CREATE YOUR FIRST GUI APPLICATION

FIRST, WE WILL IMPORT TKINTER PACKAGE AND CREATE A WINDOW AND SET ITS TITLE:

```
FROM TKINTER IMPORT *



WINDOW = TK()



WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")



WINDOW.MAINLOOP()
```

THE RESULT WILL BE LIKE THIS:

AWESOME!! OUR APPLICATION JUST WORKS.

THE LAST LINE WHICH CALLS MAINLOOP FUNCTION, THIS FUNCTION CALLS THE ENDLESS LOOP OF THE WINDOW, SO THE WINDOW WILL WAIT FOR ANY USER INTERACTION TILL WE CLOSE IT.

IF YOU FORGET TO CALL THE MAINLOOP FUNCTION, NOTHING WILL APPEAR TO THE USER.

CREATE A LABEL WIDGET

TO ADD A LABEL TO OUR PREVIOUS EXAMPLE, WE WILL CREATE A LABEL USING THE LABEL CLASS LIKE THIS:

```
LBL = LABEL(WINDOW, TEXT="HELLO")
```

THEN WE WILL SET ITS POSITION ON THE FORM USING THE GRID FUNCTION AND GIVE IT THE LOCATION LIKE THIS:

```
LBL.GRID(COLUMN=0, ROW=0)
```

SO THE COMPLETE CODE WILL BE LIKE THIS:

```
FROM TKINTER IMPORT *


WINDOW = TK()


WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")


LBL = LABEL(WINDOW, TEXT="HELLO")


LBL.GRID(COLUMN=0, ROW=0)


WINDOW.MAINLOOP()
```

AND THIS IS THE RESULT:

WITHOUT CALLING THE GRID FUNCTION FOR THE LABEL, IT WON'T SHOW UP.

SET LABEL FONT SIZE

YOU CAN SET THE LABEL FONT SO YOU CAN MAKE IT BIGGER AND MAYBE BOLD. YOU CAN ALSO CHANGE THE FONT STYLE.

TO DO SO, YOU CAN PASS THE FONT PARAMETER LIKE THIS:

```
LBL = LABEL(WINDOW, TEXT="HELLO", FONT=("ARIAL BOLD", 50))
```

NOTE THAT THE FONT PARAMETER CAN BE PASSED TO ANY WIDGET TO CHANGE ITS FONT NOT LABELS ONLY.

GREAT, BUT THE WINDOW IS SO SMALL, WE CAN EVEN SEE THE TITLE, WHAT ABOUT SETTING THE WINDOW SIZE?

SETTING WINDOW SIZE

WE CAN SET THE DEFAULT WINDOW SIZE USING GEOMETRY FUNCTION LIKE THIS:

```
WINDOW.GEOMETRY('350X200')
```

THE ABOVE LINE SETS THE WINDOW WIDTH TO 350 PIXELS AND THE HEIGHT TO 200 PIXELS.

LET'S TRY ADDING MORE GUI WIDGETS LIKE BUTTONS AND SEE HOW TO HANDLE BUTTON CLICK EVENT.

ADDING A BUTTON WIDGET

LET'S START BY ADDING THE BUTTON TO THE WINDOW, THE BUTTON IS CREATED AND ADDED TO THE WINDOW THE SAME AS THE LABEL:

```
BTN = BUTTON(WINDOW, TEXT="CLICK ME")



BTN.GRID(COLUMN=1, ROW=0)
```

SO OUR WINDOW WILL BE LIKE THIS:

```
FROM TKINTER IMPORT *



WINDOW = TK()



WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")



WINDOW.GEOMETRY('350X200')
```

```
LBL = LABEL(WINDOW, TEXT="HELLO")



LBL.GRID(COLUMN=0, ROW=0)



BTN = BUTTON(WINDOW, TEXT="CLICK ME")



BTN.GRID(COLUMN=1, ROW=0)



WINDOW.MAINLOOP()
```

THE RESULT LOOKS LIKE THIS:

NOTE THAT WE PLACE THE BUTTON ON THE SECOND
COLUMN OF THE WINDOW WHICH IS 1. IF YOU FORGET AND
PLACE THE BUTTON ON THE SAME COLUMN WHICH IS 0, IT
WILL SHOW THE BUTTON ONLY, SINCE THE BUTTON WILL BE
ON THE TOP OF THE LABEL.

CHANGE BUTTON FOREGROUND AND BACKGROUND COLORS

YOU CAN CHANGE FOREGROUND FOR A BUTTON OR ANY
OTHER WIDGET USING **FG** PROPERTY.
ALSO, YOU CAN CHANGE THE BACKGROUND COLOR FOR
ANY WIDGET USING **BG** PROPERTY.

```
BTN = BUTTON(WINDOW, TEXT="CLICK ME", BG="ORANGE", FG="RED")
```

NOW, IF YOU TRIED TO CLICK ON THE BUTTON, NOTHING
HAPPENS BECAUSE THE CLICK EVENT OF THE BUTTON ISN'T
WRITTEN YET.

HANDLE BUTTON CLICK EVENT

FIRST, WE WILL WRITE THE FUNCTION THAT WE NEED TO EXECUTE WHEN THE BUTTON IS CLICKED:

```
DEF CLICKED():



    LBL.CONFIGURE(TEXT="BUTTON WAS CLICKED !!")
```

THEN WE WILL WIRE IT WITH THE BUTTON BY SPECIFYING THE FUNCTION LIKE THIS:

BTN = BUTTON(WINDOW, TEXT=**"CLICK ME"**, COMMAND=CLICKED)

NOTE THAT, WE TYPED CLICKED ONLY NOT CLICKED() WITH PARENTHESES.

NOW THE FULL CODE WILL BE LIKE THIS:

```
FROM TKINTER IMPORT *



WINDOW = TK()



WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")



WINDOW.GEOMETRY('350X200')



LBL = LABEL(WINDOW, TEXT="HELLO")



LBL.GRID(COLUMN=0, ROW=0)



DEF CLICKED():



    LBL.CONFIGURE(TEXT="BUTTON WAS CLICKED !!")
```

```
BTN = BUTTON(WINDOW, TEXT="CLICK ME", COMMAND=CLICKED)


BTN.GRID(COLUMN=1, ROW=0)


WINDOW.MAINLOOP()
```

AND WHEN WE CLICK THE BUTTON, THE RESULT AS EXPECTED:

COOL!!

GET INPUT USING ENTRY CLASS (TKINTER TEXTBOX)

IN THE PREVIOUS PYTHON GUI EXAMPLES, WE SAW HOW TO ADD SIMPLE WIDGETS, NOW LET'S TRY GETTING THE USER INPUT USING TKINTER ENTRY CLASS (TKINTER TEXTBOX). YOU CAN CREATE A TEXTBOX USING TKINTER ENTRY CLASS LIKE THIS:

```
TXT = ENTRY(WINDOW,WIDTH=10)
```

THEN YOU CAN ADD IT TO THE WINDOW USING GRID FUNCTION AS USUAL

SO OUR WINDOW WILL BE LIKE THIS:

```
FROM TKINTER IMPORT *


WINDOW = TK()


WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")
```

```
WINDOW.GEOMETRY('350X200')


LBL = LABEL(WINDOW, TEXT="HELLO")


LBL.GRID(COLUMN=0, ROW=0)


TXT = ENTRY(WINDOW,WIDTH=10)


TXT.GRID(COLUMN=1, ROW=0)


DEF CLICKED():


    LBL.CONFIGURE(TEXT="BUTTON WAS CLICKED !!")


BTN = BUTTON(WINDOW, TEXT="CLICK ME", COMMAND=CLICKED)


BTN.GRID(COLUMN=2, ROW=0)


WINDOW.MAINLOOP()
```

AND THE RESULT WILL BE LIKE THIS:

NOW, IF YOU CLICK THE BUTTON, IT WILL SHOW THE SAME OLD MESSAGE, WHAT ABOUT SHOWING THE ENTERED TEXT ON THE ENTRY WIDGET?

FIRST, YOU CAN GET ENTRY TEXT USING GET FUNCTION. SO WE CAN WRITE THIS CODE TO OUR CLICKED FUNCTION LIKE THIS:

```
DEF CLICKED():

    RES = "WELCOME TO " + TXT.GET()

    LBL.CONFIGURE(TEXT= RES)
```

IF YOU CLICK THE BUTTON AND THERE IS A TEXT ON THE ENTRY WIDGET, IT WILL SHOW "WELCOME TO" CONCATENATED WITH THE ENTERED TEXT.

AND THIS IS THE COMPLETE CODE:

```
FROM TKINTER IMPORT *

WINDOW = TK()

WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")

WINDOW.GEOMETRY('350X200')

LBL = LABEL(WINDOW, TEXT="HELLO")

LBL.GRID(COLUMN=0, ROW=0)

TXT = ENTRY(WINDOW,WIDTH=10)

TXT.GRID(COLUMN=1, ROW=0)
```

```
DEF CLICKED():


    RES = "WELCOME TO " + TXT.GET()


    LBL.CONFIGURE(TEXT= RES)



BTN = BUTTON(WINDOW, TEXT="CLICK ME", COMMAND=CLICKED)



BTN.GRID(COLUMN=2, ROW=0)



WINDOW.MAINLOOP()
```

RUN THE ABOVE CODE AND CHECK THE RESULT:

AWESOME!!

EVERY TIME WE RUN THE CODE, WE NEED TO CLICK ON THE
ENTRY WIDGET TO SET FOCUS TO WRITE THE TEXT, WHAT
ABOUT SETTING THE FOCUS AUTOMATICALLY?

SET FOCUS TO ENTRY WIDGET
THAT'S SUPER EASY, ALL WE NEED TO DO IS TO CALL
FOCUS FUNCTION LIKE THIS:

```
TXT.FOCUS()
```

AND WHEN YOU RUN YOUR CODE, YOU WILL NOTICE THAT
THE ENTRY WIDGET HAS THE FOCUS SO YOU CAN WRITE
YOUR TEXT RIGHT AWAY.

DISABLE ENTRY WIDGET

TO DISABLE ENTRY WIDGET, YOU CAN SET THE STATE PROPERTY TO DISABLED:

```
TXT = ENTRY(WINDOW,WIDTH=10, STATE='DISABLED')
```

NOW, YOU WON'T BE ABLE TO ENTER ANY TEXT.

ADD A COMBOBOX WIDGET

TO ADD A COMBOBOX WIDGET, YOU CAN USE THE COMBOBOX CLASS FROM TTK LIBRARY LIKE THIS:

```
FROM TKINTER.TTK IMPORT *

COMBO = COMBOBOX(WINDOW)
```

THEN YOU CAN ADD YOUR VALUES TO THE COMBOBOX.

```
FROM TKINTER IMPORT *

FROM TKINTER.TTK IMPORT *

WINDOW = TK()

WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")

WINDOW.GEOMETRY('350X200')

COMBO = COMBOBOX(WINDOW)
```

```
COMBO['VALUES']= (1, 2, 3, 4, 5, "TEXT")



COMBO.CURRENT(1) #SET THE SELECTED ITEM



COMBO.GRID(COLUMN=0, ROW=0)



WINDOW.MAINLOOP()
```

AS YOU CAN SEE, WE ADD THE COMBOBOX ITEMS USING THE TUPLE.

TO SET THE SELECTED ITEM, YOU CAN PASS THE INDEX OF THE DESIRED ITEM TO THE CURRENT FUNCTION.
TO GET THE SELECT ITEM, YOU CAN USE THE GET FUNCTION LIKE THIS:

```
COMBO.GET()
```

ADD A CHECKBUTTON WIDGET (TKINTER CHECKBOX)

TO CREATE A CHECKBUTTON WIDGET, YOU CAN USE CHECKBUTTON CLASS LIKE THIS:

```
CHK = CHECKBUTTON(WINDOW, TEXT='CHOOSE')
```

ALSO, YOU CAN SET THE CHECKED STATE BY PASSING THE CHECK VALUE TO THE CHECKBUTTON LIKE THIS:

```
FROM TKINTER IMPORT *



FROM TKINTER.TTK IMPORT *
```

```
WINDOW = TK()


WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")


WINDOW.GEOMETRY('350X200')


CHK_STATE = BOOLEANVAR()


CHK_STATE.SET(TRUE) #SET CHECK STATE


CHK = CHECKBUTTON(WINDOW, TEXT='CHOOSE', VAR=CHK_STATE)


CHK.GRID(COLUMN=0, ROW=0)


WINDOW.MAINLOOP()
```

CHECK THE RESULT:

SET CHECK STATE OF A CHECKBUTTON

HERE WE CREATE A VARIABLE OF TYPE BOOLEANVAR WHICH IS NOT A STANDARD PYTHON VARIABLE, IT'S A TKINTER VARIABLE, AND THEN WE PASS IT TO THE CHECKBUTTON CLASS TO SET THE CHECK STATE AS THE HIGHLIGHTED LINE IN THE ABOVE EXAMPLE.

YOU CAN SET THE BOOLEAN VALUE TO FALSE TO MAKE IT UNCHECKED.

ALSO, YOU CAN USE INTVAR INSTEAD OF BOOLEANVAR AND SET THE VALUE TO 0 OR 1.

```
CHK_STATE = INTVAR()



CHK_STATE.SET(0) #UNCHECK



CHK_STATE.SET(1) #CHECK
```

THESE EXAMPLES GIVE THE SAME RESULT AS THE BOOLEANVAR.

ADD RADIO BUTTONS WIDGETS

TO ADD RADIO BUTTONS, SIMPLY YOU CAN USE RADIOBUTTON CLASS LIKE THIS:

```
RAD1 = RADIOBUTTON(WINDOW,TEXT='FIRST', VALUE=1)
```

NOTE THAT YOU SHOULD SET THE VALUE FOR EVERY RADIO BUTTON WITH A DIFFERENT VALUE, OTHERWISE, THEY WON'T WORK.

```
FROM TKINTER IMPORT *



FROM TKINTER.TTK IMPORT *



WINDOW = TK()



WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")



WINDOW.GEOMETRY('350X200')



RAD1 = RADIOBUTTON(WINDOW,TEXT='FIRST', VALUE=1)
```

```
RAD2 = RADIOBUTTON(WINDOW,TEXT='SECOND', VALUE=2)


RAD3 = RADIOBUTTON(WINDOW,TEXT='THIRD', VALUE=3)


RAD1.GRID(COLUMN=0, ROW=0)


RAD2.GRID(COLUMN=1, ROW=0)


RAD3.GRID(COLUMN=2, ROW=0)


WINDOW.MAINLOOP()
```

THE RESULT OF THE ABOVE CODE LOOKS LIKE THIS:

ALSO, YOU CAN SET THE COMMAND OF ANY OF THESE RADIO BUTTONS TO A SPECIFIC FUNCTION, SO IF THE USER CLICKS ON ANY ONE OF THEM, IT RUNS THE FUNCTION CODE.

THIS IS AN EXAMPLE:

```
RAD1 = RADIOBUTTON(WINDOW,TEXT='FIRST', VALUE=1,
COMMAND=CLICKED)


DEF CLICKED():


# DO WHAT YOU NEED
```

PRETTY SIMPLE!!

GET RADIO BUTTON VALUE (SELECTED RADIO BUTTON)

TO GET THE CURRENTLY SELECTED RADIO BUTTON OR THE RADIO BUTTON VALUE, YOU CAN PASS THE VARIABLE PARAMETER TO THE RADIO BUTTONS AND LATER YOU CAN GET ITS VALUE.

```
FROM TKINTER IMPORT *


FROM TKINTER.TTK IMPORT *


WINDOW = TK()


WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")


SELECTED = INTVAR()


RAD1 = RADIOBUTTON(WINDOW,TEXT='FIRST', VALUE=1,
VARIABLE=SELECTED)


RAD2 = RADIOBUTTON(WINDOW,TEXT='SECOND', VALUE=2,
VARIABLE=SELECTED)


RAD3 = RADIOBUTTON(WINDOW,TEXT='THIRD', VALUE=3,
VARIABLE=SELECTED)


DEF CLICKED():


    PRINT(SELECTED.GET())
```

```
BTN = BUTTON(WINDOW, TEXT="CLICK ME", COMMAND=CLICKED)


RAD1.GRID(COLUMN=0, ROW=0)


RAD2.GRID(COLUMN=1, ROW=0)


RAD3.GRID(COLUMN=2, ROW=0)


BTN.GRID(COLUMN=3, ROW=0)



WINDOW.MAINLOOP()
```

EVERY TIME YOU SELECT A RADIO BUTTON, THE VALUE OF THE VARIABLE WILL BE CHANGED TO THE VALUE OF THE SELECTED RADIO BUTTON.

ADD A SCROLLEDTEXT WIDGET (TKINTER TEXTAREA)

TO ADD A SCROLLEDTEXT WIDGET, YOU CAN USE THE SCROLLEDTEXT CLASS LIKE THIS:

```
FROM TKINTER IMPORT SCROLLEDTEXT



TXT = SCROLLEDTEXT.SCROLLEDTEXT(WINDOW,WIDTH=40,HEIGHT=10)
```

HERE WE SPECIFY THE WIDTH AND THE HEIGHT OF THE SCROLLEDTEXT WIDGET, OTHERWISE, IT WILL FILL THE ENTIRE WINDOW.

```
FROM TKINTER IMPORT *
```

```
FROM TKINTER IMPORT SCROLLEDTEXT


WINDOW = TK()


WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")


WINDOW.GEOMETRY('350X200')


TXT = SCROLLEDTEXT.SCROLLEDTEXT(WINDOW,WIDTH=40,HEIGHT=10)


TXT.GRID(COLUMN=0,ROW=0)



WINDOW.MAINLOOP()
```

THE RESULT AS YOU CAN SEE:

SET SCROLLEDTEXT CONTENT
TO SET SCROLLEDTEXT CONTENT, YOU CAN USE THE
INSERT METHOD LIKE THIS:

```
TXT.INSERT(INSERT,'YOU TEXT GOES HERE')
```

DELETE/CLEAR SCROLLEDTEXT CONTENT
TO CLEAR THE CONTENTS OF A SCROLLEDTEXT WIDGET,
YOU CAN USE DELETE METHOD LIKE THIS:

```
TXT.DELETE(1.0,END)
```

GREAT!!

CREATE A MESSAGEBOX

TO SHOW A MESSAGE BOX USING TKINTER, YOU CAN USE MESSAGEBOX LIBRARY LIKE THIS:

```
FROM TKINTER IMPORT MESSAGEBOX


MESSAGEBOX.SHOWINFO('MESSAGE TITLE','MESSAGE CONTENT')
```

PRETTY EASY!!

LET'S SHOW A MESSAGE BOX WHEN THE USER CLICKS A BUTTON.

```
FROM TKINTER IMPORT *


FROM TKINTER IMPORT MESSAGEBOX


WINDOW = TK()


WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")


WINDOW.GEOMETRY('350X200')


DEF CLICKED():


    MESSAGEBOX.SHOWINFO('MESSAGE TITLE', 'MESSAGE CONTENT')


BTN = BUTTON(WINDOW,TEXT='CLICK HERE', COMMAND=CLICKED)


BTN.GRID(COLUMN=0,ROW=0)
```

```
WINDOW.MAINLOOP()
```

WHEN YOU CLICK THE BUTTON, AN INFO MESSAGEBOX WILL
APPEAR.

SHOW WARNING AND ERROR MESSAGES

YOU CAN SHOW A WARNING MESSAGE OR ERROR MESSAGE
THE SAME WAY. THE ONLY THING THAT NEEDS TO BE
CHANGED IS THE MESSAGE FUNCTION

```
MESSAGEBOX.SHOWWARNING('MESSAGE TITLE', 'MESSAGE
CONTENT')  #SHOWS WARNING MESSAGE



MESSAGEBOX.SHOWERROR('MESSAGE TITLE', 'MESSAGE
CONTENT')    #SHOWS ERROR MESSAGE
```

SHOW ASKQUESTION DIALOGS

TO SHOW A YES NO MESSAGE BOX TO THE USER, YOU CAN
USE ONE OF THE FOLLOWING MESSAGEBOX FUNCTIONS:

```
FROM TKINTER IMPORT MESSAGEBOX



RES = MESSAGEBOX.ASKQUESTION('MESSAGE TITLE','MESSAGE CONTENT')



RES = MESSAGEBOX.ASKYESNO('MESSAGE TITLE','MESSAGE CONTENT')



RES = MESSAGEBOX.ASKYESNOCANCEL('MESSAGE TITLE','MESSAGE
CONTENT')



RES = MESSAGEBOX.ASKOKCANCEL('MESSAGE TITLE','MESSAGE CONTENT')
```

```
RES = MESSAGEBOX.ASKRETRYCANCEL('MESSAGE TITLE','MESSAGE
CONTENT')
```

YOU CAN CHOOSE THE APPROPRIATE MESSAGE STYLE ACCORDING TO YOUR NEEDS. JUST REPLACE THE SHOWINFO FUNCTION LINE FROM THE PREVIOUS LINE AND RUN IT.

ALSO, YOU CAN CHECK WHAT BUTTON WAS CLICKED USING THE RESULT VARIABLE

IF YOU CLICK **OK** OR **YES** OR **RETRY**, IT WILL RETURN **TRUE** VALUE, BUT IF YOU CHOOSE **NO** OR **CANCEL**, IT WILL RETURN **FALSE**.
THE ONLY FUNCTION THAT RETURNS ONE OF THREE VALUES IS **ASKYESNOCANCEL** FUNCTION, IT RETURNS **TRUE OR FALSE OR NONE**.

ADD A SPINBOX (NUMBERS WIDGET)
TO CREATE A SPINBOX WIDGET, YOU CAN USE SPINBOX CLASS LIKE THIS:

```
SPIN = SPINBOX(WINDOW, FROM_=0, TO=100)
```

HERE WE CREATE A SPINBOX WIDGET AND WE PASS THE FROM_ AND TO PARAMETERS TO SPECIFY THE NUMBERS RANGE FOR THE SPINBOX.

ALSO, YOU CAN SPECIFY THE WIDTH OF THE WIDGET USING THE WIDTH PARAMETER:

```
SPIN = SPINBOX(WINDOW, FROM_=0, TO=100, WIDTH=5)
```

CHECK THE COMPLETE EXAMPLE:

```
FROM TKINTER IMPORT *




WINDOW = TK()




WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")
```

```
WINDOW.GEOMETRY('350X200')



SPIN = SPINBOX(WINDOW, FROM_=0, TO=100, WIDTH=5)



SPIN.GRID(COLUMN=0,ROW=0)



WINDOW.MAINLOOP()
```

YOU CAN SPECIFY THE NUMBERS FOR THE SPINBOX INSTEAD OF USING THE WHOLE RANGE LIKE THIS:

```
SPIN = SPINBOX(WINDOW, VALUES=(3, 8, 11), WIDTH=5)
```

HERE THE SPINBOX WIDGET ONLY SHOWS THESE 3 NUMBERS ONLY 3, 8 AND 11.

SET DEFAULT VALUE FOR SPINBOX

TO SET THE SPINBOX DEFAULT VALUE, YOU CAN PASS THE VALUE TO THE TEXTVARIABLE PARAMETER LIKE THIS:

```
VAR =INTVAR()



VAR.SET(36)



SPIN = SPINBOX(WINDOW, FROM_=0, TO=100, WIDTH=5,
TEXTVARIABLE=VAR)
```

NOW, IF YOU RUN THE PROGRAM, IT WILL SHOW 36 AS A DEFAULT VALUE FOR THE SPINBOX.

ADD A PROGRESSBAR WIDGET

TO CREATE A PROGRESS BAR, YOU CAN USE THE PROGRESSBAR CLASS LIKE THIS:

```
FROM TKINTER.TTK IMPORT PROGRESSBAR



BAR = PROGRESSBAR(WINDOW, LENGTH=200)
```

YOU CAN SET THE PROGRESS BAR VALUE LIKE THIS:

```
BAR['VALUE'] = 70
```

YOU CAN SET THIS VALUE BASED ON ANY PROCESS YOU WANT LIKE DOWNLOADING A FILE OR COMPLETING A TASK.

CHANGE PROGRESSBAR COLOR

CHANGING PROGRESSBAR COLOR IS A BIT TRICKY, BUT SUPER EASY.

FIRST, WE WILL CREATE A STYLE AND SET THE BACKGROUND COLOR AND FINALLY SET THE CREATED STYLE TO THE PROGRESSBAR.

CHECK THE FOLLOWING EXAMPLE:

```
FROM TKINTER IMPORT *



FROM TKINTER.TTK IMPORT PROGRESSBAR



FROM TKINTER IMPORT TTK



WINDOW = TK()



WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")
```

```
WINDOW.GEOMETRY('350X200')



STYLE = TTK.STYLE()



STYLE.THEME_USE('DEFAULT')



STYLE.CONFIGURE("BLACK.HORIZONTAL.TPROGRESSBAR",
BACKGROUND='BLACK')



BAR = PROGRESSBAR(WINDOW, LENGTH=200,
STYLE='BLACK.HORIZONTAL.TPROGRESSBAR')



BAR['VALUE'] = 70



BAR.GRID(COLUMN=0, ROW=0)



WINDOW.MAINLOOP()
```

AND THE RESULT WILL BE LIKE THIS:

ADD A FILEDIALOG (FILE & DIRECTORY CHOOSER)

TO CREATE A FILE DIALOG (FILE CHOOSER), YOU CAN USE THE FILEDIALOG CLASS LIKE THIS:

```
FROM TKINTER IMPORT FILEDIALOG



FILE = FILEDIALOG.ASKOPENFILENAME()
```

AFTER YOU CHOOSE A FILE AND CLICK OPEN, THE FILE VARIABLE WILL HOLD THAT FILE PATH.

ALSO, YOU CAN ASK FOR MULTIPLE FILES LIKE THIS:

```
FILES = FILEDIALOG.ASKOPENFILENAMES()
```

SPECIFY FILE TYPES (FILTER FILE EXTENSIONS)

YOU CAN SPECIFY THE FILE TYPES FOR A FILE DIALOG USING FILETYPES PARAMETER, JUST SPECIFY THE EXTENSIONS IN TUPLES.

```
FILE = FILEDIALOG.ASKOPENFILENAME(FILETYPES = (("TEXT
FILES","*.TXT"),("ALL FILES","*.*")))
```

YOU CAN ASK FOR A DIRECTORY USING ASKDIRECTORY METHOD:

```
DIR = FILEDIALOG.ASKDIRECTORY()
```

YOU CAN SPECIFY THE INITIAL DIRECTORY FOR THE FILE DIALOG BY SPECIFYING THE INITIALDIR LIKE THIS:

```
FROM OS IMPORT PATH



FILE = FILEDIALOG.ASKOPENFILENAME(INITIALDIR=
PATH.DIRNAME(__FILE__))
```

EASY!!

ADD A MENU BAR

TO ADD A MENU BAR, YOU CAN USE MENU CLASS LIKE THIS:

```
FROM TKINTER IMPORT MENU



MENU = MENU(WINDOW)
```

```
MENU.ADD_COMMAND(LABEL='FILE')



WINDOW.CONFIG(MENU=MENU)
```

FIRST, WE CREATE A MENU, THEN WE ADD OUR FIRST LABEL, AND FINALLY, WE ASSIGN THE MENU TO OUR WINDOW.

YOU CAN ADD MENU ITEMS UNDER ANY MENU BY USING ADD_CASCADE() FUNCTION LIKE THIS:

```
MENU.ADD_CASCADE(LABEL='FILE', MENU=NEW_ITEM)
```

SO OUR CODE WILL BE LIKE THIS:

```
FROM TKINTER IMPORT *


FROM TKINTER IMPORT MENU


WINDOW = TK()


WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")


MENU = MENU(WINDOW)


NEW_ITEM = MENU(MENU)


NEW_ITEM.ADD_COMMAND(LABEL='NEW')


MENU.ADD_CASCADE(LABEL='FILE', MENU=NEW_ITEM)
```

```
WINDOW.CONFIG(MENU=MENU)



WINDOW.MAINLOOP()
```

USING THIS WAY, YOU CAN ADD MANY MENU ITEMS AS YOU WANT.

```
FROM TKINTER IMPORT *


FROM TKINTER IMPORT MENU


WINDOW = TK()


WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")


MENU = MENU(WINDOW)


NEW_ITEM = MENU(MENU)


NEW_ITEM.ADD_COMMAND(LABEL='NEW')


NEW_ITEM.ADD_SEPARATOR()


NEW_ITEM.ADD_COMMAND(LABEL='EDIT')


MENU.ADD_CASCADE(LABEL='FILE', MENU=NEW_ITEM)
```

```
WINDOW.CONFIG(MENU=MENU)




WINDOW.MAINLOOP()
```

HERE WE ADD ANOTHER MENU ITEM CALLED EDIT WITH A MENU SEPARATOR.

YOU MAY NOTICE A DASHED LINE AT THE BEGINNING, WELL, IF YOU CLICK THAT LINE, IT WILL SHOW THE MENU ITEMS IN A SMALL SEPARATE WINDOW.

YOU CAN DISABLE THIS FEATURE BY DISABLING THE TEAROFF FEATURE LIKE THIS:

```
NEW_ITEM = MENU(MENU, TEAROFF=0)
```

JUST REPLACE THE NEW_ITEM IN THE ABOVE EXAMPLE WITH THIS ONE AND IT WON'T SHOW THE DASHED LINE ANYMORE.

I DON'T NEED TO REMIND YOU THAT YOU CAN TYPE ANY CODE THAT WORKS WHEN THE USER CLICKS ON ANY MENU ITEM BY SPECIFYING THE COMMAND PROPERTY.

```
NEW_ITEM.ADD_COMMAND(LABEL='NEW', COMMAND=CLICKED)
```

ADD A NOTEBOOK WIDGET (TAB CONTROL)
TO CREATE A TAB CONTROL, THERE ARE 3 STEPS TO DO SO.

- FIRST, WE CREATE A TAB CONTROL USING NOTEBOOK CLASS
- CREATE A TAB USING FRAME CLASS.
- ADD THAT TAB TO THE TAB CONTROL.

- PACK THE TAB CONTROL SO IT BECOMES VISIBLE IN THE WINDOW.

```
FROM TKINTER IMPORT *


FROM TKINTER IMPORT TTK


WINDOW = TK()


WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")


TAB_CONTROL = TTK.NOTEBOOK(WINDOW)


TAB1 = TTK.FRAME(TAB_CONTROL)


TAB_CONTROL.ADD(TAB1, TEXT='FIRST')


TAB_CONTROL.PACK(EXPAND=1, FILL='BOTH')


WINDOW.MAINLOOP()
```

YOU CAN ADD MANY TABS AS YOU WANT THE SAME WAY.

ADD WIDGETS TO NOTEBOOKS

AFTER CREATING TABS, YOU CAN PUT WIDGETS INSIDE THESE TABS BY ASSIGNING THE PARENT PROPERTY TO THE DESIRED TAB.

```
FROM TKINTER IMPORT *
```

```
FROM TKINTER IMPORT TTK

WINDOW = TK()

WINDOW.TITLE("WELCOME TO LIKEGEEKS APP")

TAB_CONTROL = TTK.NOTEBOOK(WINDOW)

TAB1 = TTK.FRAME(TAB_CONTROL)

TAB2 = TTK.FRAME(TAB_CONTROL)

TAB_CONTROL.ADD(TAB1, TEXT='FIRST')

TAB_CONTROL.ADD(TAB2, TEXT='SECOND')

LBL1 = LABEL(TAB1, TEXT= 'LABEL1')

LBL1.GRID(COLUMN=0, ROW=0)

LBL2 = LABEL(TAB2, TEXT= 'LABEL2')

LBL2.GRID(COLUMN=0, ROW=0)

TAB_CONTROL.PACK(EXPAND=1, FILL='BOTH')
```

```
WINDOW.MAINLOOP()
```

ADD SPACING FOR WIDGETS (PADDING)

YOU CAN ADD PADDING FOR YOUR CONTROLS TO MAKE IT LOOKS WELL ORGANIZED USING PADX AND PADY PROPERTIES.

JUST PASS PADX AND PADY TO ANY WIDGET AND GIVE THEM A VALUE.

```
LBL1 = LABEL(TAB1, TEXT= 'LABEL1', PADX=5, PADY=5)
```