+150+200")
Label(window,text="Hello",fg="red",bg="black").pack()
window.mainloop()
#####################
```



1-Label
2-Button
3- Entry
4- Checkbox
5-RadioButton
6-Form
7-List
8-Slider

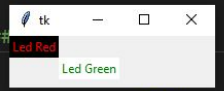
Position on form

pack

```
#########################################
from tkinter import *
window=Tk()
window.geometry("200x50+150+200")
Label(window,text="Led Red",fg="red",bg="black").pack()
Label(window,text="Led Green",fg="green",bg="black").pack()
window.mainloop()
```

grid

```
from tkinter import *
window=Tk()
window.geometry("200x50+150+200")
Label(window,text="Led Red",fg="red",bg="black").grid(row=0,column=0)
Label(window,text="Led Green",fg="green",bg="white").grid(row=1,column=1)
window.mainloop()

#########################################
```



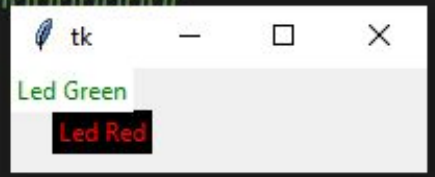
OUTPUT DEBUG CONSOLE TERMINAL

place

```
# window.mainloop()
#########################################
from tkinter import *
window=Tk()
window.geometry("200x50+150+200")
Label(window,text="Led Red",fg="red",bg="black").place(x=20,y=20)
Label(window,text="Led Green",fg="green",bg="white").place(x=20,y=40)
window.mainloop()
#################################################3
```

Mix on positioning but not preferred

```
##########################################
from tkinter import *
window=Tk()
window.geometry("200x50+150+200")
Label(window,text="Led Red",fg="red",bg="black").place(x=20,y=20)
Label(window,text="Led Green",fg="green",bg="white").grid(row=1,column=1)
window.mainloop()
```

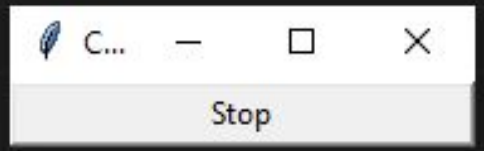


label

```
from tkinter import *
window=Tk()
window.geometry("200x50+150+200")
Label(window,text="Hello",fg="red",bg="black").pack()
window.mainloop()
#########################################
```

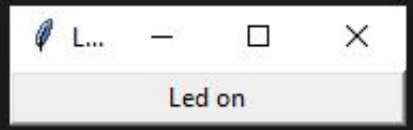
Button(pack)

```python
import tkinter as tk
m = tk.Tk()
m.title('Counting Seconds')
button = tk.Button(m, text='Stop', width=25, command=m.destroy)
button.pack()
m.mainloop()
```



Callback function

```python
import tkinter as tk
def Led_on():
    print("Led is on ")
m = tk.Tk()
m.title('Led Blink')
button = tk.Button(m, text='Led on', width=25, command=Led_on)
button.pack()
m.mainloop()
```

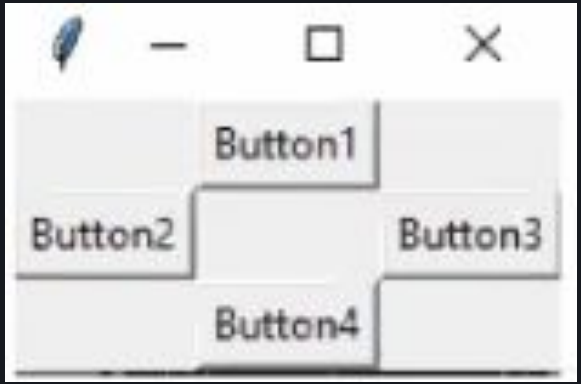


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Embedded System\Embedded Linux\My presentation\01python\Session 7> python .\lab1.
Led is on
Led is on
Led is on
```

Quick Task

Make this template and each button display different name



Frame

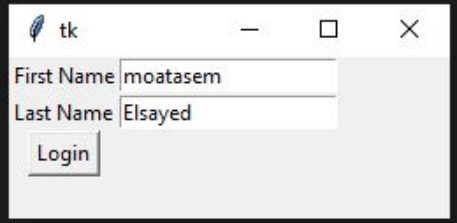
```
from tkinter import *
root = Tk()
frame = Frame(root)
frame.pack()
bottomframe = Frame(root)
bottomframe.pack( side = BOTTOM )
redbutton = Button(frame, text = 'Red', fg ='red')
redbutton.pack( side = LEFT)
greenbutton = Button(frame, text = 'Brown', fg='brown')
greenbutton.pack( side = LEFT )
bluebutton = Button(frame, text ='Blue', fg ='blue')
bluebutton.pack( side = LEFT )
blackbutton = Button(bottomframe, text ='Black', fg ='black')
blackbutton.pack( side = BOTTOM)
root.mainloop()
```

Entry

```
11  ##################################################3
12  import tkinter as tk
13  master = tk.Tk()
14  tk.Label(master, text='First Name').grid(row=0)
15  tk.Label(master, text='Last Name').grid(row=1)
16  e1 = tk.Entry(master)
17  e2 = tk.Entry(master)
18  e1.grid(row=0, column=1)
19  e2.grid(row=1, column=1)
20  tk.mainloop()
21
```

Login Project

```python
from tkinter import *
def Login():
    print("Welcome ",e1.get())
    print("Secoand name  is ",e2.get())
master = Tk()
Label(master, text='First Name').grid(row=0)
Label(master, text='Last Name').grid(row=1)
Label(master, text='Last Name').grid(row=1)
e1 = Entry(master)
e2 = Entry(master)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)
Btn=Button(master,text="Login",command=Login).grid(row=2,column=0)
mainloop()
```

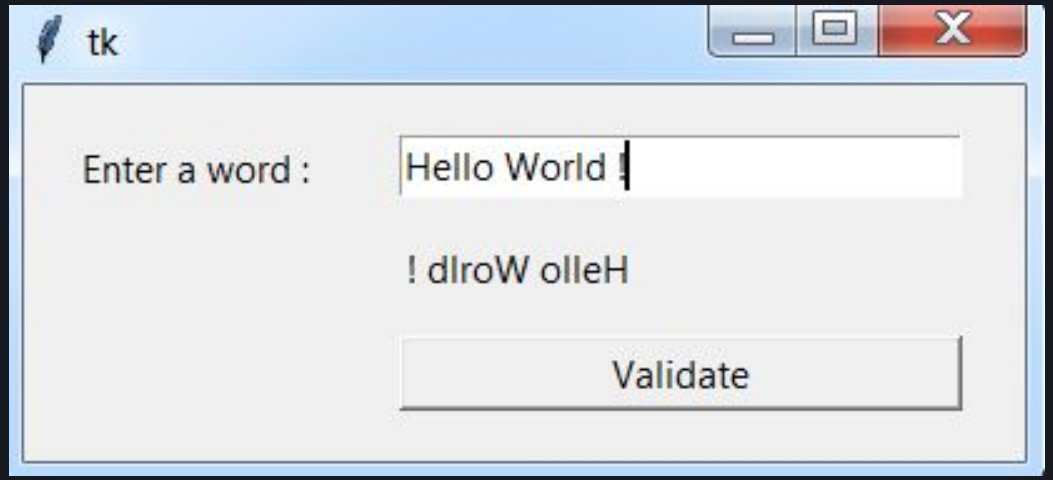


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Embedded System\Embedded Linux\My presentation\01python\Session 7> python .\lab1.py
Welcome  moatasem
Secoand name  is  Elsayed
```

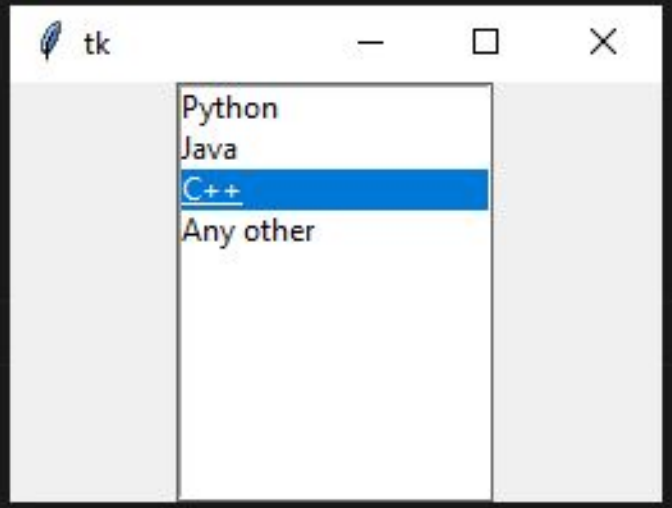
Quick task

Write a program that asks the user to type a word and return him its reverse



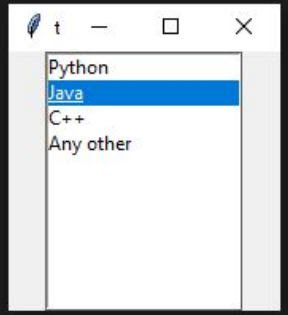
Listbox

```
from tkinter import *

top = Tk()
Lb = Listbox(top)
Lb.insert(1, 'Python')
Lb.insert(2, 'Java')
Lb.insert(3, 'C++')
Lb.insert(4, 'Any other')
Lb.pack()
top.mainloop()
```



Bind for Listbox

```python
def items_selected(event):
    # get selected indices
    selected_index = Lb.curselection()
    # get selected items
    print(selected_index)
    print(Lb.get(selected_index))

top = Tk()
Lb = Listbox(top)
Lb.insert(1, 'Python')
Lb.insert(2, 'Java')
Lb.insert(3, 'C++')
Lb.insert(4, 'Any other')
Lb.pack()
Lb.bind('<<ListboxSelect>>', items_selected)
top.mainloop()
##########################################################
```

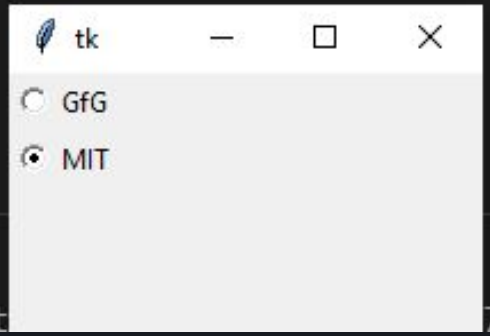


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Embedded System\Embedded Linux\My presentation\01python\Session 7> pyth
(0,)
Python
(1,)
Java
```

RadioButton

```
71
72  from tkinter import *
73  root = Tk()
74  v = IntVar()
75  Radiobutton(root, text='GfG', variable=v, value=1).pack(anchor=W)
76  Radiobutton(root, text='MIT', variable=v, value=2).pack(anchor=W)
77  mainloop()
78
79
80
```

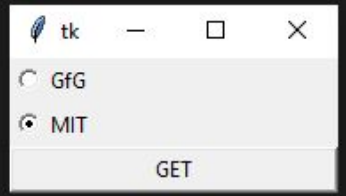


BLEMS OUTPUT DEBUG CONSOLE TERMINAL

D:\Embedded System\Embedded Linux\My present 7> python .\lab1.

Action

```python
from tkinter import *

def DisplayValue():
    global v
    print(v.get())
root = Tk()
v = IntVar()
Radiobutton(root, text='GfG', variable=v, value=1).pack(anchor=W)
Radiobutton(root, text='MIT', variable=v, value=2).pack(anchor=W)
button = Button(root, text='GET', width=25, command=DisplayValue)
button.pack()
mainloop()
```

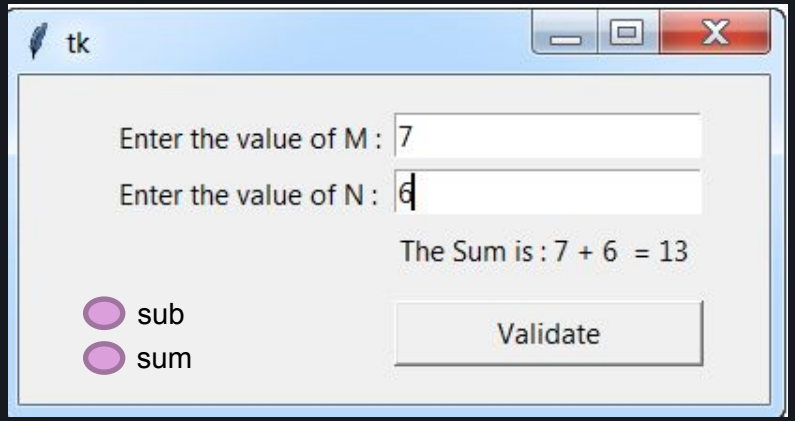


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
PS D:\Embedded System\Embedded Linux\My presentation\01python\Session 7> python .\lab1.py
1
2
```

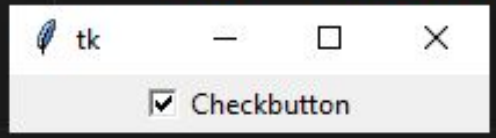
Quick Task

- Create a graphical application in Python Tkinter that asks the user to enter two integers and displays their sum



checkbox

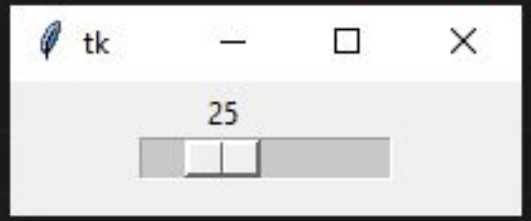
```
147
148 import tkinter
149 parent_widget = tkinter.Tk()
150 checkbutton_widget = tkinter.Checkbutton(parent_widget,
151                                          text="Checkbutton")
152 checkbutton_widget.select()
153 checkbutton_widget.pack()
154 tkinter.mainloop()
155
156
157
```



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

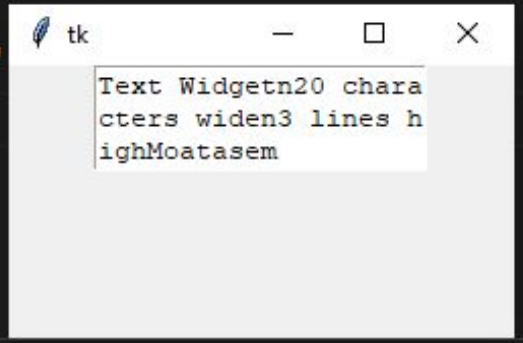
scale

```python
##########################################################
import tkinter
parent_widget = tkinter.Tk()
scale_widget = tkinter.Scale(parent_widget, from_=0, to=100,
                             orient=tkinter.HORIZONTAL)
scale_widget.set(25)
scale_widget.pack()
tkinter.mainloop()
```



Text Widget

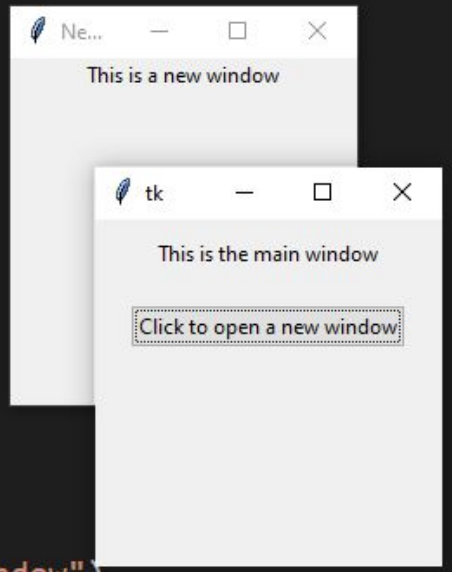
```
(variable) parent_widget: Tk
parent_widget = tkinter.Tk()
text_widget = tkinter.Text(parent_widget,
                           width=20, height=3)
text_widget.insert(tkinter.END,
    "Text Widgetn20 characters widen3 lines high"
text_widget.insert(tkinter.END,
    "Moatasem")
text_widget.pack()
tkinter.mainloop()
```



EMS OUTPUT DEBUG CONSOLE TERMINAL

New Form

```python
def openNewWindow():
    # Toplevel object which will
    # be treated as a new window
    newWindow = Toplevel(master)
    # sets the title of the
    # Toplevel widget
    newWindow.title("New Window")
    # sets the geometry of toplevel
    newWindow.geometry("200x200")
    # A Label widget to show in toplevel
    Label(newWindow,
        text ="This is a new window").pack()

label = Label(master,text ="This is the main window")
label.pack(pady = 10)
btn = Button(master,text ="Click to open a new window",command = openNewWindow)
btn.pack(pady = 10)
# mainloop, runs infinitely
mainloop()
```



Out of scope

LabelFrame Widget
Canvas Widget
Menu Widget
OptionMenu Widget
LabelFrame Widget

opencv

To open camera

penCv_labs > camera.py > main

```python
# pip install opencv-python
import cv2


def main():
    # Open the default camera (index 0)
    cap = cv2.VideoCapture(0)
    # Check if the camera was opened successfully
    if not cap.isOpened():
        print("Error: Unable to access the camera.")
        return
    # Set the window name for the live video stream
    window_name = "Camera Preview"
    while True:
        # Capture frame-by-frame
        ret, frame = cap.read()
        # Check if the frame was captured successfully
        if not ret:
            print("Error: Unable to capture frame.")
            break
        # Display the frame in a window
        cv2.imshow(window_name, frame)
        # Break the loop when the 'q' key is pressed
        if cv2.waitKey(1) == ord('q'):
            break

    # Release the camera and close the window
    cap.release()
    cv2.destroyAllWindows()


if __name__ == "__main__":
    main()
```

Capture an image

```python
import cv2


def main():
    # Open the default camera (index 0)
    cap = cv2.VideoCapture(0)
    if not cap.isOpened():
        print("Error: Unable to access the camera.")
        return
    # Set the window name for the live video stream
    window_name = "Camera Preview"
    cv2.namedWindow(window_name)

    while True:
        # Capture frame-by-frame
        ret, frame = cap.read()
        # Check if the frame was captured successfully
        if not ret:
            print("Error: Unable to capture frame.")
            break
        # Display the frame in a window
        cv2.imshow(window_name, frame)

        # Check for the 'c' key press
        key = cv2.waitKey(1)
        if key == ord('c'):  # Press 'c' key to take a picture
            # Save the captured frame as an image
            cv2.imwrite("captured_image.jpg", frame)
            print("Image captured successfully!")
        # Break the loop when the 'q' key is pressed
        if key == ord('q'):
            break
    # Release the camera and close the window
    cap.release()
    cv2.destroyAllWindows()


if __name__ == "__main__":
    main()
```

Os.system vs os.popen vs subprocess.popen

```python
import os

# Execute a shell command using os.system
os.system("ls -l")
```

```python
import os

# Execute a shell command using os.popen and read its output
output_file = os.popen("ls -l")
output_text = output_file.read()
print(output_text)
```

```python
import subprocess

# Execute a shell command using subprocess.Popen and capture its output
result = subprocess.Popen(["ls", "-l"], stdout=subprocess.PIPE, text=True)
output_text, _ = result.communicate()
print(output_text)
```

Cont ..

1.os.system(command):

- `os.system` is the **simplest** of the three and is used to run a command in a subshell.
- It returns the **exit status** of the command executed (return code), not the actual output of the command.

2. **os.popen(command[, mode]):**
   - `os.popen` is an older function that is similar to `os.system` but allows you to capture the output of the command as a file-like object.
   - The `mode` parameter is optional and specifies the mode in which the command's output is opened. The default is `"r"`, which means read mode.
   - You can use the file-like object returned by `os.popen` to read the output of the command.

Cont ...

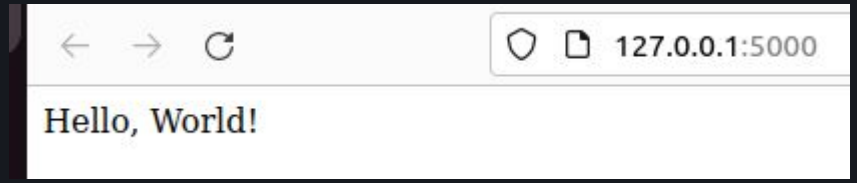
`subprocess.Popen(args, *, stdin=None, stdout=None, stderr=None):`

- `subprocess.Popen` is a more powerful and flexible way to run shell commands and interact with them programmatically.
- It provides more control over input/output streams and other process-related attributes.
- The `args` parameter is the command and its arguments provided as a list of strings.
- You can specify `stdin`, `stdout`, and `stderr` parameters to redirect the standard input, output, and error streams of the command, respectively.
- It returns a Popen object, which can be used to communicate with the process and obtain its output.

```
result = subprocess.Popen(["ls", "-l"],
                          stderr=subprocess.PIPE, stdout=subprocess.PIPE, text=True)
output_text, error_text = result.communicate()
print((output_text))
print((error_text))
```

Hello world with flask

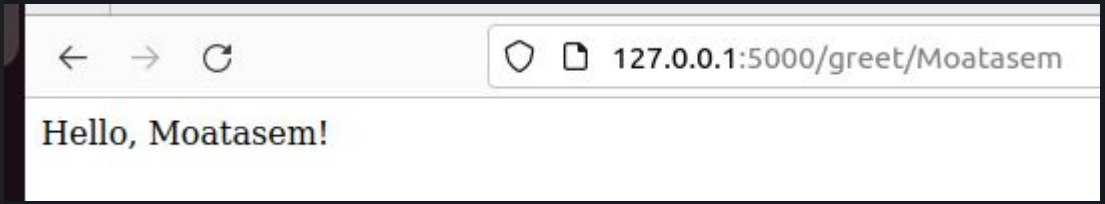
```python
# pip install flask
from flask import Flask

app = Flask(__name__)


@app.route('/')
def hello_world():
    return 'Hello, World!'


if __name__ == '__main__':
    app.run()
```



Cont ..

sc2.py > ...

```python
from flask import Flask

app = Flask(__name__)


@app.route('/greet/<name>')
def greet_person(name):
    return f'Hello, {name}!'


if __name__ == '__main__':
    app.run()
```



Cont ..


127.0.0.1:5000
Moatasem Elsayed


.vscode
Flask
templates
index.html
dynamic.py
hello.py
render.py
network
build
client.py
CMakeLists.txt
from flask import Flask, render_template
app = Flask(__name__)
app.route('/')
ef index():
return render_template('index.html', title='My Name', message='Moatasem Elsayed')
f __name__ == '__main__':


<!DOCTYPE html>
<html>
<head>
<title>{{ title }}</title>
</head>
<body>
<h1>{{ message }}</h1>
</body>
</html>

Cont ..

POST localhost:5000/login Send

Query Headers 2 Auth Body 1 Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

Form Encoded

☑	username	moatasem
☑	password	elsayed
☐	name	value

Status: 200 OK Size: 21 Bytes Time: 2 ms

Response Headers 5 Cookies Results Docs

```
1  Logged in as moatasem
```

ask > login.py > ...

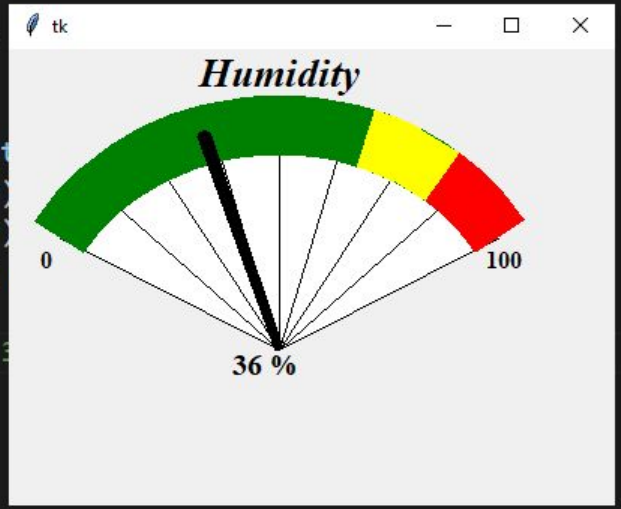
```python
from flask import Flask, ren

app = Flask(__name__)



@app.route('/login', methods=['POST'])
def login():
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        return f'Logged in as {username}'
    return render_template('login.html')


if __name__ == '__main__':
    app.run()

```

templates > login.html > ...

```html
<!DOCTYPE html>
<html>
<head>
    <title>Login</title>
</head>
<body>
    <form method="post">
        <label>Username:</label>
        <input type="text" name="username">
        <br>
        <label>Password:</label>
        <input type="password" name="password">
        <br>
        <input type="submit" value="Login">
    </form>
</body>
</html>
```

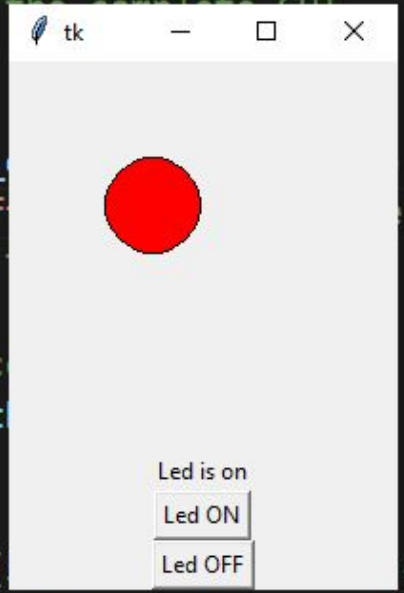
Task: after you understand try to make gauge like pic


```
cnvs.create_arc(coord, start=50, extent=20, outline="yellow", style= "arc", width=40)
# add needle/value pointer
id_needle = cnvs.create_arc(coord, start= 119, extent=1,

# Add some labels
cnvs.create_text(180,15,font="Times 20 italic bold", text
cnvs.create_text(25,140,font="Times 12 bold", text=low_r)
cnvs.create_text(330,140,font="Times 12 bold", text=hi_r)
id_text = cnvs.create_text(170,210,font="Times 15 bold")

root.after(3000, update_gauge)#call  update_gauge after

root.mainloop()
```



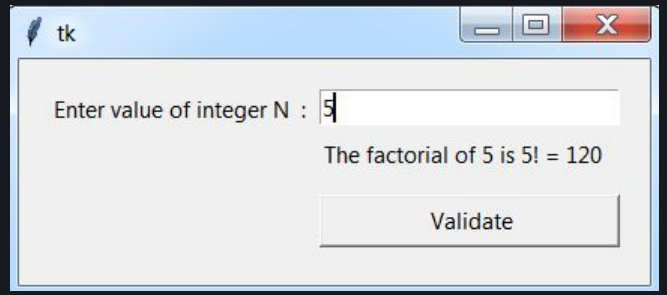
Task

```
6   def Led_on():
7           my_canvas.itemconfig(my_oval, fill="red")  # Fill the circle
8           label                                  on")  # Updating the label
9           root
10
11  def Led_off(
12          my_ca                                          # Fill the circl
13          label                                          label
14          root
15
16  root = tk.Tk
17  my_canvas =                                            # Create 200x200
18  my_canvas.pack()
19
20  my_oval = my_canvas.create_oval(                       # Create a circle
```



Task

- Write a program in Python that displays a window to the user that asks them to enter an integer N and displays its factorial



Task

1

2

3

Create server receive multiple clients and keep alive

```
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

D:\python\Workspace\socket>python server.py
('192.168.1.27', 57056)
{"id": "XYZ", "Value": 385, "type": "Temperture"}
('192.168.1.27', 57057)
{"id": "XYZ", "Value": 211, "type": "Temperture"}
('192.168.1.27', 57058)
{"id": "XYZ", "Value": 103, "type": "Temperture"}
('192.168.1.27', 57059)
{"id": "XYZ", "Value": 170, "type": "Temperture"}
('192.168.1.27', 57060)
{"id": "XYZ", "Value": 293, "type": "Temperture"}
('192.168.1.27', 57061)
{"id": "XYZ", "Value": 125, "type": "Temperture"}
('192.168.1.27', 57062)
{"id": "XYZ", "Value": 88, "type": "Temperture"}
('192.168.1.27', 57063)
{"id": "XYZ", "Value": 131, "type": "Temperture"}
('192.168.1.27', 57064)
{"id": "XYZ", "Value": 101, "type": "Temperture"}
('192.168.1.27', 57065)
{"id": "XYZ", "Value": 305, "type": "Temperture"}
```

```
Microsoft Windows [Version 10.0.19043.1288]
(c) Microsoft Corporation. All rights reserved.

D:\python\Workspace\socket>python ClientThreads.py
I got the message
I got the message
I got the message
I got the message
I got the message
I got the message
I got the message
I got the message
I got the message
```

Session 1 (3hr)	***-introduction to python*** ***-datatypes*** ***-in/out,,loops,conditions***
Session 2(3hr)	***-strings*** ***-function*** ***-modules*** ***-list,tuple,set*** ***-pyautogui***

Session 3(3hr)	***-dictionary*** ***-class*** ***-files*** ***-csv*** ***-threads*** ***-error handling***
Session 4 (3hr)	***-socket*** ***-GUI*** ***-advanced modules***