# 01076105, 01075106

# Object Oriented Programming

# Object Oriented Programming Project

TTKBootstrap

# TTKBootstrap

- เป็น Library GUI ของ Python ที่พัฒนามาจาก TKinter
- https://ttkbootstrap.readthedocs.io/

## 📦 Features

**✓ Built-in Themes**
Over a dozen curated dark and light themes

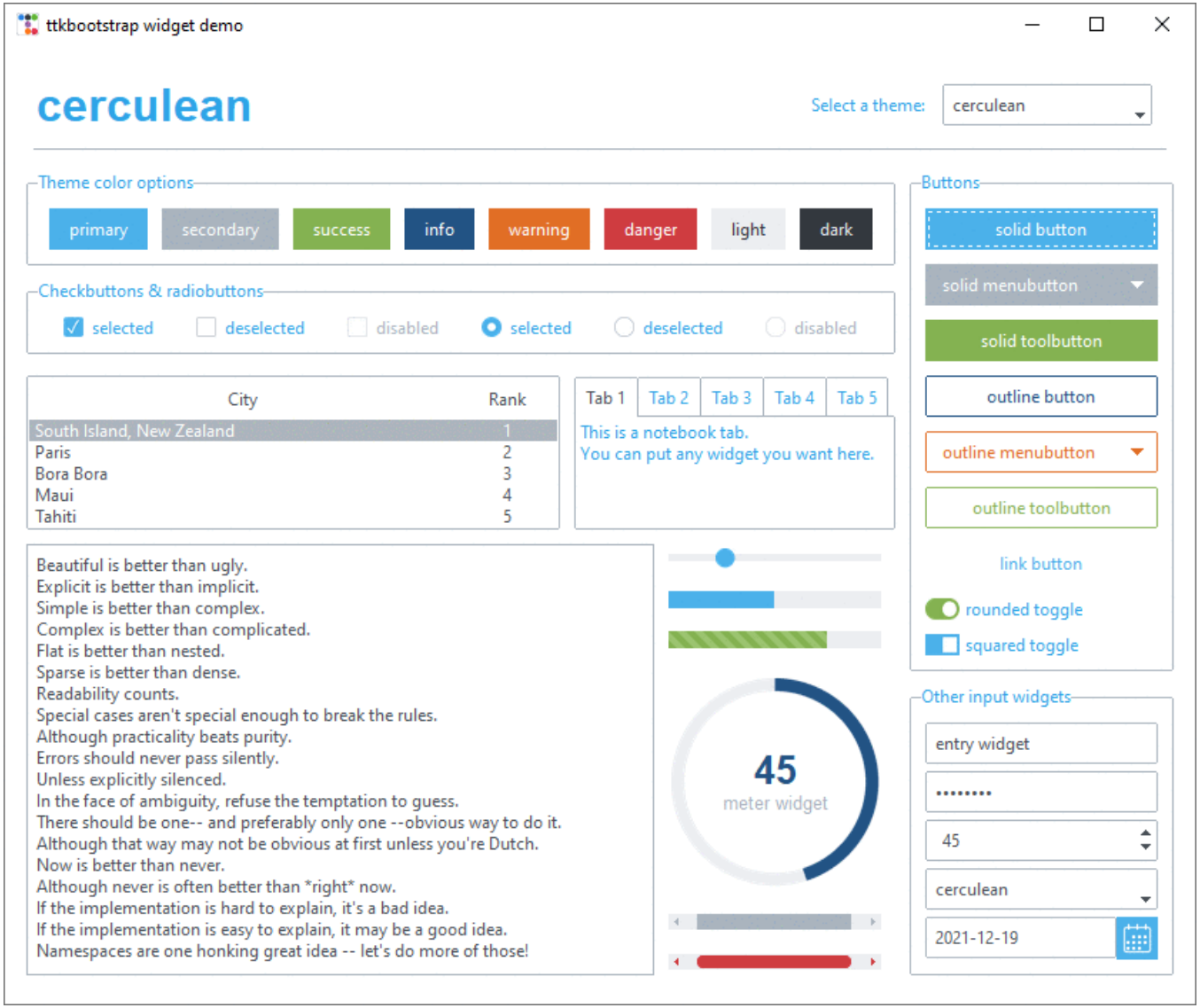
**✓ Pre-defined Styles:**
Loads of beautiful pre-defined widget styles such as **outline** and **round toggle** buttons.

**✓ Simple keyword API:**
Apply colors and types using simple keywords such as **primary** and **striped** instead of the legacy approach of **primary.Striped.Horizontal.TProgressbar**. If you've used Bootstrap for web development, you are already familiar with this approach using css classes.

**✓ Lots of new Widgets:**
ttkbootstrap comes with several new beautifully designed widgets such as Meter, DateEntry, and Floodgauge. Additionally, dialogs are now themed and fully customizable.

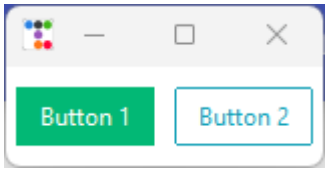
**✓ Built-in Theme Creator:**
Want to create your own theme? Easy! ttkbootstrap includes a built-in theme creator that enables you to easily build, load, expore, and apply your own custom themes.

# TTKBootstrap



# TTKBootstrap

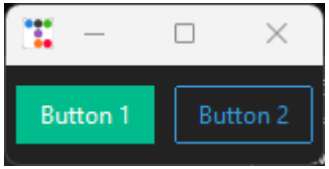
- การติดตั้ง
  - python -m pip install ttkbootstrap




```
import ttkbootstrap as ttk
from ttkbootstrap.constants import *

root = ttk.Window()

b1 = ttk.Button(root, text="Button 1", bootstyle=SUCCESS)
b1.pack(side=LEFT, padx=5, pady=10)

b2 = ttk.Button(root, text="Button 2", bootstyle=(INFO, OUTLINE))
b2.pack(side=LEFT, padx=5, pady=10)

root.mainloop()
```

# TTKBootstrap

- การเลือก Theme
- ให้ป้อนชื่อ theme ลงใน windows ตามตัวอย่าง

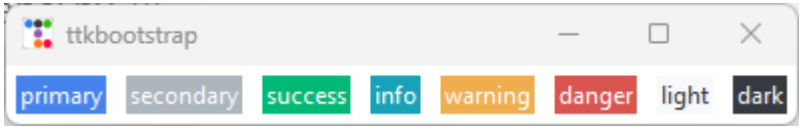



```
import ttkbootstrap as ttk
from ttkbootstrap.constants import *

root = ttk.Window(themename="darkly")

b1 = ttk.Button(root, text="Button 1", bootstyle=SUCCESS)
b1.pack(side=LEFT, padx=5, pady=10)

b2 = ttk.Button(root, text="Button 2", bootstyle=(INFO, OUTLINE))
b2.pack(side=LEFT, padx=5, pady=10)

root.mainloop()
```

# TTKBootstrap

- Label คือ ส่วนที่แสดงข้อความ มี 2 แบบ คือ default label มีสีตามนี้ (สีจะเปลี่ยนไปตาม theme)


- และ Inverse Label




| Keyword | Description | Example |
|---|---|---|
| primary | The default color for most widgets | primary |
| secondary | Typically a *gray* color | secondary |
| success | Typically a *green* color | success |
| info | Typically a *blue* color | info |

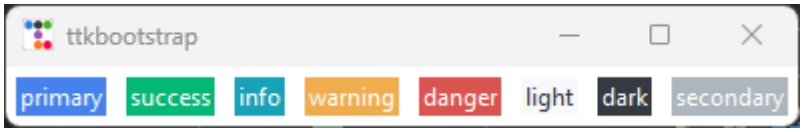
| Keyword | Description | Example |
|---|---|---|
| warning | Typically an *orange* color | warning |
| danger | Typically a *red* color | danger |
| light | Typically a *light gray* color | light |
| dark | Typically a *dark gray* color | dark |

# TTKBootstrap

```
import ttkbootstrap as ttk
from ttkbootstrap.constants import *

root = ttk.Window()

# default label style
l1 = ttk.Label(root, text = "primary", bootstyle="PRIMARY.Inverse")
l1.pack(side=LEFT, padx=5, pady=5)

l2 = ttk.Label(root, text='secondary', bootstyle="SECONDARY.Inverse")
l2.pack(side=LEFT, padx=5, pady=5)

l3 = ttk.Label(root, text='success', bootstyle="SUCCESS.Inverse")
l3.pack(side=LEFT, padx=5, pady=5)

l4 = ttk.Label(root, text='info', bootstyle="INFO.Inverse")
l4.pack(side=LEFT, padx=5, pady=5)

root.mainloop()
```

# TTKBootstrap

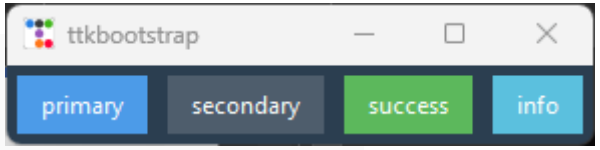
- `l1 = ttk.Label(root, text = "primary", bootstyle="PRIMARY.Inverse")`
- คือการสร้าง Object Label
- `l1.pack(side=LEFT, padx=5, pady=5)`
- Pack มีความหมายว่าให้นำ Label ไปใส่ใน Window
- side = มีความหมายว่าให้ตรึงด้านซ้าย เช่น ถ้าแก้ secondary ให้ตรึงด้านขวาจะได้ (ถ้าไม่ใส่จะอยู่ตรงกลาง




- padx คือ เติมช่องว่างในแกน x
- pady คือ เติมช่องว่างในแกน y

# TTKBootstrap

- Button คือ ปุ่ม แต่สามารถใช้แสดงข้อความเช่นเดียวกับ Label ได้

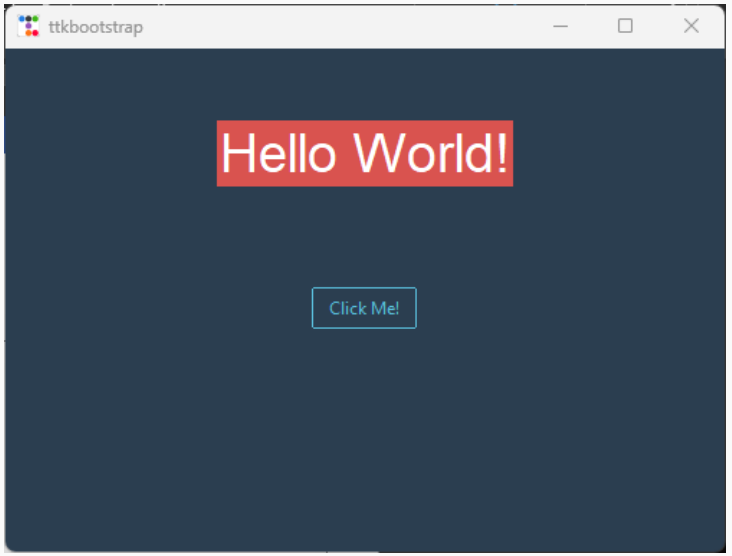



```
import ttkbootstrap as ttk
from ttkbootstrap.constants import *

root = ttk.Window(themename="superhero")

b1 = ttk.Button(root, text='primary', bootstyle=PRIMARY)
b1.pack(side=LEFT, padx=5, pady=5)

b2 = ttk.Button(root, text='secondary', bootstyle=SECONDARY)
b2.pack(side=LEFT, padx=5, pady=5)

b3 = ttk.Button(root, text='success', bootstyle=SUCCESS)
b3.pack(side=LEFT, padx=5, pady=5)

b4 = ttk.Button(root, text='info', bootstyle=INFO)
b4.pack(side=LEFT, padx=5, pady=5)

root.mainloop()
```

# TTKBootstrap

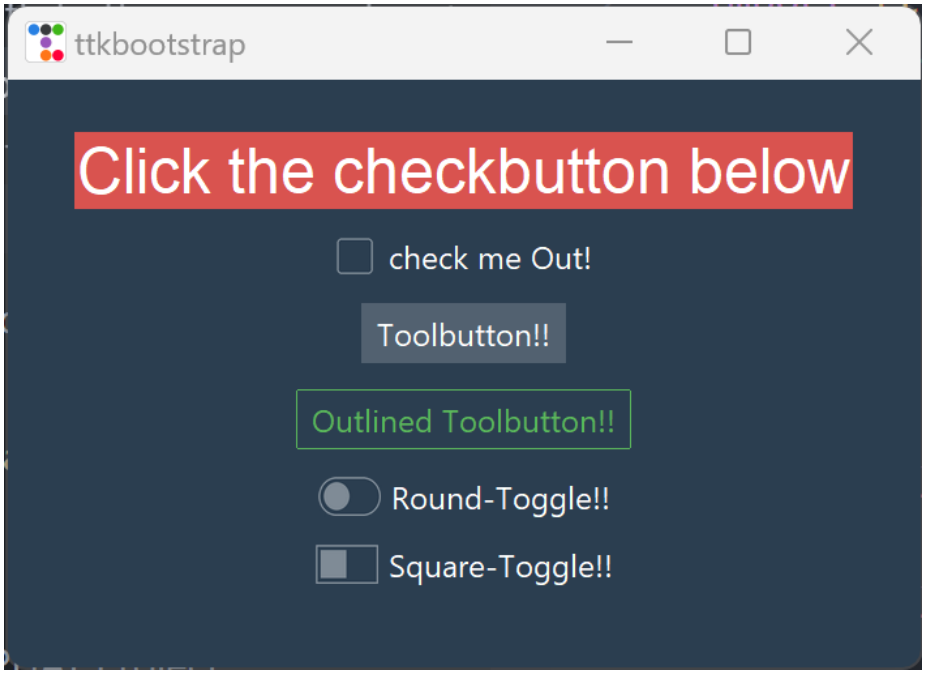
- Button สามารถใส่ command ได้ โดยจะเป็น function ที่ถูกเรียกมาทำงานเมื่อกด

```
import ttkbootstrap as ttk
from ttkbootstrap.constants import *

root = ttk.Window(themename='superhero')
root.geometry('500x350')

counter = 0
def changer():
    global counter
    counter += 1
    if counter % 2 == 0:
        my_label.config(text="Hello World!")
    else:
        my_label.config(text="Goodbye World!")

my_label = ttk.Label(root, text="Hello World", font=("Helvetica", 28), bootstyle="danger,inverse")
my_label.pack(pady=50)

my_button = ttk.Button(text="Click Me!", bootstyle="info, outline", command=changer)
my_button.pack(pady=20)

root.mainloop()
```



# TTKBootstrap

- Button ใน TTKBootstrap มีหลายแบบ ได้แก่ Button ปกติ Tool Button, Tool Button Outlined, Round Toggle, Square Button



# TTKBootstrap

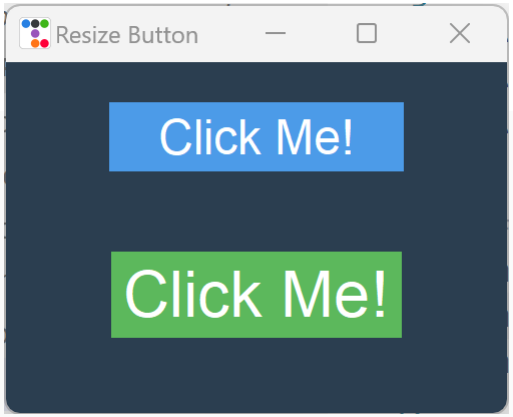
```
#Toolbutton
var2 = IntVar()
my_check2 = ttk.Checkbutton(bootstyle="danger, toolbutton",
                            text="Toolbutton!!",
                            variable=var2,
                            onvalue=1,
                            offvalue=0,
                            command=checker)
my_check2.pack(pady=10)

#Toolbutton outlined
var3 = IntVar()
my_check3 = ttk.Checkbutton(bootstyle="success, toolbutton, outline",
                            text="Outlined Toolbutton!!",
                            variable=var3,
                            onvalue=1,
                            offvalue=0,
                            command=checker)
my_check3.pack(pady=10)
```

# TTKBootstrap

```
#Round Toggle
var4 = IntVar()
my_check4 = ttk.Checkbutton(bootstyle="info, round-toggle",
                            text="Round-Toggle!!",
                            variable=var4,
                            onvalue=1,
                            offvalue=0,
                            command=checker)
my_check4.pack(pady=10)

#Square Toggle
var5 = IntVar()
my_check5 = ttk.Checkbutton(bootstyle="info, square-toggle",
                            text="Square-Toggle!!",
                            variable=var5,
                            onvalue=1,
                            offvalue=0,
                            command=checker)
my_check5.pack(pady=10)
```

# TTKBootstrap

- เราสามารถเปลี่ยน style เองได้

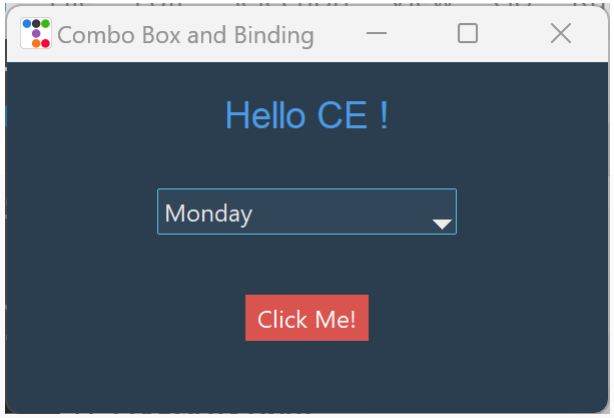



```
from tkinter import *
import ttkbootstrap as ttk

root = ttk.Window(themename='superhero')
root.title("Resize Button")
root.geometry('500x350')

#Style
my_style = ttk.Style()
my_style.configure('my.TButton', font=("Helvetica",18), width=10)
my_style.configure('success.Outline.TButton', font=("Helvetica",24))

my_button = ttk.Button(text="Click Me!", bootstyle="info", style="my.TButton")
my_button.pack(pady=40)

my2_button = ttk.Button(text="Click Me!", bootstyle="info", style="success.Outline.TButton")
my2_button.pack(pady=40)

root.mainloop()
```

https://ttkbootstrap.readthedocs.io/en/latest/styleguide/button/

# TTKBootstrap

- Combobox สำหรับเลือกข้อมูลที่มีหลายค่า

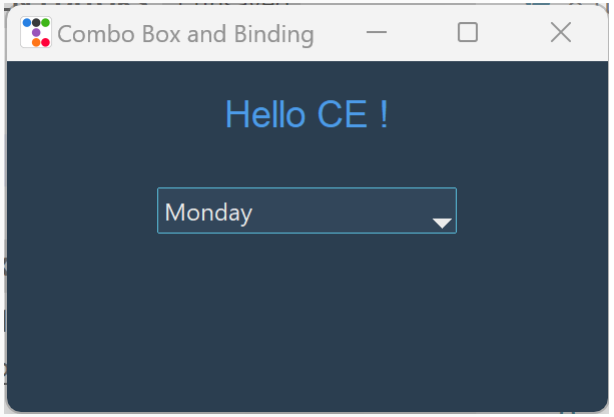



```
from tkinter import *
import ttkbootstrap as ttk

root = ttk.Window(themename='superhero')
root.title("Combo Box and Binding")
root.geometry('600x350')

my_label = ttk.Label(root, text = "Hello CE !", font=("Helvetica",14), bootstyle="PRIMARY")
my_label.pack(pady=30)

def clicker():
        my_label.config(text=f"You Clinked on {my_combo.get()}!")

days = ['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']

my_combo = ttk.Combobox(root, bootstyle="info", values=days)
my_combo.pack(pady=20)

# set default value
my_combo.current(0)

my_button = ttk.Button(text="Click Me!", bootstyle="danger", command=clicker)
my_button.pack(pady=40)

root.mainloop()
```

# TTKBootstrap

- Combobox สามารถ bind กับ function โดยไม่ต้องกดปุ่มได้

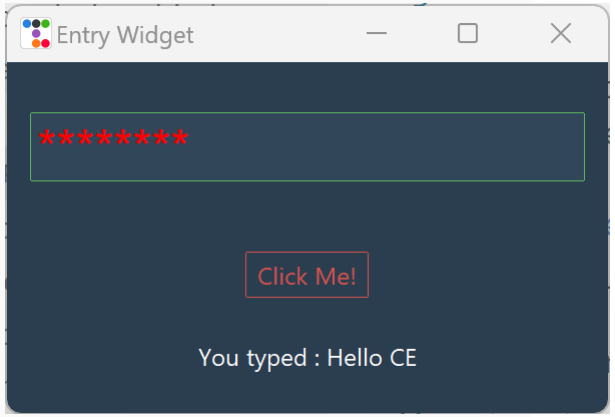
```
from tkinter import *
import ttkbootstrap as ttk

root = ttk.Window(themename='superhero')
root.title("Combo Box and Binding")
root.geometry('600x350')

my_label = ttk.Label(root, text = "Hello CE !", font=("Helvetica",14), bootstyle="PRIMARY")
my_label.pack(pady=30)

def click_bind(e):
    my_label.config(text=f"You Clinked on {my_combo.get()}!")

days = ['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']

my_combo = ttk.Combobox(root, bootstyle="info", values=days)
my_combo.pack(pady=20)

# set default value
my_combo.current(0)

#bind combo box
my_combo.bind("<<ComboboxSelected>>", click_bind)

root.mainloop()
```



# TTKBootstrap

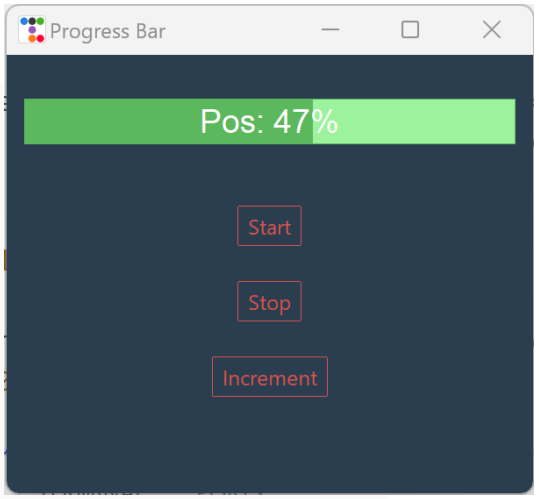
- ช่องป้อนข้อมูล รับข้อมูลได้ และ กำหนดให้แสดงเป็นอักษรอื่นได้

```
from tkinter import *
import ttkbootstrap as ttk

root = ttk.Window(themename='superhero')
root.title("Entry Widget")
root.geometry('600x350')

def speak():
    my_label.config(text=f"You typed : {my_entry.get()}")

my_entry = ttk.Entry(root, bootstyle="success",
                     font=("Helvetica",18),
                     foreground="red",
                     width=20,show="*")
my_entry.pack(pady=50)

my_button = ttk.Button(root, text="Click Me!", bootstyle="danger, outline", command=speak)
my_button.pack(pady=20)

my_label = ttk.Label(root, text = "")
my_label.pack(pady=20)

root.mainloop()
```



# TTKBootstrap

- สามารถทำ Progress Bar ได้ เรียกว่า Floodgate



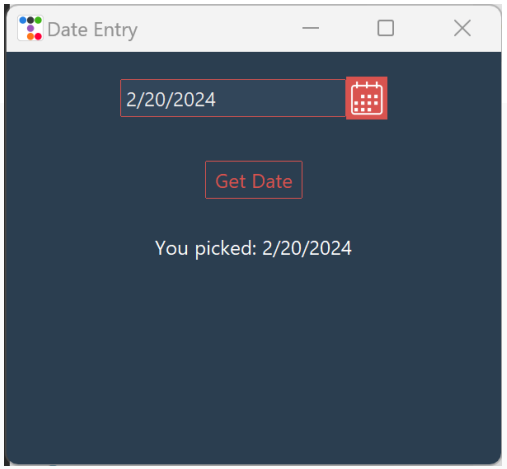

```
from tkinter import *
import ttkbootstrap as ttk

root = ttk.Window(themename='superhero')
root.title("Progress Bar")
root.geometry('600x500')

def starter():
    my_gauge.start()
def stoper():
    my_gauge.stop()
def incrementer():
    my_gauge.step(10)
```

มี Progressbar ใช้คล้ายกัน

```
my_gauge = ttk.Floodgauge(root, bootstyle="success", font=("Helvetica",14),mask="Pos: {}%",maximum=80,
                          orient="horizontal",value=0, mode="determinate")  # try indeterminate
my_gauge.pack(pady=50,fill=X,padx=20)

start_button = ttk.Button(root, text="Start", bootstyle="danger, outline", command=starter)
start_button.pack(pady=20)

stop_button = ttk.Button(root, text="Stop", bootstyle="danger, outline", command=stoper)
stop_button.pack(pady=20)

increment_button = ttk.Button(root, text="Increment", bootstyle="danger, outline", command=incrementer)
increment_button.pack(pady=20)

root.mainloop()
```

# TTKBootstrap

- Date Picker

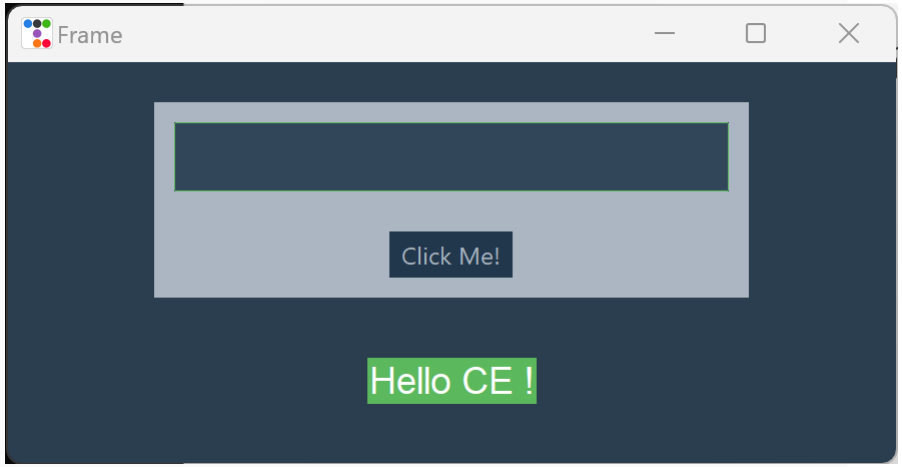



```
from tkinter import *
import ttkbootstrap as ttk
from datetime import date

root = ttk.Window(themename='superhero')
root.title("Date Entry")
root.geometry('600x500')

def datey():
    my_label.config(text=f"You picked: {my_date.entry.get()}")


my_date = ttk.DateEntry(root, bootstyle="danger", startdate=date.today())
my_date.pack(pady=30)

my_button = ttk.Button(root, text="Get Date", bootstyle="danger, outline", command=datey)
my_button.pack(pady=20)

my_label = ttk.Label(root, text = "You Picked: ")
my_label.pack(pady=20)

root.mainloop()
```

# TTKBootstrap

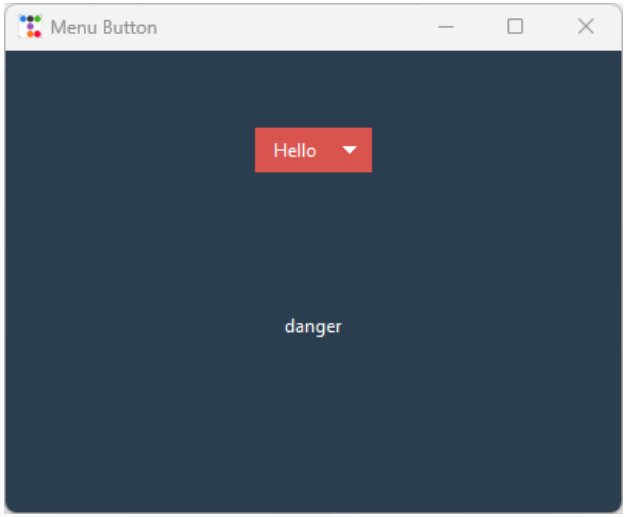
- กรณีที่มี widget มาก เราอาจต้องจัด UI ออกเป็นส่วนๆ เรียกว่า Frame

```
from tkinter import *
import ttkbootstrap as ttk
from datetime import date

root = ttk.Window(themename='superhero')
root.title("Frame")
root.geometry('500x400')

def speak():
    pass

my_frame = ttk.Frame(root, bootstyle="light")
my_frame.pack(pady=40)

my_entry = ttk.Entry(my_frame, bootstyle="success",
                     font=("Helvetica",18))
my_entry.pack(pady=20, padx=20)

my_button = ttk.Button(my_frame, text="Click Me!", bootstyle="dark", command=speak)
my_button.pack(pady=20, padx=20)

my_label = ttk.Label(root, text = "Hello CE !", font=("Helvetica",14), bootstyle="inverse success")
my_label.pack(pady=20)

root.mainloop()
```



# TTKBootstrap

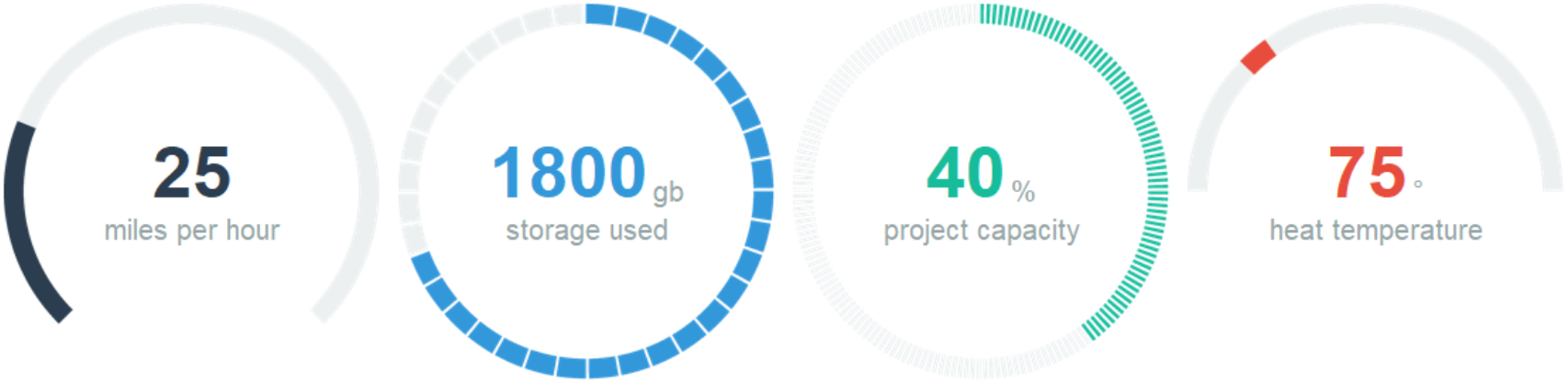
- สามารถสร้างเมนูได้




```
root = ttk.Window(themename='superhero')
root.title("Menu Button")
root.geometry('500x400')

def stuff(x):
    my_menu.config(bootstyle=x)
    my_label.config(text=x)

my_menu = ttk.Menubutton(root, bootstyle="warning", text="Hello")
my_menu.pack(pady=50)

inside_menu = ttk.Menu(my_menu)

# add item to menu
item_var = StringVar()
for x in ['primary','secondary','danger', 'info', 'outline primary',
          'outline secondary', 'outline danger', 'outline info']:
    inside_menu.add_radiobutton(label=x, variable=item_var, command=lambda x=x: stuff(x))

# associate the inside menu with menubutton
my_menu['menu'] = inside_menu

my_label = ttk.Label(root, text="")
my_label.pack(pady=40)

root.mainloop()
```

# TTKBootstrap

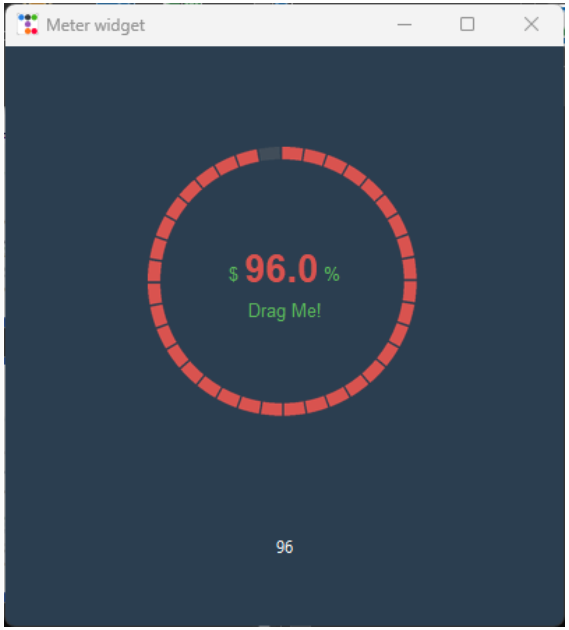
- Meter Widget
- ใช้ในการเลือกข้อมูลในแบบวงกลม สามารถปรับแต่งได้หลากหลาย




https://ttkbootstrap.readthedocs.io/en/latest/api/widgets/meter/#ttkbootstrap.widgets.Meter.__init__

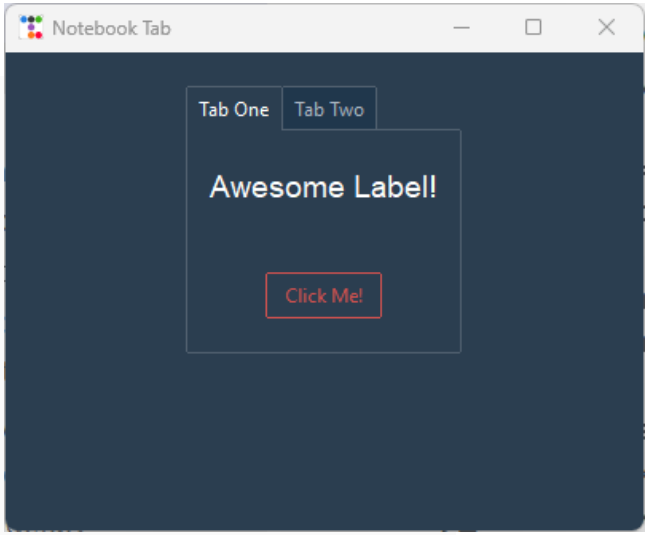
# TTKBootstrap

```python
from tkinter import *
import ttkbootstrap as ttk

root = ttk.Window(themename='superhero')
root.title("Meter widget")
root.geometry('400x500')

def update_label():
    my_label.config(text=my_meter.amountusedvar.get())

my_meter = ttk.Meter(root, bootstyle="danger",
                     subtext="Drag Me!",
                     interactive=True,
                     textleft="$", textright="%",
                     metertype="full",  # try semi
                     stripethickness=10,# แสดงแถบในวงกลม
                     metersize=200, padding=50,
                     amountused=20,     # ค่าเริ่มต้น
                     amounttotal=100,   # ค่ามากสุด
                     subtextstyle="success"
                     )
my_meter.pack(pady=20)

my_label = ttk.Label(root, text=my_meter.amountusedvar.get())
my_label.pack(pady=10)

update_label()  # Set the initial label text
my_meter.amountusedvar.trace_add('write', lambda *args: update_label())

root.mainloop()
```



# TTKBootstrap

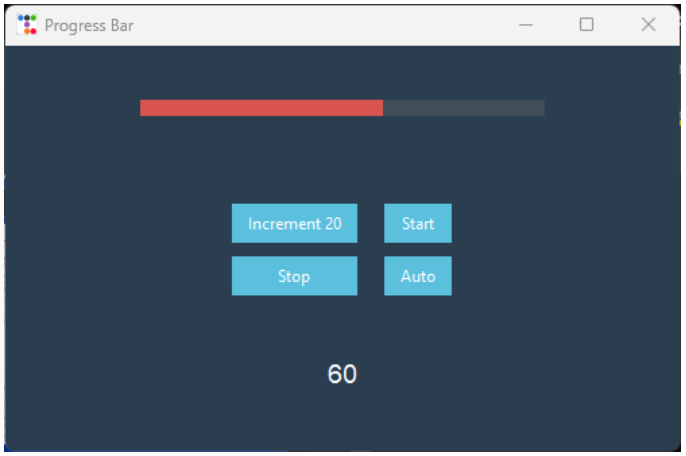
- Tab Notebook ใช้สำหรับสร้างพื้นที่ทำงานในรูปแบบ Tab




```
root = ttk.Window(themename='superhero')
root.title("Notebook Tab")
root.geometry('400x300')

my_notebook = ttk.Notebook(root, bootstyle="dark")
my_notebook.pack(pady=20)

tab1=ttk.Frame(my_notebook)
tab2=ttk.Frame(my_notebook)

my_label = Label(tab1, text="Awesome Label!", font=("Helvetica",14))
my_label.pack(pady=20)
my_text = Text(tab1, width=70, height=10)
my_label.pack(padx=10, pady=20)
my_button = ttk.Button(tab1, text="Click Me!", bootstyle="danger outline")
my_button.pack(pady=20)

my_notebook.add(tab1, text="Tab One")
my_notebook.add(tab2, text="Tab Two")

root.mainloop()
```

# TTKBootstrap

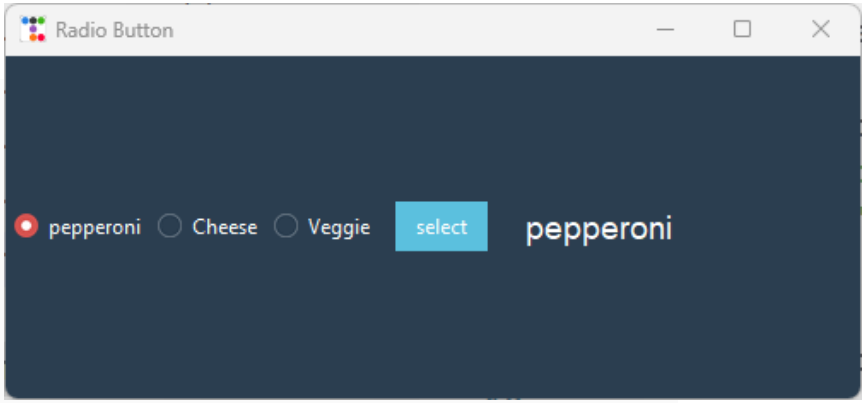
- การใช้ grid เพื่อจัดองค์ประกอบ




```
root = ttk.Window(themename='superhero')
root.title("Progress Bar")
root.geometry('500x300')

def starter():
    my_progress.start()
def stoper():
    my_progress.stop()
def incrementer():
    my_progress['value'] += 20
    my_label.config(text=my_progress["value"])

my_progress = ttk.Progressbar(root, bootstyle="danger", maximum=100, length=300,
                              value=0, mode="determinate")  # try indeterminate
my_progress.pack(pady=40)

my_frame = ttk.Frame(root)
my_frame.pack(pady=20)

increment_button = ttk.Button(my_frame, text="Increment 20", bootstyle="info", command=incrementer)
increment_button.grid(column=0,row=0,padx=10,pady=5,sticky="news")

increment_button = ttk.Button(my_frame, text="Start", bootstyle="info", command=starter)
increment_button.grid(column=1,row=0,padx=10,pady=5,sticky="news")

increment_button = ttk.Button(my_frame, text="Stop", bootstyle="info", command=stoper)
increment_button.grid(column=0,row=1,padx=10,pady=5,sticky="news")

increment_button = ttk.Button(my_frame, text="Auto", bootstyle="info", command=incrementer)
increment_button.grid(column=1,row=1,padx=10,pady=5,sticky="news")

my_label = ttk.Label(root, text="", font=("Helvetica",14))
my_label.pack(pady=20)

root.mainloop()
```

# TTKBootstrap

- Radio Button

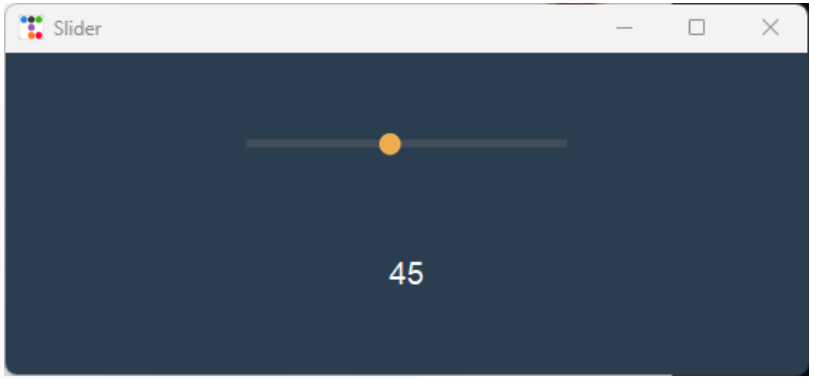



```
root = ttk.Window(themename='superhero')
root.title("Radio Button")
root.geometry('500x200')

def clicker():
    my_label.config(text=my_topping.get())

toppings = ["pepperoni","Cheese","Veggie"]
my_topping = StringVar()

for topping in toppings:
    ttk.Radiobutton(root, bootstyle="danger", variable=my_topping, text=topping,
                    value=topping).pack(side="left",padx=5,pady=5)

my_button = ttk.Button(root, text="select", bootstyle="info", command=clicker)
my_button.pack(side="left",padx=10,pady=5)

my_label = ttk.Label(root, text="", font=("Helvetica",14))
my_label.pack(side="left",padx=10)

root.mainloop()
```

ทดลองเปลี่ยนเป็น danger toolbutton

# TTKBootstrap

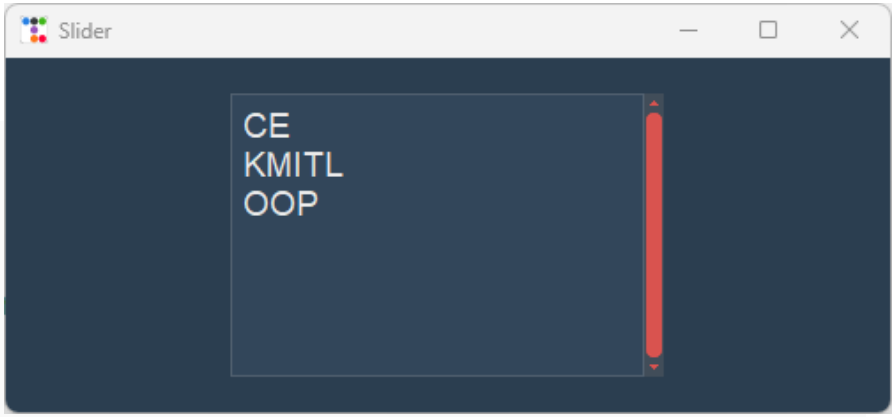
- Slider




```
from tkinter import *
import ttkbootstrap as ttk

root = ttk.Window(themename='superhero')
root.title("Slider")
root.geometry('500x200')

def scaler(e):
    my_label.config(text=int(my_scale.get()))

my_scale = ttk.Scale(root, bootstyle="warning", length=200,
                     orient="horizontal", from_=0, to=100,
                     command=scaler)
my_scale.pack(pady=50)

my_label = ttk.Label(root, text="", font=("Helvetica",14))
my_label.pack(pady=10)

root.mainloop()
```

# TTKBootstrap

- Scrollbar

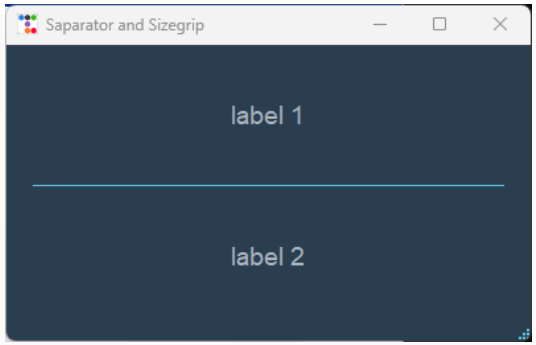



```
from tkinter import *
import ttkbootstrap as ttk

root = ttk.Window(themename='superhero')
root.title("Slider")
root.geometry('500x200')

my_frame = ttk.Frame(root)
my_frame.pack(pady=20)
my_scroll = ttk.Scrollbar(my_frame, orient="vertical", bootstyle="danger round")
my_scroll.pack(side="right", fill="y")

my_text =  Text(my_frame, width=20, height=25,
                yscrollcommand=my_scroll.set, wrap="none", font=("Helvetica",14))
my_text.pack()

my_scroll.config(command=my_text.yview)

root.mainloop()
```

# TTKBootstrap

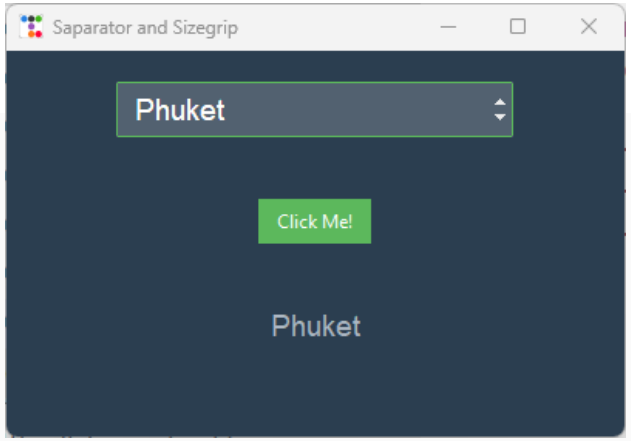
- Separator คือเส้นคั่น, Sizegrip คือ ส่วนที่ไว้ขยาย window

```
from tkinter import *
import ttkbootstrap as ttk

root = ttk.Window(themename='superhero')
root.title("Saparator and Sizegrip")
root.geometry('400x250')

label1 = ttk.Label(root, text="label 1", bootstyle="light", font=("Helvetica",14))
label1.pack(pady=40)

my_sep = ttk.Separator(root, bootstyle="info", orient="horizontal")
my_sep.pack(fill=X, padx=20)

label2 = ttk.Label(root, text="label 2", bootstyle="light", font=("Helvetica",14))
label2.pack(pady=40)

my_sizegrip = ttk.Sizegrip(root, bootstyle="info")
my_sizegrip.pack(anchor="se", fill="both", expand="true")

root.mainloop()
```



# TTKBootstrap

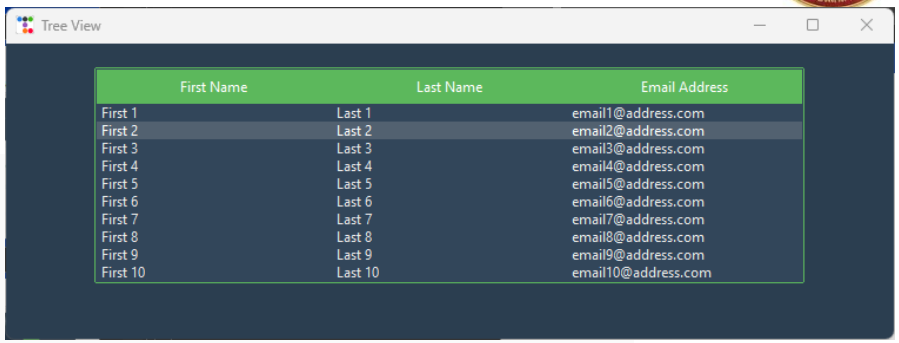
- Spinner




```
root = ttk.Window(themename='superhero')
root.title("Saparator and Sizegrip")
root.geometry('400x250')

def spiny():
    my_label.config(text=my_spin.get())

stuff = ["Bangkok", "Phuket", "Chonburi", "Chiang Mai"]

my_spin = ttk.Spinbox(root, bootstyle="success", font=("Helvetica",14),
                      from_=0, to=20, values=stuff, state="readonly")
my_spin.pack(pady=20)
my_spin.set("Bangkok")

my_button = ttk.Button(root, text="Click Me!", bootstyle="success", command=spiny)
my_button.pack(pady=20)

my_label = ttk.Label(root, text="", bootstyle="light", font=("Helvetica",14))
my_label.pack(pady=20)

root.mainloop()
```

# TTKBootstrap

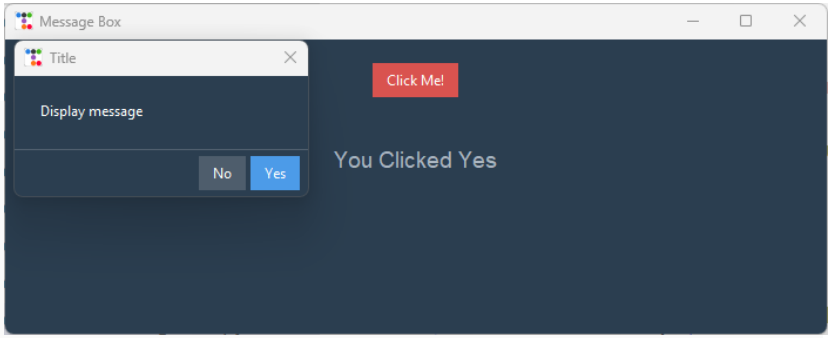
- Treeview




```
root = ttk.Window(themename='superhero')
root.title("Tree View")
root.geometry('400x250')

columns = ("first_name", "last_name", "email")
my_tree = ttk.Treeview(root, bootstyle="success", columns=columns,
                       show="headings")
my_tree.pack(pady=20)

my_tree.heading("first_name", text="First Name")
my_tree.heading("last_name", text="Last Name")
my_tree.heading("email", text="Email Address")

contacts = []
for n in range(1,20):
    contacts.append((f'First {n}', f'Last {n}', f'email{n}@address.com'))

for contact in contacts:
    my_tree.insert('',END, values=contact)

root.mainloop()
```

# TTKBootstrap

- MessageBox




```
from tkinter import *
import ttkbootstrap as ttk
from ttkbootstrap.dialogs import Messagebox

root = ttk.Window(themename='superhero')
root.title("Message Box")
root.geometry('500x250')

def clicker():
    mb = Messagebox.yesno("Display message", "Title")
    my_label.config(text=f'You Clicked {mb}')

my_button = ttk.Button(root, text="Click Me!", bootstyle="danger", command=clicker)
my_button.pack(pady=20)

my_label = ttk.Label(root, text="", bootstyle="light", font=("Helvetica",14))
my_label.pack(pady=20)

root.mainloop()
```

ทดลองเปลี่ยนเป็น ok, okcancel, show_info, show_error, show_question, show_warning, yesnocancel, retrycancle

# TTKBootstrap

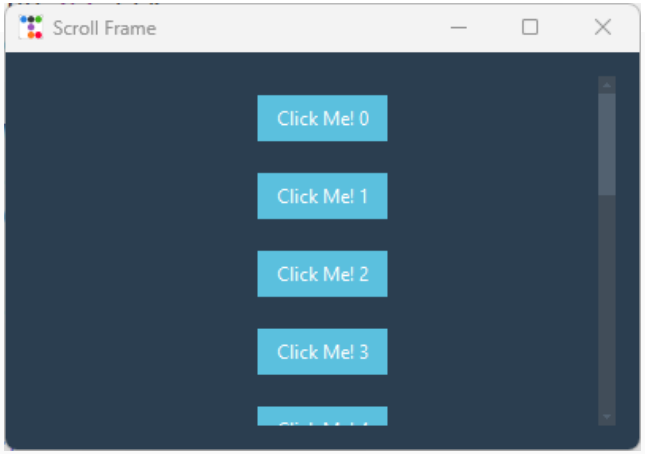
- Color picker

```
from tkinter import *
import ttkbootstrap as ttk
from ttkbootstrap.dialogs.colorchooser import ColorChooserDialog

root = ttk.Window(themename='superhero')
root.title("Color Picker Box")
root.geometry('500x250')

def color_picker():
    my_color = ColorChooserDialog()
    my_color.show()
    colors = my_color.result
    my_label.config(text=colors.hex) # .hsl .rgb
    root.configure(background=colors.hex)

my_button = ttk.Button(root, text="Click Me!", bootstyle="danger", command=color_picker)
my_button.pack(pady=20)

my_label = ttk.Label(root, text="", bootstyle="light", font=("Helvetica",14))
my_label.pack(pady=20)

root.mainloop()
```

# TTKBootstrap

- Scroll frame ใช้ประโยชน์ได้มาก กรณีข้อมูลยาวเกินหน้าจอ

```
from tkinter import *
import ttkbootstrap as ttk
from ttkbootstrap.scrolled import ScrolledFrame

root = ttk.Window(themename='superhero')
root.title("Scroll Frame")
root.geometry('400x250')

my_frame = ScrolledFrame(root, autohide=False)
my_frame.pack(pady=15, padx=15, fill=BOTH, expand=YES)

for x in range(21):
    ttk.Button(my_frame, text=f"Click Me! {x}", bootstyle="info").pack(pady=10)

root.mainloop()
```

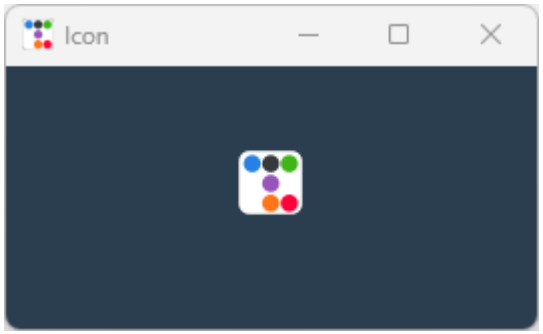


# TTKBootstrap

- แสดง Icon

```
from tkinter import *
import ttkbootstrap as ttk
from ttkbootstrap.icons import Icon

root = ttk.Window(themename='superhero')
root.title("Icon")
root.geometry('400x250')

img = PhotoImage(data=Icon.icon)

my_label = ttk.Label(image=img)
my_label.pack(pady=40)
```

| Name | Type | Description |
|---|---|---|
| `icon` | `str` | The ttkbootstrap icon. |
| `error` | `str` | An error image. |
| `warning` | `str` | A warning image. |
| `question` | `str` | A question image. |
| `info` | `str` | An info image. |




# For your attention