Tkinter GUI

Brief Description

- A `GUI` (graphical user interface) makes the programme easier to use. It allows you, to create screens, text boxes and buttons to help the user navigate through the programme in a more user-friendly way.
- `Tkinter` is a library of features in Python that allows you to create a GUI

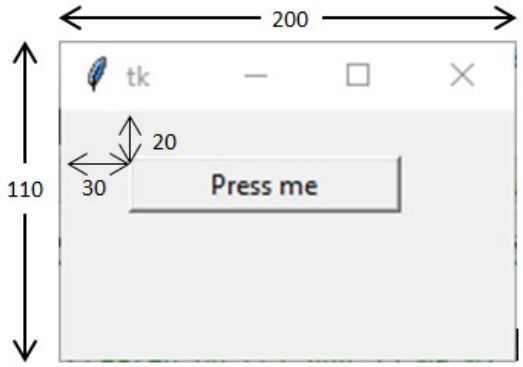
Tkinter Widgets

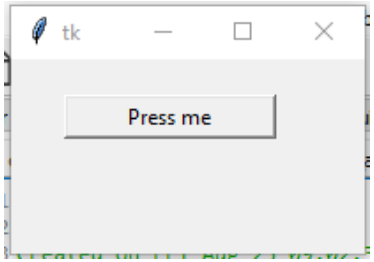
- **`Button:`** – **`Button`** widget is used to place the buttons in the **`tkinter`**.
- **`Canvas:`** – **`Canvas`** is used to draw shapes in your **GUI**.
- **`Checkbutton:`** – **`Checkbutton`** is used to create the check buttons in your application. You can select more than one option at a time.
- **`Entry:`** – **`Entry`** widget is used to create input fields in the **GUI**.
- **`Frame:`** – **`Frame`** is used as containers in the **tkinter**.
- **`Label:`** – **`Label`** is used to create a single line widgets like **text**, **images**, etc..,
- **`Menu:`** – **`Menu`** is used to create menus in the **GUI**.

Geometry and Place

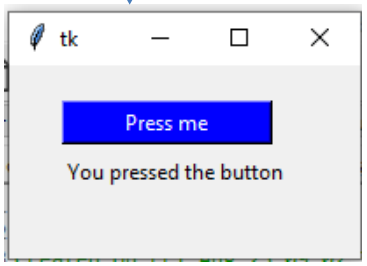
- Look at the code and in particular the measurements that are used in the `window.geometry` and `button.place` lines.
- The `geometry` line in the code determines the size of the window
- The `place` line in the code determines the position of the individual item on the window
- Now look at the window that the code produces

```
from tkinter import *


def Call():
    msg = Label(window, text = "You pressed the button")
    msg.place(x = 30, y = 50)
    button["bg"] = "blue"
    button["fg"] = "white"

window = Tk()
window.geometry("200x110")
button = Button(text = "Press me", command = Call)
button.place(x = 30, y = 20, width=120, height=25)
window.mainloop()
```





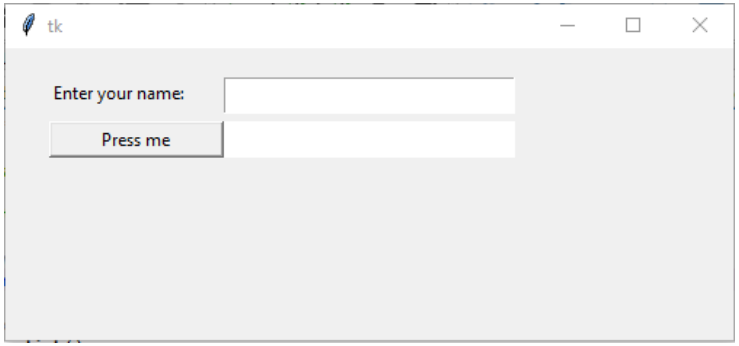
Once the button is pressed, it will run the "Call" function and change the window to look like this

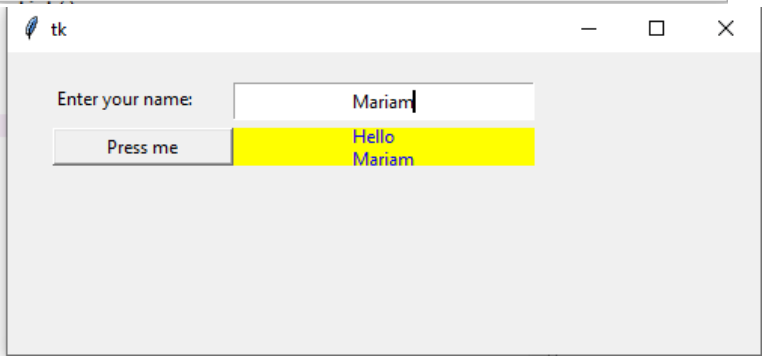


```
# Import thie tkinter function
from tkinter import *

# Define a functing called click
def click():
    #Save the content of the text box and store it in a variable called name
    name = textbox1.get()
    # get the name that was entered in textbox1 and add it to create
    # this string that is saved in variable called message
    message = str("Hello " + name)
    textbox2["bg"] = "yellow" # Background colour
    textbox2["fg"] = "red" # font colour
    # Change the content of the textbox to display
    #the value of the variable message
    textbox2["text"] = message

# Create a window that will act as a display. It is referred to as "window"
window = Tk()
# Define the size
window.geometry("500x200")

# Add text to the screen displaying the message
label1 = Label(text = "Enter your name:")
# position the label
label1.place(x = 30, y = 20)

# Create a blank text box or entry box. These boxes can be used by the user
# to input data or used to display output
textbox1 = Entry(text = "")
# Specify the position in the window. If the position is not specified,
# the item will not appear in the window
textbox1.place(x = 150, y = 20, width = 200, height = 25)
# Specify the justification
textbox1["justify"] = "center"
textbox1.focus()

# Create a button that will run the function "click"
button1 = Button(text = "Press me", command = click)
# Specify the position of the button in the window.
button1.place(x =30, y = 50, width = 120, height = 25)

# Create a message box which is used to display and output
textbox2 = Message(text = "")
# Specify the position in the window. If the position is not specified,
# the item will not appear in the window
textbox2.place(x = 150, y = 50, width = 200, height = 25)
# Specify the background colour
textbox2["bg"] = "white"
# Specify the font colour
textbox2["fg"] = "black"

# Now make it working with this code
window.mainloop()
```

- This creates a window that asks the user to enter their name. When they click on a button it should display the message “Hello” and their name and change the background and font colour of the message box

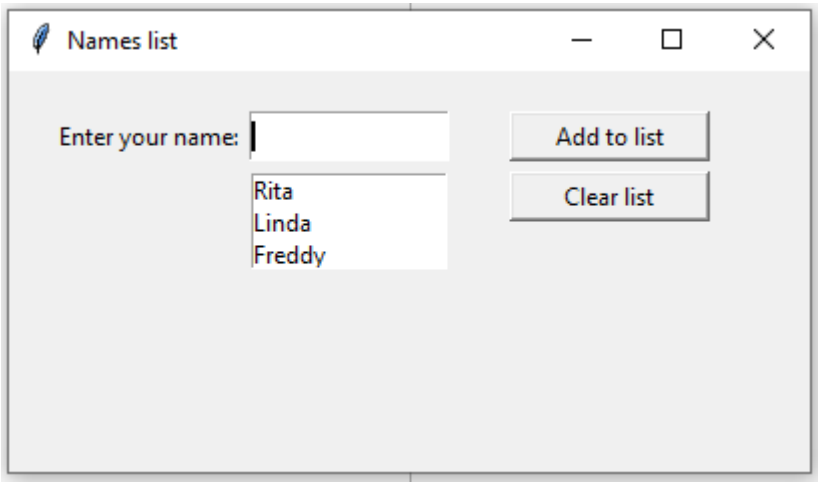
The comments in the code explain what each line of code does.





Basic Tkinter GUI – more example

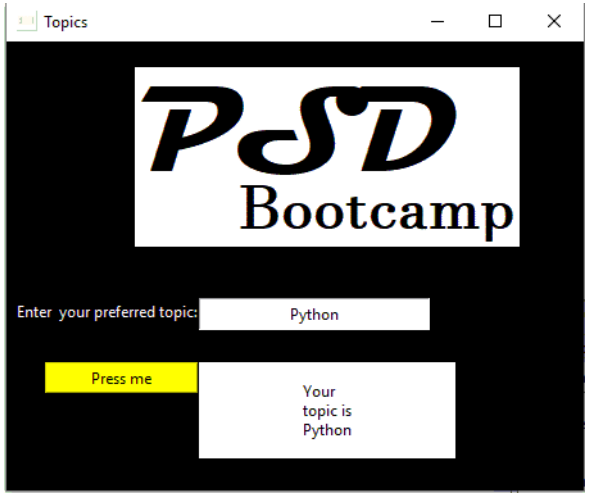
```
# Define a function called click
def add_name():
    name = name_box.get()
    name_list.insert(END, name)
    # Delete the content of an entry or  list box
    name_box.delete(0, END)
    name_box.focus()

def clear_list():
    name_list.delete(0, END)
    name_box.focus()

window = Tk()
window.title("Names list")
window.geometry("400x200")

label1 = Label(text = "Enter your name:")
label1.place(x = 20, y = 20, width = 100, height = 25)

name_box = Entry(text = 0)
name_box.place(x = 120, y = 20, width = 100, height = 25)
name_box.focus()

button1 = Button(text = "Add to list", command = add_name)
button1.place(x = 250, y = 20, width = 100, height = 25)

name_list = Listbox() # Create a drop-down list box
name_list.place(x = 120, y = 50, width = 100, height = 50)

button2 = Button(text = "Clear list", command = clear_list)
button2.place(x = 250, y = 50, width = 100, height = 25)

# This mus be at the end of the programme to make sure it keeps working.
window.mainloop()
```



This programme creates a window that asks the user to enter a name in a text box. When they click on a button, it adds it to the end of the list that is displayed on the screen. There is another button that clears the list.

More Tkinter

When using images in your programme, it is easier if they are stored in the same folder as the programme. Otherwise you need to include the entire location of the files.



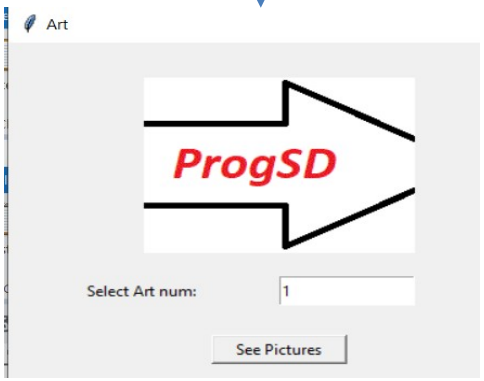

```
# Display an image in a label widget.
# The image will change while the program is running
#  photobox.image makes it changable, updatable
logo = PhotoImage(file = "logo.gif")
logoimage = Label(window, image = logo)
photobox.image = photo
logoimage.place(x = 100, y = 20, width = 250, height = 140)
```

```
from tkinter import *

def click():
    name = textbox1.get()
    message = str("Your topic is " + name)
    textbox2["text"] = message

window = Tk()
window.title("Topics")
window.geometry("450x350")
# Chnages the icon displayed in the title of the window
window.wm_iconbitmap("stripes.ico")
# chnage the background colour to black
window.configure(background = "black")

# Display an image in a label widget.
# The image will not chnage while the program is running
logo = PhotoImage(file = "logo.gif")
logoimage = Label(image = logo)
logoimage.place(x = 100, y = 20, width = 250, height = 140)

label1 = Label(text = "Enter  your preferred topic:")
label1.place(x = 5, y = 200)
label1["bg"]= "black"
label1["fg"] = "white"

textbox1 = Entry(text ="")
textbox1.place(x = 150, y = 200, width = 180, height = 25)
textbox1["justify"] = "center"
textbox1.focus()

button1 = Button(text = "Press me", command = click)
button1.place(x = 30, y = 250, width = 120, height = 25)
button1["bg"] = "yellow"

textbox2 = Message(text = "")
textbox2.place(x = 150, y = 250, width = 200, height = 75)
textbox2["bg"] = "white"
textbox2["fg"] = "black"
window.mainloop()
```

See next slide for a complete example of a program where images are changed while the program is running,

More Tkinter

Outputs




```
from tkinter import *
def Pict_clicked():
    ''' Variable numb will store the number the user enters. '''
    numb = selection.get()
    '''
    I have already created a number of pictures that I have
    save as 1.gif, 2.gif, etc... note that all should be .gif.
    So the user's choice (imgchoice) will be a gif image starting
    with the number selected above
    '''.
    imgchoice = numb + ".gif"
    '''The chosen image is stored in a variabl called photo.
    PhotoImage class is used to display images in labels,
    buttons, canvases, and text widget;only GIF and PGM/PPM
    image formats are supported'''
    photo = PhotoImage(file = imgchoice)
    ''' This next code ensures that we change the image while to the code is runn
    it means that as the user changes his/choice by providing a different
    number each time, a different image is shown '''
    photobox.image = photo
    photobox["image"] = photo
    photobox.update()
window = Tk()
window.title("Art")
window.geometry("500x450")

art = PhotoImage(file ='1.gif')
photobox = Label(window, image = art)
photobox.image = art
photobox.place( x =100, y = 30, width = 200, height = 150)

label = Label(text = "Select Art num: ")
label.place(x = 50, y = 200, width = 100, height = 25)

selection = Entry(text = "")
selection.place(x = 200, y = 200, width = 100, height = 25)
selection.focus()

button = Button(text = "See Pictures", command = Pict_clicked)
button.place(x = 150, y = 250, width = 100, height = 25)

window.mainloop()
```