

ttk Module နှင့် Label တစ်ခု, Button တစ်ခု, Function တစ်ခု ပါဝင်သည့် လေ့ကျင့်ခန်း

လေ့လာခါစသူများအတွက် တစ်ခါတစ်ရံ ttk ဆိုသည့် စာလုံးကို တွေ့မြင် ကြလိမ့်မည်။ ttk ဆိုသည်မှာ themed tk ၏ အတိုခေါက်ဖြစ်သည်။ (ttk stands for 'themed tk'.) ဥပမာ

```
ttk.Label(root, text="A Label").grid(column=0, row=0)
```

ttk သည် tkinter ထဲတွင်ပါဝင်သည့် ပိုကောင်းအောင် ပြုလုပ်ထားသည့် သီးသန့် module တစ်ခု ဖြစ်သည်။ ttk module တွင် advanced widget များ ပါဝင်သောကြောင့် GUI များ ပိုကောင်းလာသည်။ ပိုလှပလာပြီး ကြည့်လိုကောင်းလာသည်။ ttk module ကို tkinter ၏ extension အဖြစ်လည်း မှတ်ယူနိုင်သည်။ ttk module ကို အသုံးပြုရန်အတွက် import လုပ်ရန် လိုအပ်သည်။

```
import tkinter as tk
from tkinter import ttk
```

လေ့ကျင့်ခန်း

GUI တစ်ခုတွင် root window တစ်ခု ပါဝင်သည်။ ထို root window ထဲတွင် label တစ်ခု နှင့် button တစ်ခု ပါဝင်သည်။ button ကို click လိုက်လျှင် label ၏ အရောင် ပြောင်းသွားမည့် လေ့ကျင့်ခန်း တစ်ခုကို ရှင်းပြထားသည်။

အဆင့်(၁) root window တစ်ခု တည်ဆောက်မည်။

အဆင့်(၂) root window ထဲတွင် my_label ဆိုသည့် label ထည့်မည်။

အဆင့်(၃) root window ထဲတွင် action_button ဆိုသည့် label ထည့်မည်။

အဆင့်(၄) action_button ကို click လိုက်လျှင် အရောင် ပြောင်းသွားစေရန်အတွက် clickMe() ဆိုသည့် function တစ်ခုကို ရေးရမည်။

အဆင့်(၅) မိမိ လိုချင်သည့်အတိုင်းဖြစ်လျှင် code များ အားလုံးပေါင်းမည်။

အဆင့်(၁) root window တစ်ခု တည်ဆောက်မည်။

```
import tkinter as tk
from tkinter import ttk

root = tk.Tk() # Create instance
root.title("Python GUI") # Add a title
root.resizable(0, 0) # Disable resizing the GUI

root.mainloop() # Start GUI
```

အဆင့်(၂) root window ထဲတွင် my_label ဆိုသည့် label ထည့်မည်။

```
# adding a Label
my_label = ttk.Label(root, text="A Label" font = "Arial 22 bold")
my_label.grid(column=0, row=0)
```

my_label ဟု အမည်ပေးထားသည့် label widget တစ်ခုကို တည်ဆောက်သည်။ assign လုပ်သည် ရေးသည်။ my_label သည် variable တစ်ခု ဖြစ်သည်။ .grid() ဖြင့် my_label ၏ နေရာကို သတ်မှတ်ပေးသည်။ တစ်နည်းအားဖြင့် column=0, row=0 နေရာတွင် နေရာချသည်။

အောက်တွင် ဖော်ပြထားသည့် clickMe() function ဖြင့် my_label ၏ property ကို ပြောင်းသည်။ my_label သည် module-level variable ဖြစ်သောကြောင့် my_label ၏ property များကို function မှ access လုပ်နိုင်သည်။

ထို့သို့ function မှ my_label ၏ property များကို access လုပ်နိုင်ရန်အတွက် my_label ကို အရင် တည်ဆောက်ပြီးမှ clickMe() function ကို အောက်တွင် define လုပ်ရသည်။

clickMe() function ကို အရင် define လုပ်ပြီးမှ my_label ကို တည်ဆောက်လျှင် my_label ၏ property များကို access မလုပ်နိုင်ပါ။

အဆင့်(၃) root window ထဲတွင် action_button ဆိုသည့် label ထည့်မည်။

```
# Adding a Button
action_button = ttk.Button(root, text="Click Me!", command=clickMe)
action_button.grid(column=1, row=0)
```

action_button ဟု နာမည်ပေးထားသည့် button ကို တည်ဆောက်ပြီး clickMe() function ဖြင့် bind လုပ်သည်။ command=clickMe သည် button နှင့် clickMe() function ကို bind လုပ်သည့် အရာ ဖြစ်သည်။

clickMe() ဆိုသည့် function ကို define မလုပ်ရသေးသောကြောင့် command=clickMe သည် အလုပ်လုပ် လိမ့်မည် မဟုတ်ပေ။

command=clickMe ကို ရေးသည့် အခါ လက်သညးကွင်း(parentheses) မပါဘဲ ရေးသည်ကို သတိပြုပါ။

အဆင့်(၄) action_button ကို click လိုက်လျှင် အရောင် ပြောင်းသွားစေရန်အတွက် clickMe() ဆိုသည့် function တစ်ခုကို define လုပ်မည်။

clickMe() ၏ အလုပ်နှစ်ခုမှာ

(က) action_button ကို click လိုက်သည့်အခါ button အပေါ်တွင် ရှိနေသည့် Click Me ဆိုသည့် စာသားသည် "I have been Clicked!" ဆိုသည့် text အဖြစ်သို့ ပြောင်းသွားမည်။ action_button ကို .configure() ဖြင့် text ကို "I have been Clicked!" အဖြစ်သို့ ပြောင်းသည်။

(ခ) my_label မှ text အရောင်သည် အနီရောင်သို့ ပြောင်းသွားမည်။

my_label ကို .configure() ဖြင့် text အရောင်ကို အနီရောင် (foreground='red') ဖြစ်အောင် ပြောင်းသည်။

```
def clickMe():
    action_button.configure(text="I have been Clicked!")
    my_label.configure(foreground='red')
```

clickMe() function သည် event handler ဖြစ်သည်။ button ကို click လုပ်လိုက်သည့်အခါ clickMe() function ကို invoke လုပ်သည်။ တစ်နည်းအားဖြင့် button ကို click လုပ်လိုက်သည့်အခါ clickMe() function ကို လှမ်းခေါ်ပြီး အလုပ်လုပ်ခိုင်းသည်။

အဆင့်(၅) code များ အားလုံးပေါင်းမည်။

မိမိ လိုချင်သည့်အတိုင်းဖြစ်လျှင် code များ အားလုံးပေါင်းမည်။

GUI သည် event-driven ဖြစ်သည်။ button ကို click လုပ်လိုက်ခြင်းဖြင့် event တစ်ခုကို ဖန်တီးလိုက်သည်။ ထို event ဖြစ်သည့်နှင့် တစ်ပြိုင်နက် button ၏ command property ဖြင့် function ကို call လုပ်သည်။

```
In [1]: # imports
import tkinter as tk
from tkinter import ttk

root = tk.Tk() # Create instance
root.title("Python GUI") # Add a title
root.resizable(0, 0) # Disable resizing the GUI

# adding a Label
my_label = ttk.Label(root, text="A Label", font = "Arial 22 bold")
my_label.grid(column=0, row=0)

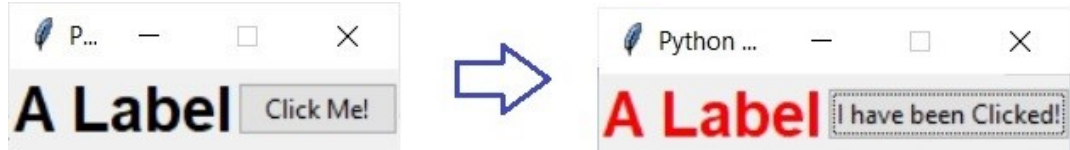
def clickMe():
    action_button.configure(text="I have been Clicked!")
    my_label.configure(foreground='red')

# Adding a Button
action_button = ttk.Button(root, text="Click Me!", command=clickMe)
action_button.grid(column=1, row=0)

root.mainloop() # Start GUI
```

```
In [2]: # အောက်မှ code သည် jupyter ipython notebook တွင် ပုံများကို ဖော်ပြရန်အတွက်  
        # ဖြစ်သည်။  
        from IPython.display import Image  
        Image(filename="click_me.jpg")
```

Out[2]:



Ref: Python GUI Programming Cookbook by Burkhard Meier

- End -

In []: