

Tkinter

- Used ~~for~~ for Creating GUI Applications
- Inbuilt module

Using Tkinter

- first import all from tkinter

```
from tkinter import *
```

- Then create a window (a new screen)

```
root = Tk()
```

This command creates a window & assigns it to variable root.
All transactions to & from window (Eg- Getting entry from user)
happens through the variable assigned to it ie root in this case.

- root variable is similar to a file handle used to open & manipulate the file

- The above command creates a free space which can be used to pack widgets

- All widgets to be packed After this

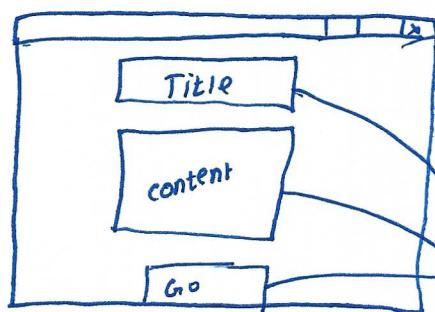
- After all widgets are packed, this command is to be used.

```
root.mainloop()
```

This statement retains every widget packed as widgets may appear and disappear immediately.

Frame & Canvas

- Frame & Canvas are entities where widgets can be packed
(like an invisible container)
- It is like a defined space allocated inside the Root (main window)
- A Frame can be packed inside another frame.



- Creating frame -

`<Frame-name> = Frame (root [], width = 200, height = 300)`

`<frame-name>. pack()`

Frame name
can be any variable

place to pack the
frame somewhere in the page.
it can be also packed inside another frame.

- Canvas is similar to a frame.
It is used for creating shapes, lines, etc...
- Canvas is not used in this project.

Pack

- The pack command is used to pack to pack the given widget somewhere in the free space.

`frame1 . pack(side = LEFT)`

↑
name of a
frame that is
to be packed

↑
place to
pack it.

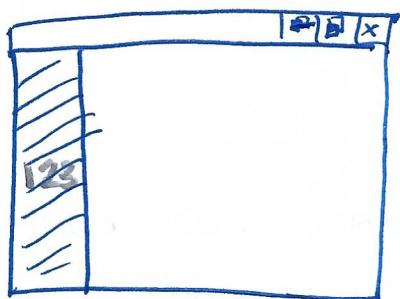
Can be packed in -

- TOP
- BOTTOM
- LEFT
- RIGHT
- :

→ fill attribute of Pack -

```
lbl = Label (root, text = '123', bg = 'blue')  
lbl.pack (side = LEFT, fill = Y)
```

The output will be like



Note - Packing can be also done with grid

Real life Example of frame -

Suppose a flat is bought without walls. It will cause difficulties organising

If the flat has walls, then organising will be easy.
Different activities can be held in a defined space.

Hence rooms in a flat are like frames in a root window

Methods for using tkinter widgets

① Creating a Label - Label is used to pack text within a window.

Eg1 - `lbl = Label (root, text = 'text to be written')`

↑ ↑ ↑
variable for Place to text to be
this particular Label be packed. displayed.
Also can be packed
in frame

~~Label creation~~
`lbl.pack()`

↑
This packs the created label inside the free space
created by ~~root~~.

Try

```
from tkinter import *
root = Tk()
lbl = Label (root, text = 'Hello World')
lbl.pack()
root.mainloop()
```

Eg2 - `lbl2 = Label (root, text = 'Hello', fg = 'blue', font = 'font face, size')`

↑ ↑ ↑ ←
The font color Like Times New Roman Size of
of the text text.

`lbl2.pack()`.

Other ~~attributes~~ of Label -

→ `bg` - background color

②

Button -

- Button is used for user to navigate within a page
- It can trigger some action.
- To make a button do something, a function must be defined.

Eg - `btn = Button(root, text='click me')`

Variable for button → `btn.pack()`



This button does nothing.

Eg - Try
from tkinter import *
def hi():
 print ('hello')

main

Buttons

root = TK()

btn2 = Button(root, text='click me', command=hi)

btn2.pack()

root.mainloop()

↑
Passing argument is
not easy

→ Button can also have a foreground & background color by using attributes such as `fg='<color>'` or `bg='<color>'`

Additional info - Mouse click Events - bind method

Try

```
from tkinter import *
root = Tk()

def leftclick():
    print('left')
def rightclick():
    print('right')

frame = Frame(root, width=100, height=200)
frame.bind('<Button-1>', leftclick)
frame.bind('<Button-2>', rightclick)
frame.pack()
root.mainloop()
```

③ Entry Using text boxes (VIMP)

→ These are used for acquiring information from users

Eg - `txt = Entry (root, [, width = 30])` # There is no attribute for height
`txt.pack()`

→ This just lets us type data but nothing happens.

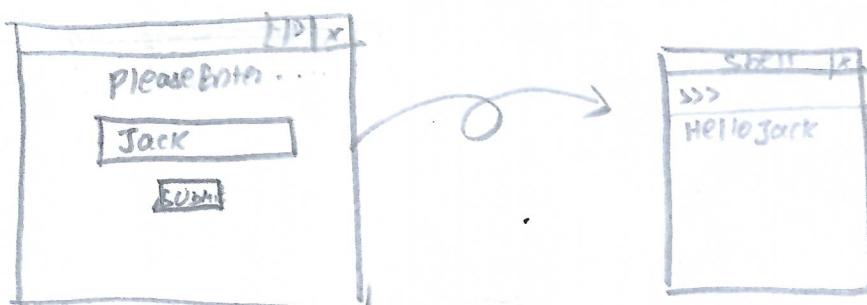
→ To collect data, something has to trigger it -

→ A button may be used...

Try

```
from tkinter import *
root = Tk()
def readandprint():
    box = Entry()
    name = box.get()
    print ('Hello', name)

# main
lb1 = Label (root, text='please Enter your name')
lb1.pack()
box = Entry (root, width = 30)
box.pack()
btn = Button (root, text='Submit', command = readandprint)
btn.pack()
```



Attributes of Entry

- ① width - Width of text box
- ② bg - background of text box
- ③ show - Show = '*' Used to display other characters (used in password Entry field)

~~Another method~~ Another method -

If box is name of text box,

box.insert(0, 'Default text')

This will enter a default text in the beginning.

V-IMP Note → ~~getting value through Entry~~ will always return a string

Other types of Entry -

I) Drop down menu

Try

from tkinter import *

root = TK()

options = Frame(root)

options.pack()

items = [1, 2, 3]

a variable = StringVar(options)

a variable.set(items[0])

w = OptionMenu(option variable, *items)

w.pack()

Btn = Button(root, text='submit', command=go)

Btn.pack

root.mainloop()

def go():

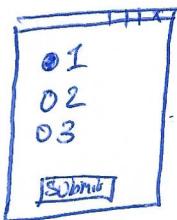
~~print(a variable)~~

print(variable.get())

II) Radio Buttons.

Try

```
from tkinter import *
root = Tk()
def go():
    print(var.get())
var = IntVar()
rb1 = Radiobutton(root)
rb2 = " "
rb3 = " "
rb1.config(text='1', variable=var, value=1)
rb2.config(text='2', variable=var, value=2)
rb3.config(text='3', variable=var, value=3)
rb1.pack()
rb2.pack()
rb3.pack()
subbtn = Button(root, text='Submit', command=go)
subbtn.pack()
root.mainloop()
```



④ Inserting Images

Only tkinter module is used - PIL is not used.

Code

```
<image variable> = PhotoImage (file='<image path>')
```

```
<label variable> = Label(root, image=<image variable>)
```

```
<label variable>.pack()
```

→ Images are added as label only

→ It is important to add root.mainloop here after packing

An image can be also added as a button.

Code

```
<image variable> = PhotoImage (file='<image path>')
```

```
<button variable> = Button (root, image=<image variable>, command=
```

```
<button variable>.pack()
```

→ Eg - 'hellocic.png'

<some fn>)

It can be also a radiobutton.

Code for Captcha

```

from tkinter import *
import random

def captcha():
    global errorstreams
    def go():
        global errorstreams
        errorstreams.pack_forget()#This command deletes the frame
        errorstreams=Frame(root)#This creates the frame again
        errorstreams.pack(side=BOTTOM)
        captcha=captchaentry.get()#get the value entered by the user
        if listoffilesofcaptcha[captchanumber][1]!=captchaentry.get():#Evaluates the captcha
            lblcaptchastats=Label(errorstreams,text='Wrong Captcha Please Retry', fg='red')
            lblcaptchastats.pack()
        else:
            lblcaptchastats=Label(errorstreams,text='Success', fg='green')
            lblcaptchastats.pack()

        captchaframe = Frame(root,width=1000, height= 1000)# A frame created
        captchaframe.pack()
        captchanumber= random.randint(1,5)-1 #picks a random captcha
        photo=PhotoImage(file=listoffilesofcaptcha[captchanumber][0])#loads its image path
        captchaphoto=Label(captchaframe,image=photo)
        captchaentry = Entry(captchaframe, width=20)#asking for the value
        captchaentry.pack(side=BOTTOM)
        captchaphoto.pack()

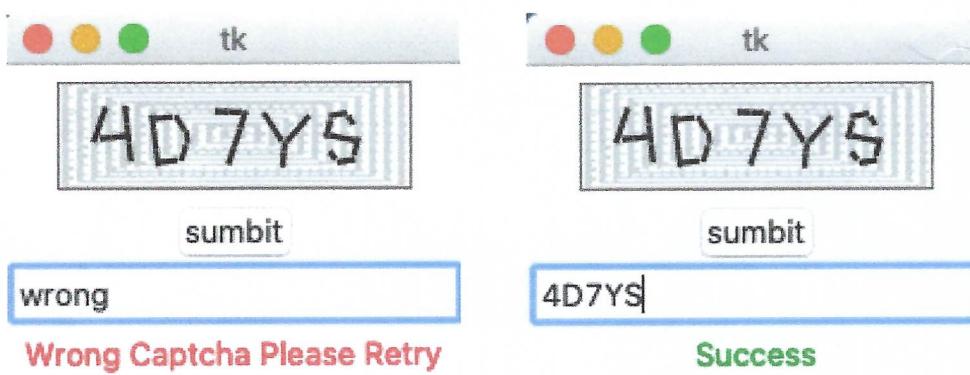
        btn=Button(captchaframe,text='sumbit',command=go)#submit the value for evaluation
        btn.pack()

    errorstreams=Frame(root)#This frame is just to give status of the entered captcha
    errorstreams.pack(side=BOTTOM)
    root.mainloop()

root =Tk()
listoffilesofcaptcha=[('captcha 1.png','28ivw'),('captcha 2.png','k4ez'),('captcha 3.png','4D7YS'),
                      ('captcha 4.png','6ne3'),('captcha 5.png','e5hb')]
#list of Paths of captcha image and their respective value
captcha()

```

OUTPUT



⑤

Destroying a frame V.I.MP

- If we add more frames, the frames & its content just gets packed on top of other.
- Hence to create a new page all such frames have to be destroyed & new frames have to be created.
- In the project a main page is static but the interlinking effect is created by destroying a frame & creating a new frame for the next page.

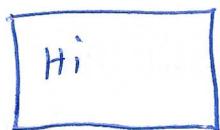
Code

```
<frame name>.pack_forget()
```

After this code is executed the frame will no longer exist.

Uses

- internal linking
- It can be used to modify text



initially

Can be converted to



by Destroying the frame,
Creating the frame again
packing the new label

Sample Program

```
import tkinter as tk  
def back_to_main():  
    firstframe.pack_forget()  
    mainframe.pack_forget()  
    Secondpage()  
def
```

```
from tkinter import *  
def tosecondpage():  
    global firstpageframe  
    firstpageframe.pack_forget()  
    Secondpage()  
  
def tofirstpage():  
    global secondpageframe  
    secondpageframe.pack_forget()  
    firstpage()  
  
def firstpage():  
    global firstpageframe  
    firstpageframe = Frame(root)  
    firstpageframe.pack()  
    lbl = Label(firstpageframe, text='In first page now')  
    lbl.pack()  
    btn = Button(firstpageframe, text='Second Page', command=  
                tosecondpage)  
    btn.pack()  
root.mainloop()
```

```
def secondpage :  
    global secondpageframe  
    secondpageframe = Frame(root)  
    #first  
    secondpageframe.pack()  
    lb1 = Label(secondpageframe, text='In Second Page Now')  
    lb1.pack()  
  
    btn = Button(secondpageframe, text='First Page', command=tofirstpage)  
    btn.pack()  
    root.mainloop()  
  
# Main  
root = Tk()  
#first  
firstpage()
```

→ This ~~is~~ code changes the content but the ~~page~~ page is just the same.

⑥ Message box

→ This is another ~~window~~ window which pops out & Shows info or asks question.

Code-I - Showing info

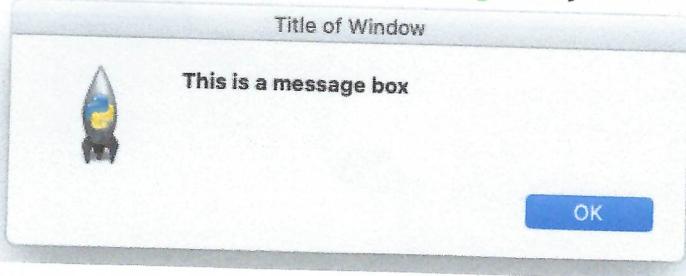
```
import tkinter.messagebox  
tkinter.messagebox.showinfo('<title>', '<message>')
```

Code II - Ask a question

```
import tkinter.messagebox  
answer = tkinter.messagebox.askquestion('<window title>', '<question>')  
if answer == 'yes':  
    print('<..>')  
else:  
    print('<..>')
```

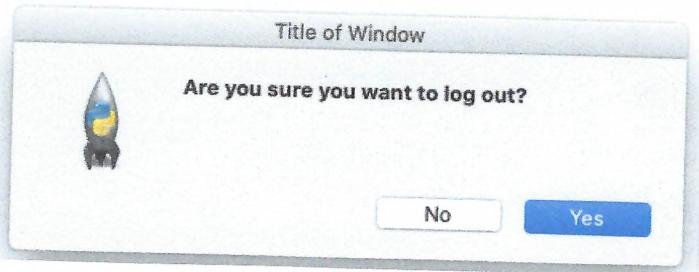
CODE I EXAMPLE

```
import tkinter.messagebox  
tkinter.messagebox.showinfo('Title of Window', 'This is a message box')
```



CODE II EXAMPLE

```
import tkinter.messagebox  
answer=tkinter.messagebox.askquestion('Title of Window', 'Are you sure you want to log out?')  
if answer == 'yes':  
    print('User wants to log out')  
else:  
    print('User wants to stay')
```



Output in shell

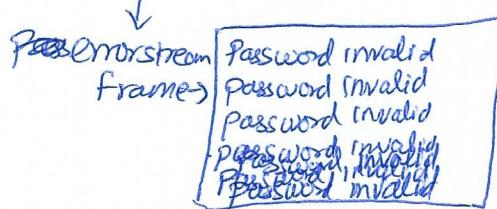
```
User wants to log out  
=> #yes was clicked  
---
```

A Method to Evaluate for Entered password

```
# pwd is the password entered by the user
countforpasswordcaptletter, countforpasswordsmallletter, countforpasswordnumber=0,0,0 #PASSWORD EVALUATION
for varia in pwd:
    if ord(varia) in range(65,91):
        countforpasswordcaptletter+=1
    elif ord(varia) in range(97,123):
        countforpasswordsmallletter+=1
    elif ord(varia) in range(48,58):
        countforpasswordnumber+=1
if countforpasswordcaptletter==0 or countforpasswordsmallletter == 0 or countforpasswordnumber ==0 or len(pwd)<8:
    passwordinvalidlabel=Label(errorstreams,text='Your Password is weak', fg='red')#EVALUATION FOR STRONG PASSWORD
    passwordinvalidlabel.pack()

if pwd != pwdconf:# IF PASSWORD ENTERED MATCHES WITH THE CONFIRMED ONE
    passdoesnotmatchlabel=Label(errorstreams,text='Passwords Entered does not match', fg='red')
    passdoesnotmatchlabel.pack()
```

- Checks for validity of password
- Error- Stream is a frame in root to print ~~Error~~ ~~error~~ Error if pass does not meet Criteria
- The frame gets destroyed & recreated everytime submit button is clicked
- If frame is not cleared the Output in screen will become like this after multiple errors in password



A Method to use String as Variable in Python

→ If $x = 'abc'$ is a string,
this abc can be treated as a variable & a
value can be assigned to it with the following
Code

Eg - $x = 'abc'$
`globals()[x] = 123`
`print(abc)`
`print(x)`
`print(globals()[x])`

Output →

123
'abc'
123.

SMTP Library

SMPP → Simple Mail Transfer Protocol

Used for sending mails from Python

Code

```
import smtplib  
Try:  
    m = '<recipients email>'  
    Subj = '<subject of email>'  
msg  
    msg = '<message of email . use + to concatenate string.  
            Dont use comma. All integers must be converted to  
            String here . use \n for a new line in mail>'
```

With `smtplib.SMTP('smtp.gmail.com', 587)` as `smtp: gmail` → Port number for gmail

```
    smtp.ehlo() # identifying ourselves to server telling hello  
                # ehlo is for extended Smtp  
    smtp.starttls() # everything after this is encrypted  
                    # nospace here  
    smtp.ehlo()  
    smtp.login('<ouremail>', '<our password - App pass created>')  
    msg = f'Subject: {Subj}\n\n{msg}' # formatting our email  
    smtp.sendmail('<ouremail>', m, msg)
```

Except :

```
print('Please check your Internet connection')
```

But Errors may occur due to Variety of other reasons

#Sample code for otp verification

```
import smtplib

m=input("Please enter your email id: ")#receiver address
print('Verify your account')
val= random.randint(1000,9999)
with smtplib.SMTP('smtp.gmail.com', 587) as smtp:
    smtp.ehlo()
    smtp.starttls()
    smtp.ehlo()
    smtp.login('<sender address>', '<sender password ie.app password>')
    subject = 'OTP Verification'
    body = 'Your OTP for you Library account is ',val
    msg = f'Subject: {subject}\n\n{body}'
    smtp.sendmail('<sender address>',m,msg)

for a in range (1,5):
    ver=int(input('pls enter the otp you have received in your email: '))
    if ver == val:
        print("You have entered a correct otp")
        break
    else:
        print("it is not a correct otp you have entered")
        continue
```