Python Tkinter Tutorial

Python GUI

To develop GUI application in Python, there are multiple options in terms of python packages. The most generally used package is **tkinter**.

In this Python GUI Tutorial, we will use tkinter to learn how to develop GUI applications. You may be wondering on why we are using tkinter. The answer is quite simple. There is a large tkinter community online that can help you, through forums and other websites.

While tkinter provides the widgets with the all the functionality and behavior aspects, there is another module named `tkinter.ttk` which provides themed widget set.

Getting Started with Tkinter

Tkinter is an inbuilt python package. You can import the package and start using the package functions and classes.

```
import tkinter as tk
```

or you can use the other variation of importing the package

```
from tkinter import *
```

Create a Simple GUI Window

To create a GUI Window, tkinter provides Tk() class. The syntax of Tk() class is:

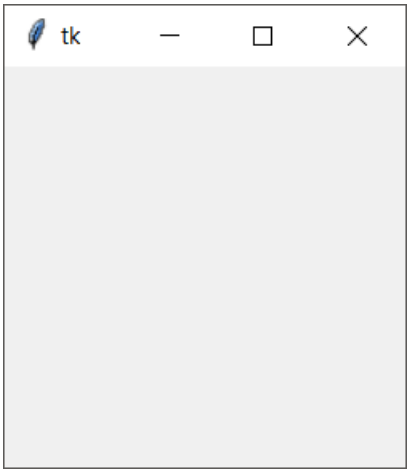
```
Tk(screenName=None,  baseName=None,  className='Tk',  useTk=1)
```

Following is a simple example to create a GUI Window.

example.py – Python Program

```
import tkinter as tk

main_window = tk.Tk()
main_window.mainloop()
```

Output

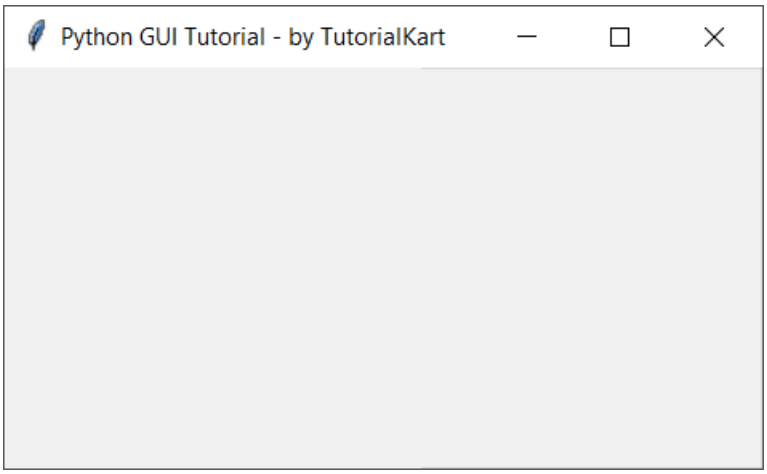


You can change the title of the window by using title function on the root or main window widget.

example.py – Python Program

```
import tkinter as tk

main_window = tk.Tk()
main_window.title('Python GUI Tutorial - by TutorialKart')
main_window.mainloop()
```

Output



Python GUI Widgets

You can add widgets into the window. Also note that there are a wide variety of widgets you can use from tkinter. In this Tkinter Tutorial, we will cover all these widgets. Following are the list of Tkinter widgets.

- Button
- Canvas
- Checkbutton
- Radiobutton
- Entry
- Frame
- Label
- Listbox

- Menu
- MenuButton
- Message
- Scale
- Scrollbar
- Text
- TopLevel
- SpinBox
- PannedWindow

After creating a GUI window using Tk() and before calling the mainloop() function on the window, you can add as many widgets as required.

```
from tkinter import *

gui = Tk()
# add widgets here
gui.mainloop()
```

Example – Tkinter Button Widget

To add a button to the Python Window, use the following syntax

```
button = Button(master, option=value)
 button.pack()
```

where master is the window to which you would like to add this button, and you may provide different options to the button constructor. The options available for button are:

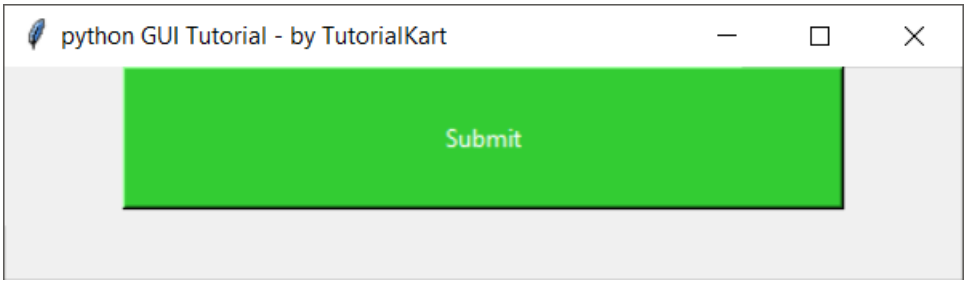
Option	Description
activebackground	button's background color when button is under the cursor
activeforeground	button's foreground color when button is under the cursor
bg	background color of button
command	function to be called on click
font	font on the button label
image	image on the button
width	width of the button
height	height of the button
text	text of the button label

In this example, we will create a simple button with values provided for some of the options,

Python Program

```
from tkinter import *

# create gui window
gui = Tk(className='Python GUI Tutorial - by TutorialKart')

#widgets start

button = Button(gui, text='Submit', width=50, height=4, bg='#33CC33', fg='#FFFFFF', activeback
button.pack()

#widgets end

gui.mainloop()
```

Output



Tkinter Problems – Solved

While working with Tkinter, you may come across some of the following issues.

- Python Tkinter Frame Width Height Not Working

Conclusion

In this Python Tutorial, we learned about Tkinter library and the widgets it provides to build a GUI application in Python.

Python Programming

- Python Tutorial
- Install Python
- Install Anaconda Python
- Python HelloWorld Program
- Python Variables
- Python Variable Data Type Conversion
- Python Comments

Control Statements

- Python If
- Python If Else
- Python While Loop
- Python For Loop

Python String

- Python String Methods
- Python String Length
- Python String Replace
- Python Split String
- Python Count Occurrences of Sub-String
- Python Sort List of Strings

Functions

- Python Functions

Python Collections

- Python List
- Python Dictionary

Advanced

- Python Multithreading

Useful Resources

- Python Interview Questions