Label .set() method နှင့် window .destroy() method, Message Box

ပုံတွင် ပြထားသည့်အတိုင်း label တစ်ခု နှင့် button နှစ်ခု ပါဝင်သည်။ Button ကို click လျှင် label မှ text ကို .set() ဖြင့် ပြောင်းပေးသည်။

.set()

.set() ဖြင့် text variable ကို ပြောင်းပေးသည့်နည်း ဖြစ်သည်။ button ကို click လျှင် button က command သည် function ကို call လုပ်သည်။ function သည် .set() method ဖြင့် မိမိ အလိုရှိသည့် text variable ကို ပြောင်းပေးသည့်နည်း ဖြစ်သည်။

"Say Goodbye" button ကို click ပြီး (၂) စက္ကန့်အကြာတွင် window သည် သူ့လိုလို ပိတ်သွားလိမ့်မည်။
root.after(2000, root.destroy)

root.quit() နှင့် root.destroy() တို့၏ ခြားနားချက်

root.quit() သည် mainloop မှ exit လုပ်ရုံသာဖြစ်ပြီး widget များကျန်ခဲ့သည်။(root.quit() causes mainloop to exit.) ဆက်လက် အသုံးပြုနေနိုင်သည်။ ဥပမာ- entry widget မှ value များကို ယူနိုင်သေးသည်။

The interpreter is still intact, as are all the widgets. If you call this function, you can have code that executes after the call to root.mainloop(), and that code can interact with the widgets (for example, get a value from an entry widget).

root.destroy() သည် widget များအားလုံးကို ဖျက်ဆီးပစ်ပြီး mainloop မှ exit လုပ်သွားခြင်း ဖြစ်သည်။ ထို့ကြောင့် မည်သည့် widget ကိုမျှ access လုပ်ရန် အဖြစ်နိုင်တော့ပါ။ (Calling root.destroy() will destroy all the widgets and exit mainloop. any attempt to access any widgets will fail because the widget no longer exists.)

root.quit() နှင့် root.destroy() တို့၏ ခြားနားချက်

root.quit() သည် mainloop မှ exit လုပ်ရုံသာဖြစ်ပြီး widget များကျန်ခဲ့သည်။(root.quit() causes mainloop to exit.) ဆက်လက် အသုံးပြုနေနိုင်သည်။ ဥပမာ- entry widget မှ value များကို ယူနိုင်သေးသည်။

The interpreter is still intact, as are all the widgets. If you call this function, you can have code that executes after the call to root.mainloop(), and that code can interact with the widgets (for example, get a value from an entry widget).

root.destroy() သည် widget များအားလုံးကို ဖျက်ဆီးပစ်ပြီး mainloop မှ exit လုပ်သွားခြင်း ဖြစ်သည်။ ထို့ကြောင့် မည်သည့် widget ကိုမျှ access လုပ်ရန် အဖြစ်နိုင်တော့ပါ။ (Calling root.destroy() will destroy all the widgets and exit mainloop. any attempt to access any widgets will fail because the widget no longer exists.)
root.quit() သည် mainloop မှ exit လုပ်ရုံသာဖြစ်ပြီး widget များကျန်ခဲ့သည်။(root.quit() causes mainloop to exit.) ဆက်လက် အသုံးပြုနေနိုင်သည်။ ဥပမာ- entry widget မှ value များကို ယူနိုင်သေးသည်။

The interpreter is still intact, as are all the widgets. If you call this function, you can have code that executes after the call to root.mainloop(), and that code can interact with the widgets (for example, get a value from an entry widget).

root.destroy() သည် widget များအားလုံးကို ဖျက်ဆီးပစ်ပြီး mainloop မှ exit လုပ်သွားခြင်း ဖြစ်သည်။ ထို့ကြောင့် မည်သည့် widget ကိုမျှ access လုပ်ရန် အဖြစ်နိုင်တော့ပါ။ (Calling root.destroy() will destroy all the widgets and exit mainloop. any attempt to access any widgets will fail because the widget no longer exists.)

In [1]:
```
# အောက်ပါ code သည် jupyter ipython notebook တွင် ပုံများကို ဖော်ပြရန်အတွက် ဖြစ်သည်။
from IPython.display import Image
Image(filename="Non_oop1.jpg")
```

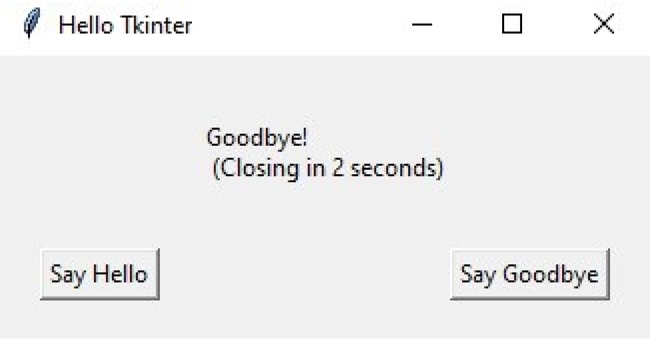
Out[1]:

In [2]:
```
Image(filename="Non_oop2.jpg")
```

Out[2]:

In [3]:
```
Image(filename="Non_oop3.jpg")
```

Out[3]:



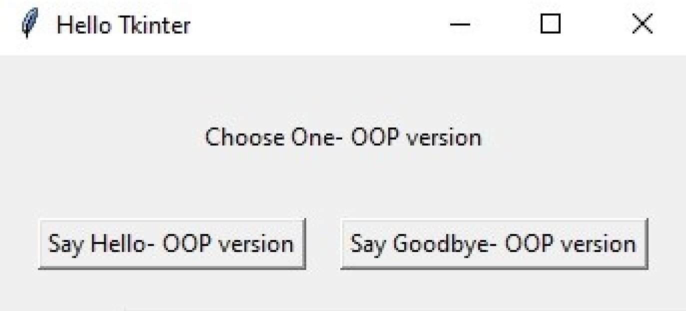
In [4]:
```
import tkinter as tk

root = tk.Tk()
root.maxsize(400, 400)
root.title("Hello Tkinter")

# .set() method ကို သုံးရန် tk.StringVar() ကို define လုပ်သည်။
label_text = tk.StringVar()
label_text.set("Choose One")

# "Say Hello" button အတွက် funciton ကို define
def say_hello():
    label_text.set("Hello World")

# "Say Goodbye" button အတွက် funciton ကို define
def say_goodbye():
    label_text.set("Goodbye! \n (Closing in 2 seconds)")
    root.after(2000, root.destroy) # window ကို ဖျက်ပစ်ရန်

# Label ကို creat လုပ်သည်။
my_txt = tk.Label(root, textvariable=label_text, anchor=tk.NW, justify=tk.LEFT, wraplength=398)
my_txt.pack(fill=tk.BOTH, expand=1, padx=100, pady=30)

# "Say Hello" button ကို creat လုပ်သည်။
hello_button = tk.Button(root, text="Say Hello", command=say_hello)
hello_button.pack(side=tk.LEFT, padx=(20, 0), pady=(0, 20))

# "Say Goodbye" button ကို creat လုပ်သည်။
goodbye_button = tk.Button(root, text="Say Goodbye", command=say_goodbye)
goodbye_button.pack(side=tk.RIGHT, padx=(0, 20), pady=(0, 20))

root.mainloop()
```

OOP Version

အပေါ်တွင် ရေးခဲ့သည့် code ကို OOP version ဖြင့် ပြန်လည် ရေးထားသည်။ OOP သဘောတရားနှင့် မရင်းနှီးသူများ ကျော်ဖတ် ပေးပါ။

```
In [5]: Image(filename="oop1.jpg")
```

Out[5]:

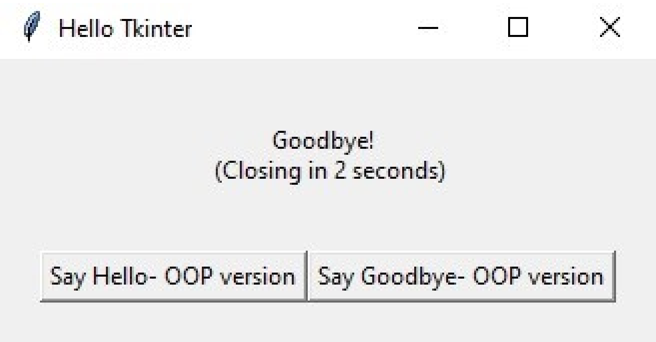



```
In [6]: Image(filename="oop2.jpg")
```

Out[6]:

```
In [7]: Image(filename="oop3.jpg")
```

Out[7]:



In [8]:
```
import tkinter as tk

class Window(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Hello Tkinter")
        # .set() method ကို သုံးရန် tk.StringVar() ကို define လုပ်သည်။
        self.label_text = tk.StringVar()
        self.label_text.set("Choose One- OOP version")

        # Label ကို creat လုပ်သည်။
        self.label = tk.Label(self, textvar=self.label_text)
        self.label.pack(fill=tk.BOTH, expand=1, padx=100, pady=3
0)

        # "Say Hello" button ကို creat လုပ်သည်။
        hello_button = tk.Button(self, text="Say Hello- OOP versi
on", command=self.say_hello)
        hello_button.pack(side=tk.LEFT, padx=(20, 0), pady=(0, 2
0))

        # "Say Goodbye" button ကို creat လုပ်သည်။
        goodbye_button = tk.Button(self, text="Say Goodbye- OOP v
ersion", command=self.say_goodbye)
        goodbye_button.pack(side=tk.RIGHT, padx=(0, 20), pady=(0,
20))

    # "Say Hello" button အတွက် funciton ကို define
    def say_hello(self):
        self.label_text.set("Hello World")

    # "Say Goodbye" button အတွက် funciton ကို define
    def say_goodbye(self):
        self.label_text.set("Goodbye! \n (Closing in 2 seconds)")
        self.after(2000, self.destroy)


if __name__ == "__main__":
    window = Window()
    window.mainloop()
```

Message Box

message box ပေါ်စေရန် tkinter ထဲမှ messagebox ကို import လုပ်ရသည်။

```
import tkinter.messagebox as msgbox
```

message box ပေါ်တွင် "Hello", "Hello World!" ကို ဖော်ပြရန်အတွက်

```
msgbox.showinfo("Hello", "Hello World!")
```

say_goodbye() function သည် အလုပ် (၃) ခု လုပ်ပေးသည်။
(၁) "Window will close in 2 seconds" စာသားကို ဖော်ပြပေးရန်

```
label_text.set("Window will close in 2 seconds")
```

.set() method ကို သုံးသည်။

(၂) "Goodbye!", "Goodbye, it's been fun!" စာသားပါသည့် message box ပေါ်ပေးရန်

```
msgbox.showinfo("Goodbye!", "Goodbye, it's been fun!")
```

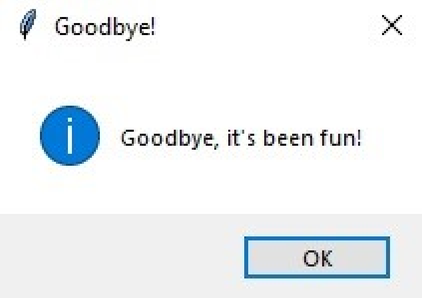
(၃) "Goodbye!" message box မှ Ok ကို click ပြီး (၂) စက္ကန့်အကြာတွင် window သည် သူ့လိုလို ပိတ်သွားစေရန်တို့ ဖြစ်သည်။
message box တွင် OK button တစ်ခုတည်ပါသည်ကို သတိပြုပါ။

In [9]:
```
Image(filename="hello_message_box.jpg")
```

Out[9]:

In [10]:
```
Image(filename="goodby_message_box.jpg")
```

Out[10]:



In [11]:
```
import tkinter as tk
import tkinter.messagebox as msgbox

root= tk.Tk()
root.title("Hello Tkinter")

def say_hello():
    msgbox.showinfo("Hello", "Hello World!")

def say_goodbye():
    label_text.set("Window will close in 2 seconds")
    msgbox.showinfo("Goodbye!", "Goodbye, it's been fun!")
    root.after(2000, root.destroy)

label_text = tk.StringVar()
label_text.set("Choose One")

label = tk.Label(root, textvar=label_text)
label.pack(fill=tk.BOTH, expand=1, padx=100, pady=30)

hello_button = tk.Button(root, text="Say Hello", command=say_hell
o)
hello_button.pack(side=tk.LEFT, padx=(20, 0), pady=(0, 20))

goodbye_button = tk.Button(root, text="Say Goodbye", command=say_
goodbye)
goodbye_button.pack(side=tk.RIGHT, padx=(0, 20), pady=(0, 20))

root.mainloop()
```

OOP Version of above code

In [12]:
```
import tkinter as tk
import tkinter.messagebox as msgbox

class Window(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Hello Tkinter")
        self.label_text = tk.StringVar()
        self.label_text.set("Choose One")

        self.label = tk.Label(self, textvar=self.label_text)
        self.label.pack(fill=tk.BOTH, expand=1, padx=100, pady=3
0)

        hello_button = tk.Button(self, text="Say Hello", command=
self.say_hello)
        hello_button.pack(side=tk.LEFT, padx=(20, 0), pady=(0, 2
0))

        goodbye_button = tk.Button(self, text="Say Goodbye", comm
and=self.say_goodbye)
        goodbye_button.pack(side=tk.RIGHT, padx=(0, 20), pady=(0,
20))

    def say_hello(self):
        msgbox.showinfo("Hello", "Hello World!")

    def say_goodbye(self):
        self.label_text.set("Window will close in 2 seconds")
        msgbox.showinfo("Goodbye!", "Goodbye, it's been fun!")
        self.after(2000, self.destroy)

if __name__ == "__main__":
    window = Window()
    window.mainloop()
```

Message Box with Yes No button and Name entry

user က ထည့်ပေးမည့် နာမည်ကို လက်ခံရန်အတွက် name_text ဆိုသည့် variabel တစ်ခုကို define လုပ်သည်။

```
name_text = tk.StringVar()
```

user နာမည်ကို ပြန်ခေါ်လိုသည်အခါ .get()ကို အသုံးပြုသည်။ ဥပမာ- name_entry.get()

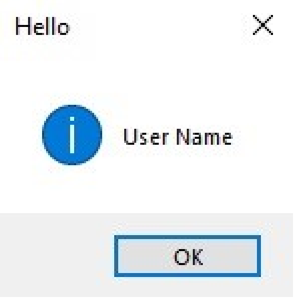
messagebox ပေါ်တွင် ပေါ်မည့် text အတွက် "Hello there " နှင့် name_entry.get() ကို concatnate လုပ်သည်။

အောက် code သည် user နာမည်ပါသည့် messagebox တစ်ခု တည်ဆောက်ပေးသည်။

```
def say_hello():
    message = "Hello there " + name_entry.get()
    msgbox.showinfo("Hello", message)
```

```
In [13]: Image(filename="Hello_User_MessageBox.jpg")
```

Out[13]:



say_goodbye() function

msgbox.askyesno() သည် messagebox ပေါ်တွင် "yes" နှင့် "no" button နှစ်ခုကို ဖော်ပြပေးရန် ဖြစ်သည်။ if statement ကို သုံးထားသည်။ True ဖြစ်လျှင် လုပ်ရန် နှင့် False ဖြစ်လျှင် လုပ်ရန်တို့ကို if .. else အောက်တွင် ရေထားသည်။

```
def say_goodbye():
    if msgbox.askyesno("Close Window?", "Would you like to close this 
window?"):
        message = "Window will close in 2 seconds - goodybye " + name
_entry.get()
        label_text.set(message)
        after(2000, self.destroy)
    else:
        msgbox.showinfo("Not Closing", "Great! This window will stay 
open.")
```

Yes ကို click လျှင် ပေါ်လာမည့် message box အတွက်

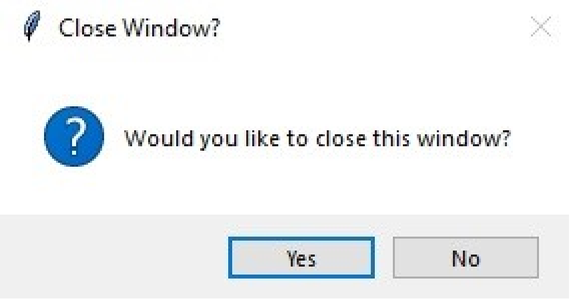
```
message = "Window will close in 2 seconds - goodybye " + name_entry.g
et()
        label_text.set(message)
        after(2000, self.destroy)
```

No ကို click လျှင် ပေါ်လာမည့် message box အတွက်

```
msgbox.showinfo("Not Closing", "Great! This window will stay open.")
```

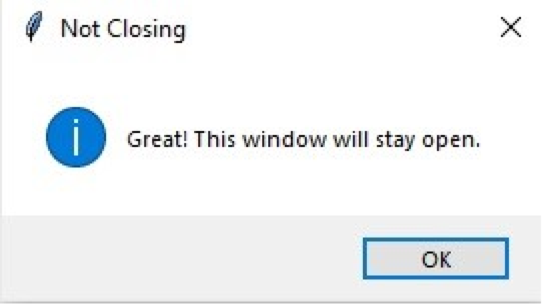
```
In [14]: Image(filename="MessageBox_Yes_No.jpg")
```

Out[14]:



In [15]:
```
Image(filename="Not_closing.jpg")
```

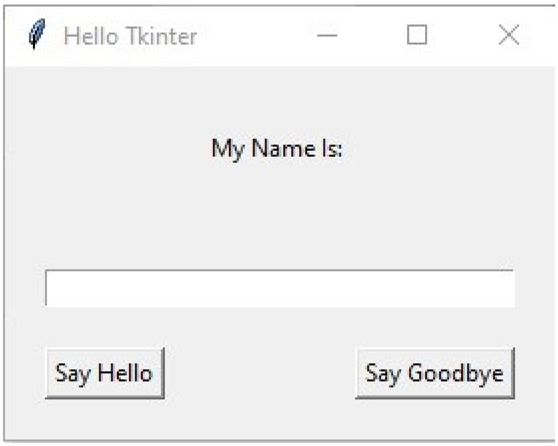
Out[15]:



In [16]:
```
import tkinter as tk
import tkinter.messagebox as msgbox

root= tk.Tk()
root.title("Hello Tkinter")

def say_hello():
    message = "Hello there " + name_entry.get()
    msgbox.showinfo("Hello", message)

def say_goodbye():
    if msgbox.askyesno("Close Window?", "Would you like to close 
this window?"):
        message = "Window will close in 2 seconds - goodybye " + 
name_entry.get()
        label_text.set(message)
        root.after(2000, root.destroy)
    else:
        msgbox.showinfo("Not Closing", "Great! This window will s
tay open.")

name_text = tk.StringVar()

label_text = tk.StringVar()
label_text.set("My Name Is: ")

label = tk.Label(root, textvar=label_text)
label.pack(fill=tk.BOTH, expand=1, padx=100, pady=30)

name_entry = tk.Entry(root, textvar= name_text)
name_entry.pack(fill=tk.BOTH, expand=1, padx=20, pady=20)

hello_button = tk.Button(root, text="Say Hello", command=say_hell
o)
hello_button.pack(side=tk.LEFT, padx=(20, 0), pady=(0, 20))

goodbye_button = tk.Button(root, text="Say Goodbye", command=say_
goodbye)
goodbye_button.pack(side=tk.RIGHT, padx=(0, 20), pady=(0, 20))

root.mainloop()
```

In [17]:
```
Image(filename="Name_Entry.jpg")
```

Out[17]:



OOP Version

In [18]:
```
import tkinter as tk
import tkinter.messagebox as msgbox

class Window(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Hello Tkinter")
        self.label_text = tk.StringVar()
        self.label_text.set("My Name Is: ")

        self.name_text = tk.StringVar()

        self.label = tk.Label(self, textvar=self.label_text)
        self.label.pack(fill=tk.BOTH, expand=1, padx=100, pady=1
0)

        self.name_entry = tk.Entry(self, textvar=self.name_text)
        self.name_entry.pack(fill=tk.BOTH, expand=1, padx=20, pad
y=20)

        hello_button = tk.Button(self, text="Say Hello", command=
self.say_hello)
        hello_button.pack(side=tk.LEFT, padx=(20, 0), pady=(0, 2
0))

        goodbye_button = tk.Button(self, text="Say Goodbye", comm
and=self.say_goodbye)
        goodbye_button.pack(side=tk.RIGHT, padx=(0, 20), pady=(0,
20))

    def say_hello(self):
        message = "Hello there " + self.name_entry.get()
        msgbox.showinfo("Hello", message)

    def say_goodbye(self):
        if msgbox.askyesno("Close Window?", "Would you like to cl
ose this window?"):
            message = "Window will close in 2 seconds - goodybye
" + self.name_entry.get()
            self.label_text.set(message)
            self.after(2000, self.destroy)
        else:
            msgbox.showinfo("Not Closing", "Great! This window wi
ll stay open.")


if __name__ == "__main__":
    window = Window()
    window.mainloop()
```

Ref: Tkinter GUI Programming by Example

End - ကောင်းထက်ညွန့်

In [ ]:
```

```