

บทที่ 8.

GUI

**** install package เพื่อใช้สร้าง table**

`pip3 install tkintertable`

`pip3 install pandastable`

`pip3 install xlwt`

ไอลเคียการสร้าง GUI โดยใช้ **tkinter**

- สร้าง root สามารถกำหนดขนาดด้วยฟังก์ชัน `root.geometry("widthxheight")`
- สร้าง widget ที่ต้องการเพื่อวางบน GUI เช่น Label, Button, Entry
- วาง widget บน GUI ด้วยฟังก์ชัน `pack()` หรือ `place()`
- กำหนดการทำงานของ widget โดยสร้าง callback function เช่นเมื่อกดปุ่มต้องการให้ทำงาน
- widget และฟังก์ชันเพื่อ `gettext`, `settext` และ `delete` ที่ควรรู้จัก

	get text	set text	delete
Button	<code>str = b.cget('text')</code>	<code>b.config(text= '...')</code>	Think ☺
Label	<code>str = l.cget('text')</code>	<code>l.config(text= '...')</code>	Think ☺
Entry	<code>e.get()</code>	<code>e.insert(0, '...')</code>	<code>e.delete(0, 'end')</code>

Color chart

AntiqueWhite1	IndianRed2	MistyRose2	SlateGray1	chocolate2	goldenrod3	gray91	magenta3	plum4	tomato2
AntiqueWhite2	IndianRed3	MistyRose3	SlateGray2	chocolate3	goldenrod4	gray92	magenta4	powder blue	tomato3
AntiqueWhite3	IndianRed4	MistyRose4	SlateGray3	coral	gray	gray51	maroon	purple	tomato4
AntiqueWhite4	LavenderBlush2	NavajoWhite2	SlateGray4	coral1	gray52	gray94	maroon1	purple1	turquoise
CadetBlue1	LavenderBlush3	NavajoWhite3	SpringGreen2	coral2	gray10	gray53	maroon2	purple2	turquoise1
CadetBlue2	LavenderBlush4	NavajoWhite4	SpringGreen3	coral3	gray11	gray54	maroon3	purple3	turquoise2
CadetBlue3	LemonChiffon2	OliveDrab1	SpringGreen4	coral4	gray12	gray55	maroon4	purple4	turquoise3
CadetBlue4	LemonChiffon3	OliveDrab2	SteelBlue1	cornflower blue	gray13	gray56	medium aquamarine	red	turquoise4
DarkGoldenrod1	LemonChiffon4	OliveDrab4	SteelBlue2	cornsilk2	gray14	gray57	green yellow	red2	violet red
DarkGoldenrod2	LightBlue1	OrangeRed2	SteelBlue3	cornsilk3	gray15	gray58	green2	medium orchid	red3
DarkGoldenrod3	LightBlue2	OrangeRed3	SteelBlue4	cornsilk4	gray16	gray59	green3	medium purple	red4
DarkGoldenrod4	LightBlue3	OrangeRed4	VioletRed1	cyan	gray17	gray60	green4	medium sea green	rosy brown
DarkOliveGreen1	LightBlue4	PaleGreen1	VioletRed2	cyan2	gray18	gray61	honeydew2	medium slate blue	royal blue
DarkOliveGreen2	LightCyan2	PaleGreen2	VioletRed3	cyan3	gray19	gray62	honeydew3	medium spring green	saddle brown
DarkOliveGreen3	LightCyan3	PaleGreen3	VioletRed4	cyan4	gray20	gray63	honeydew4	medium turquoise	salmon
DarkOliveGreen4	LightCyan4	PaleGreen4	alice blue	dark goldenrod	gray21	gray64	hot pink	medium violet red	salmon1
DarkOrange1	LightGoldenrod1	PaleTurquoise1	antique white	dark green	gray22	gray65	indian red	medium blue	salmon2
DarkOrange2	LightGoldenrod2	PaleTurquoise2	aquamarine	dark khaki	gray23	gray66	ivory2	mint cream	salmon3
DarkOrange3	LightGoldenrod3	PaleTurquoise3	aquamarine2	dark olive green	gray24	gray67	ivory3	misty rose	salmon4
DarkOrange4	LightGoldenrod4	PaleTurquoise4	aquamarine4	dark orange	gray25	gray68	ivory4	navajo white	sandy brown
DarkOrchid1	LightPink1	PaleVioletRed1	azure	dark orchid	gray26	gray69	khaki	khaki1	sea green
DarkOrchid2	LightPink2	PaleVioletRed2	azure2	dark salmon	gray27	gray70	khaki2	old lace	seashell2
DarkOrchid3	LightPink3	PaleVioletRed3	azure3	dark sea green	gray28	gray71	khaki3	olive drab	seashell3
DarkOrchid4	LightPink4	PaleVioletRed4	azure4	dark slate blue	gray29	gray72	khaki4	orange	seashell4
DarkSeaGreen1	LightSalmon2	PeachPuff2	bisque	dark slate gray	gray30	gray73	lavender	orange red	sienna1
DarkSeaGreen2	LightSalmon3	PeachPuff3	bisque2	dark turquoise	gray31	gray74	lavender blush	orange2	sienna2
DarkSeaGreen3	LightSalmon4	PeachPuff4	bisque3	dark violet	gray32	gray75	lawn green	orange3	sienna3
DarkSeaGreen4	LightSkyBlue1	RosyBrown1	bisque4	deep pink	gray33	gray76	lemon chiffon	orange4	sienna4
DarkSlateGray1	LightSkyBlue2	RosyBrown2	blanched almond	deep sky blue	gray34	gray77	light blue	orchid1	sky blue
DarkSlateGray2	LightSkyBlue3	RosyBrown3	blue	dim gray	gray35	gray78	light blue	orchid2	slate blue
DarkSlateGray3	LightSkyBlue4	RosyBrown4	blue violet	dodger blue	gray36	gray79	light coral	orchid3	slate gray
DarkSlateGray4	LightSteelBlue1	RoyalBlue1	blue2	firebrick1	gray37	gray80	light cyan	orchid4	snow
DeepPink2	LightSteelBlue2	RoyalBlue2	blue3	firebrick2	gray38	gray81	light goldenrod	pale goldenrod	snow2
DeepPink3	LightSteelBlue3	RoyalBlue3	brown1	firebrick3	gray39	gray82	light goldenrod yellow	pale green	snow3
DeepPink4	LightSteelBlue4	RoyalBlue4	brown2	firebrick4	gray40	gray83	light grey	pale turquoise	snow4
DeepSkyBlue2	LightYellow2	SeaGreen1	brown3	floral white	gray41	gray84	light pink	pale violet red	spring green
DeepSkyBlue3	LightYellow3	SeaGreen2	brown4	forest green	gray42	gray85	light salmon	papaya whip	steel blue
DeepSkyBlue4	LightYellow4	SeaGreen3	burlywood1	gainsboro	gray43	gray86	light sea green	peach puff	tan1
DodgerBlue2	MediumOrchid1	SkyBlue1	burlywood2	ghost white	gray44	gray87	light sky blue	pink	tan2
DodgerBlue3	MediumOrchid2	SkyBlue2	burlywood3	gold	gray45	gray88	light slate blue	pink1	tan4
DodgerBlue4	MediumOrchid3	SkyBlue3	burlywood4	gold2	gray46	gray89	light slate gray	pink2	thistle
HotPink1	MediumOrchid4	SkyBlue4	cadet blue	gold3	gray47	gray90	light steel blue	pink3	thistle1
HotPink2	MediumPurple1	SlateBlue1	chartreuse2	gold4	gray48	gray91	light yellow	pink4	thistle2
HotPink3	MediumPurple2	SlateBlue2	chartreuse3	goldenrod	gray49	gray92	lime green	plum1	thistle3
HotPink4	MediumPurple3	SlateBlue3	chartreuse4	goldenrod1	gray50	gray93	linen	plum2	thistle4
IndianRed1	MediumPurple4	SlateBlue4	chocolate1	goldenrod2	gray51	gray94	magenta2	plum3	tomato

8.1 Create a GUI using tkinter

8.1.1 สร้าง widget, กำหนด property, ตำแหน่งการวางบน GUI

ตัวอย่าง 8.1 สร้าง Label 3 ตัวบน GUI , วาง widget ด้วย pack()

- สร้าง Label 3 ตัว
- ตัวอย่างการสร้าง Font object
- กำหนด property เช่นขนาด สี
- กำหนด property โดยใช้ config()
- วาง widget บน gui โดยใช้ pack()

```
1 from tkinter import *
2 from tkinter.font import *
3
4 root = Tk()
5
6 l1 = Label(text='Hi, I\'m a simple Label.')
7
8 Hel20 = Font(family='Helvetica', size=20, weight='bold')
9 l2 = Label(text='Hi, I\'m another Label.')
10 l2.config(font=Hel20, fg='plum3')
11
12 l3 = Label(
13     text = 'I\'m a huge gray Label.',
14     width = 50,
15     height = 10,
16     fg = 'green',
17     bg = 'lightgray',
18     font = 'Helvetica 30 bold')
19
20 l1.pack()
21 l2.pack()
22 l3.pack()
23 root.mainloop()
```



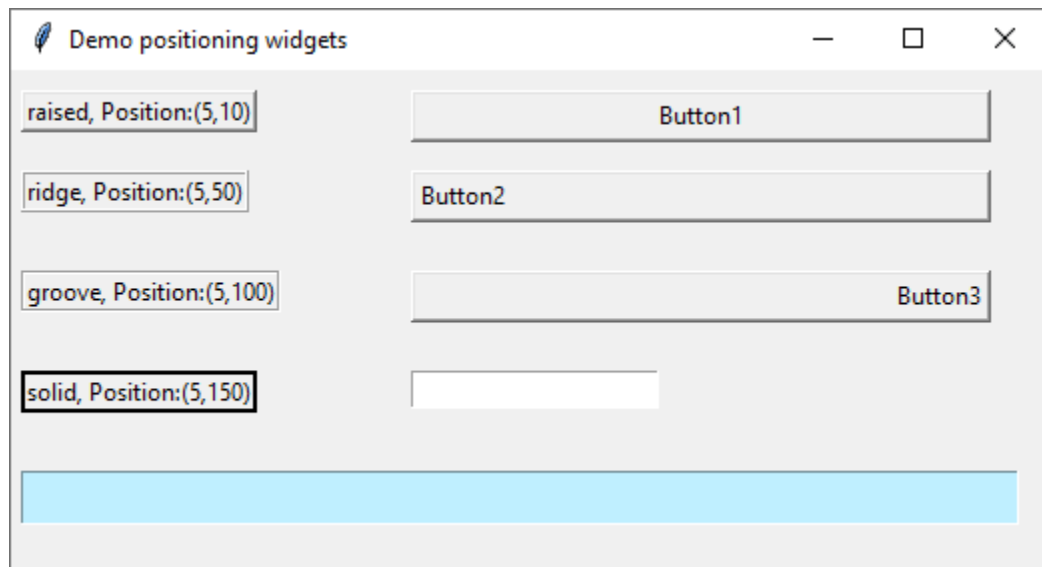
ตัวอย่าง 8.2 กำหนดขนาด GUI, label border style, button, วาง widget ด้วย place()

- การกำหนดขนาด GUI ด้วย geometry()
- root.title()
- การวาด label border รูปแบบต่างๆ
- การวาง widget ที่ตำแหน่งต่างๆด้วย place(x, y)
- การสร้าง Button และ Entry และวางบน GUI ด้วย place(x, y)
- การกำหนด text alignment ให้ Button

```

1  from tkinter import *
2
3  root = Tk()
4  root.geometry('520x250')
5  root.title("Demo positioning widgets")
6
7  l1 = Label(borderwidth=2, relief='raised', text='raised, Position:(5,10)')
8  l2 = Label(borderwidth=2, relief='ridge', text='ridge, Position:(5,50)')
9  l3 = Label(borderwidth=2, relief='groove', text='groove, Position:(5,100)')
10 l4 = Label(borderwidth=2, relief='solid', text='solid, Position:(5,150)')
11 l1.place(x=5, y = 10)
12 l2.place(x=5, y = 50)
13 l3.place(x=5, y = 100)
14 l4.place(x=5, y = 150)
15
16 b1 = Button(text='Button1', width= 40, anchor=CENTER)
17 b2 = Button(text='Button2', width= 40, anchor=W)
18 b3 = Button(text='Button3', width= 40, anchor=E)
19 b1.place(x=200, y = 10)
20 b2.place(x=200, y = 50)
21 b3.place(x=200, y = 100)
22
23 e1 = Entry()
24 e1.place(x=200,y=150)
25 e2 = Entry(bg = 'LightBlue1', fg = 'green4', font = 'none 15',width=45)
26 e2.place(x=5,y=200)
27
28 root.mainloop()

```



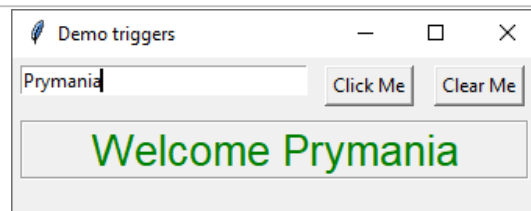
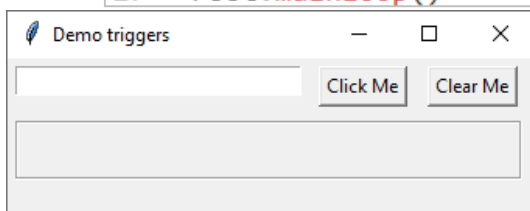
**** ทำ Lab 1: สร้าง GUI**

8.1.2 ใส่ trigger ให้ widget

ตัวอย่าง 8.3 ใส่ trigger ให้ Button เพื่อแสดง และลบข้อความ

- ใส่ trigger ให้ Button โดยกำหนดเป็น parameter command
- ใส่ trigger ให้ Button โดยใช้ฟังก์ชัน config(command)
- การ get text จาก Entry โดยใช้ฟังก์ชัน get()
- การ set text และ clear text ให้ label

```
1  from tkinter import *
2  from tkinter.font import *
3
4  def nameCallback():
5      name = nameEntry.get()
6      nameLabel.config(text = "Welcome " + name)
7
8  def clearEntry():
9      nameEntry.delete(0, 'end')
10     nameLabel.config(text='')
11
12     root = Tk()
13     root.geometry("340x100")
14     nameEntry = Entry(width=30)
15     nameButton = Button(text='Click Me', command = nameCallback)
16     clearButton = Button(text='Clear Me')
17     nameLabel = Label(fg='green', width=20,
18                      borderWidth=2, relief='groove', font = 'none 20')
19     clearButton.config(command=clearEntry)
20
21     nameEntry.place(x=5, y=5)
22     nameButton.place(x=200, y=5)
23     clearButton.place(x=270, y=5)
24     nameLabel.place(x=5, y=40)
25
26     root.title("Demo triggers")
27     root.mainloop()
```

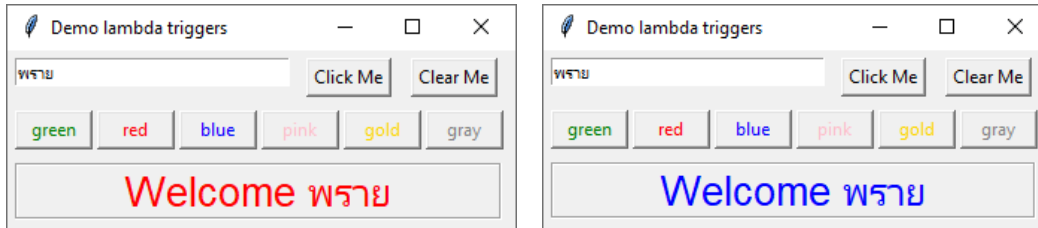


**** ทำ Lab 2: ใส่ trigger ให้ button**

8.1.3 ใช้ lamda เพื่อเรียก trigger function เดียวกัน

ตัวอย่าง 8.4 จากตัวอย่าง 8.3 เพิ่ม Button เพื่อเปลี่ยนสีข้อความ

- ใช้ lambda เพื่อส่ง parameter ที่แตกต่างกันไป trigger function เดียวกัน

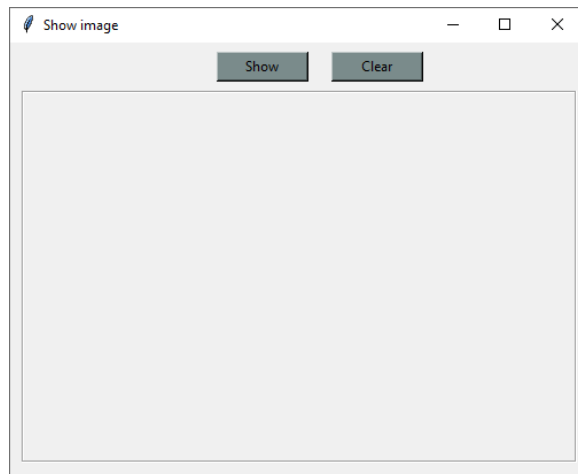
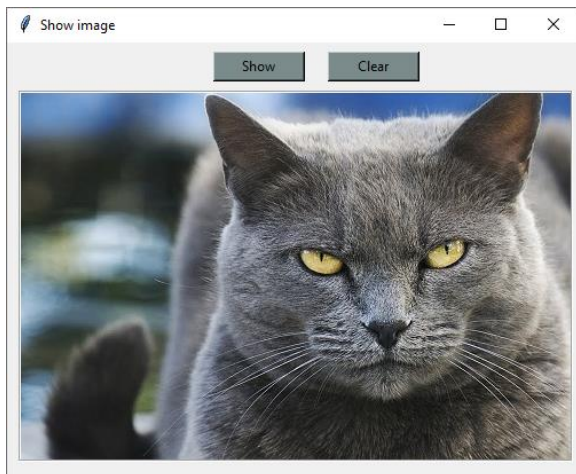


```
1 from tkinter import *
2 from tkinter.font import *
3
4 def nameCallback():
5     name = nameEntry.get()
6     nameLabel.config(text = "Welcome "+ name)
7
8 def clearEntry():
9     nameEntry.delete(0,'end')
10    nameLabel.config(text='')
11
12 def changeColor(colorText):
13     nameLabel.config(fg=colorText)
14
15 root = Tk()
16 root.geometry("340x120")
17 nameEntry = Entry(width=30)
18 nameButton = Button(text='Click Me', command = nameCallback)
19 clearButton = Button(text='Clear Me')
20 print(clearButton)
21 nameLabel = Label(fg='green', width=20,
22                 borderWidth=2, relief='groove', font = 'none 20')
23
24 clearButton.config(command=clearEntry)
25
26 greenButton= Button(text='green',fg='green',width=6, command = lambda text='green':changeColor('green'))
27 redButton = Button(text='red', fg='red', width=6, command = lambda text='red':changeColor('red'))
28 blueButton = Button(text='blue', fg='blue', width=6, command = lambda text='blue':changeColor('blue'))
29 pinkButton = Button(text='pink', fg='pink', width=6, command = lambda text='pink':changeColor('pink'))
30 goldButton = Button(text='gold', fg='gold', width=6, command = lambda text='gold':changeColor('gold'))
31 grayButton = Button(text='gray', fg='gray', width=6, command = lambda text='gray':changeColor('gray'))
```


8.1.4 แสดงรูปภาพด้วย Canvas

ตัวอย่าง 8.5 แสดงรูปภาพบน canvas

- สร้าง image โดยใช้ ImageTk, Image
`img = ImageTk.PhotoImage(file= '.....')`
- กดปุ่ม Show เพื่อแสดงภาพแมวบน Canvas โดยใช้ฟังก์ชัน
`canvas.create_image(x,y, anchor = NW, image =img)`
`canvas.image = img`
- กดปุ่ม Clear เพื่อลบภาพแมวจาก Canvas
`canvas.delete('all')`

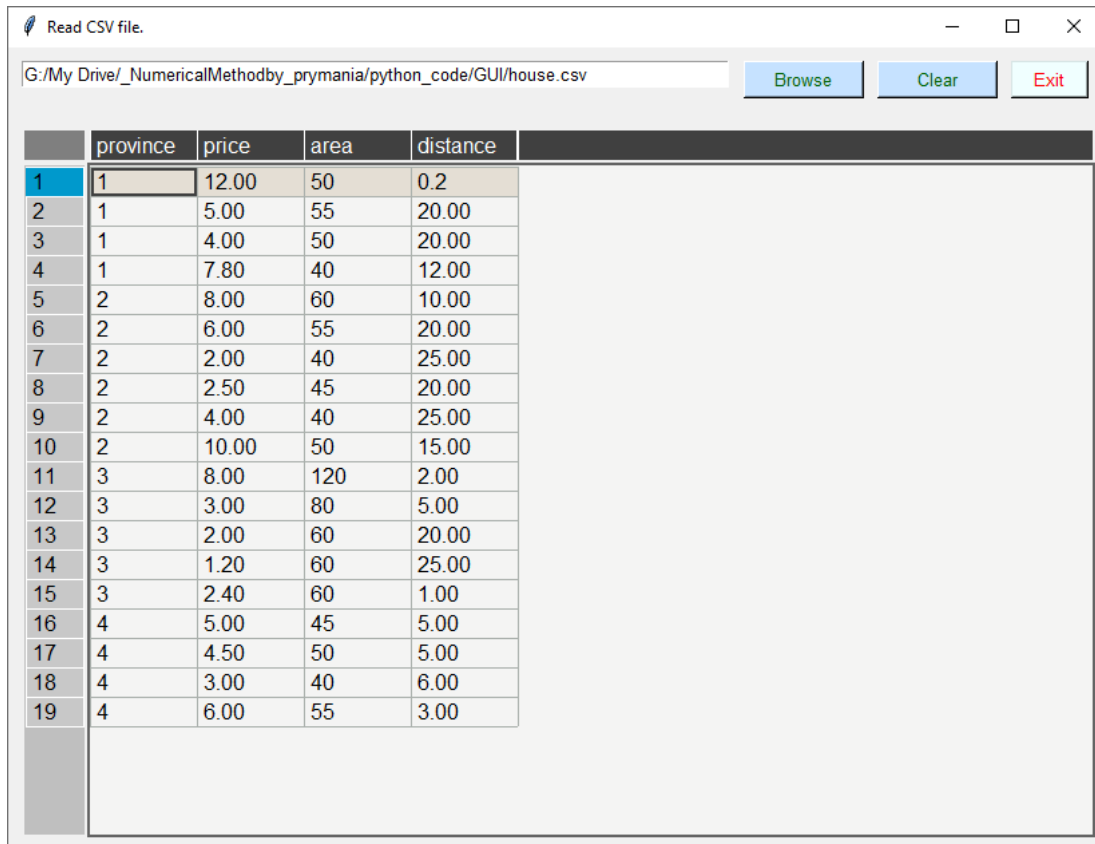


**** ทำ Lab 3: ใส่ trigger ให้ button เพื่อแสดงรูปภาพบน canvas ใช้ lambda**

8.1.5 อ่านข้อมูลจากไฟล์ csv

ตัวอย่าง 8.6 อ่านข้อมูลจากไฟล์ csv และแสดงข้อมูลบน pandastable.Table

- อ่านไฟล์ .csv โดยใช้ pandas
- แสดงข้อมูลที่อ่านจากไฟล์บน pandastable และวาง Table บน Frame



	province	price	area	distance
1	1	12.00	50	0.2
2	1	5.00	55	20.00
3	1	4.00	50	20.00
4	1	7.80	40	12.00
5	2	8.00	60	10.00
6	2	6.00	55	20.00
7	2	2.00	40	25.00
8	2	2.50	45	20.00
9	2	4.00	40	25.00
10	2	10.00	50	15.00
11	3	8.00	120	2.00
12	3	3.00	80	5.00
13	3	2.00	60	20.00
14	3	1.20	60	25.00
15	3	2.40	60	1.00
16	4	5.00	45	5.00
17	4	4.50	50	5.00
18	4	3.00	40	6.00
19	4	6.00	55	3.00

ขั้นตอนการเขียน source code

1. import package ที่ใช้ในงานนี้

```
1 import pandas as pd
2 from tkinter import *
3 from tkinter import filedialog
4 from tkinter import messagebox as msg
5 from pandastable import Table
```

- tk.filedialog เพื่อเปิด file dialog
- tk.messagebox เพื่อ pop up message กรณี error
- pd, pandastable เพื่ออ่านไฟล์ และแสดงข้อความ

2. สร้าง GUI ประกอบด้วย

- Entry เพื่อแสดงชื่อไฟล์
- Button 3 ตัว เพื่อ Browse file, ลบข้อมูลที่แสดง และออกจากโปรแกรม
- Frame เพื่อแสดง pandastable
- * เราใส่ command ให้ browseButton และ clearButton ดังนั้นสร้าง callback function หลอกๆไว้

```

39 root = Tk()
40 root.title('Read CSV file.')
41
42 filename_entry = Entry(width =75,font='none 10')
43
44 browseButton = Button(
45     text = 'Browse', width =10,
46     font = ('none 10'),
47     bg = 'SlateGray1',
48     fg = 'dark green',
49     command = readCSV)
50 clearButton = Button(
51     text = 'Clear', width =10,
52     font = ('none 10'),
53     bg = 'SlateGray1',
54     fg = 'dark green',
55     command = clearEntry)
56 exit_button = Button(
57     text = 'Exit', width =6,
58     font = ('none 10'),
59     bg = 'azure',
60     fg = 'red',
61     command = root.destroy)
62 frame =Frame()
63
64 filename_entry.place(x=10, y = 10)
65 browseButton.place(x=550, y = 10)
66 clearButton.place(x=650, y = 10)
67 exit_button.place(x=750, y = 10)
68 frame.place(x=10, y = 60)
69
70 root.geometry('820x600')
71 root.mainloop()

```

3. สร้าง callback function คือ

- **readCSV()** เพื่ออ่านไฟล์ และแสดงข้อมูลบน Table โดยใช้
 - tk.filedialog.askopenfilename() เพื่อเปิด file dialog
 - ** **filetypes** ต้องการข้อมูลแบบ tuple แต่เราต้องการให้เปิดได้เฉพาะไฟล์ .csv ประเภทเดียวเท่านั้น ดังนั้นเราใส่ , หลอกๆไว้ เพื่อให้ข้อมูลเป็น tuple
 - pandas เพื่ออ่านไฟล์ในรูปแบบ dataframe
 - วาง Table บน frame โดย table.show()
- **clearEntry()** เพื่อลบการแสดงผลข้อมูลบน Table
 - ลบ Table ออกจาก frame โดยใช้ฟังก์ชัน destroy()

```

9  def readCSV():
10     try:
11         file_name = filedialog.askopenfilename(
12             initialdir = '.',
13             title = 'Select csv/ excel file',
14             filetypes = (('csv file', '*.csv'), ))
15
16         df = pd.read_csv(file_name)
17         if(len(df)!=0):
18             filename_entry.delete(0,END)
19             filename_entry.insert(0,file_name)
20
21             table = Table(frame, dataframe=df,read_only=True, width =720, height=500)
22             table.show()
23         else:
24             msg.showinfo('No data records!', 'No records.')
25
26     except FileNotFoundError as error:
27         filename_entry.delete(0,END)
28
29  def clearEntry():
30     filename_entry.delete(0,END)
31     for widget in frame.winfo_children():
32         widget.destroy()

```

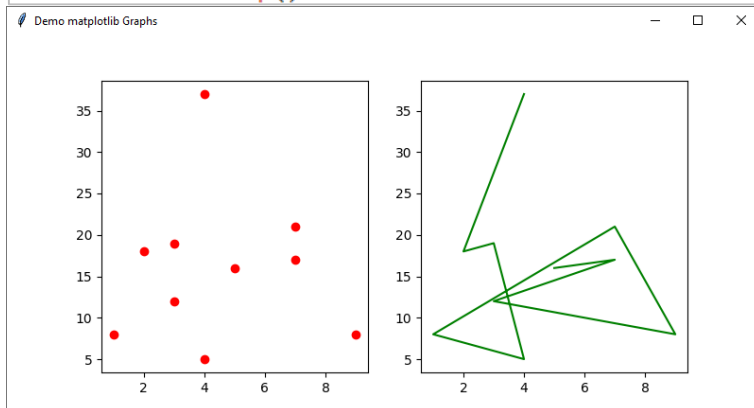
8.1.6 แสดงกราฟบน GUI

ตัวอย่าง 8.7 วาด Matplotlib graph 2 ตัว บน GUI

- random integer x and y
- plot as scatter and line graph
- plot two graphs on GUI using
 - `fig = matplotlib.Figure`
 - `canvas = matplotlib.backends.backend_tkagg.FigureCanvasTkAgg`
 - `canvas.get_tk_widget().pack()`

```
1 import numpy as np
2 from tkinter import *
3 import matplotlib.pyplot as plt
4 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
5
6 root= Tk()
7 root.title("Demo matplotlib Graphs")
8
9 x = np.random.randint(0,10,10)
10 y = np.random.randint(0,50,10)
11
12 fig = plt.Figure(figsize=(8,8), dpi=100)
13
14 plot1 = fig.add_subplot(121)
15 plot1.plot(x,y,'ro')
16
17 plot2 = fig.add_subplot(122)
18 plot2.plot(x,y,'g-')
19
20 canvas = FigureCanvasTkAgg(fig, master = root)
21 canvas.get_tk_widget().pack()
22
23 root.geometry('800x400')
24 root.mainloop()
```

1. สร้าง fig
2. สร้าง plot1 เป็น subplot บน fig
3. วาดกราฟลง plot1
4. วาง fig บน canvas
5. วาง canvas บน GUI



8.1.7 แสดงกราฟจาก DataFrame บน GUI

ตัวอย่าง 8.8 วาด graph โดยการอ่านข้อมูลจาก DataFrame แสดงเป็นกราฟแท่ง กราฟเส้น และ จุด

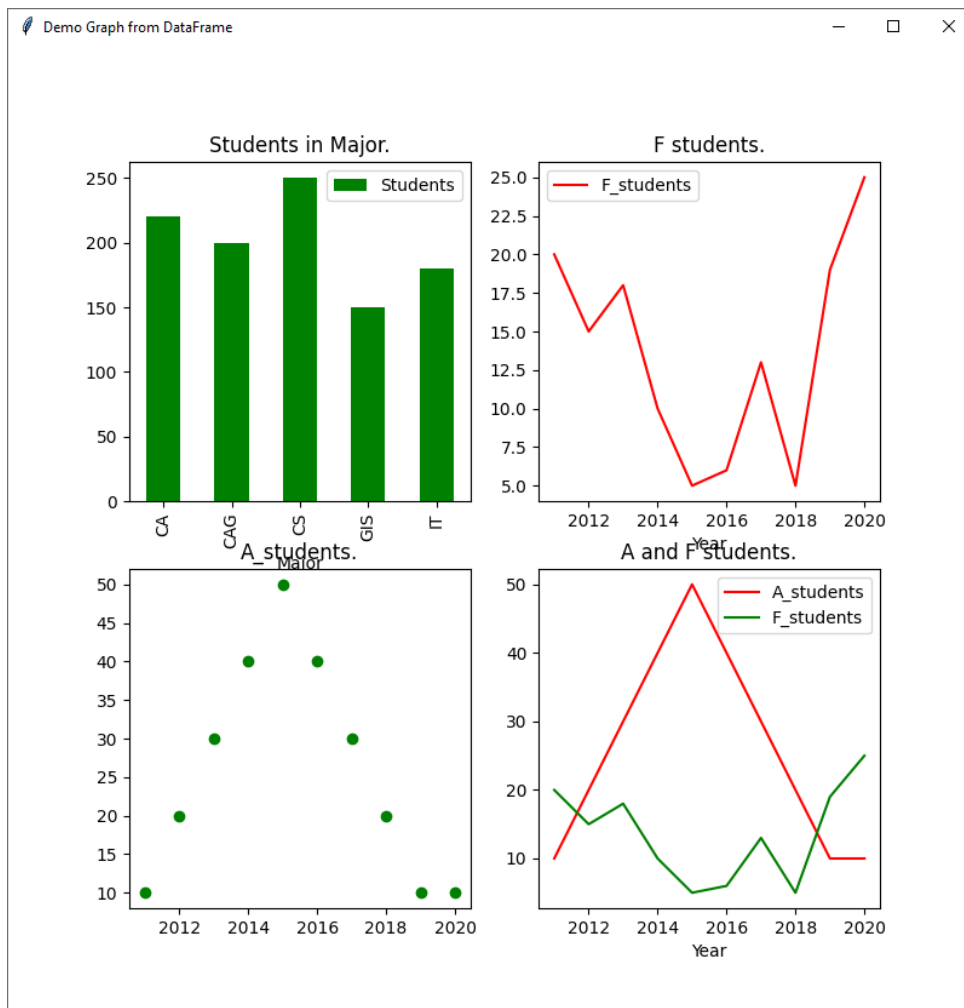
- วาดกราฟ 4 กราฟลงบน subplot(22x)
- อ่านข้อมูลจาก DataFrame เพื่อวาดกราฟ
- ใช้ groupby(column_name).sum() เพื่อกรู๊ปข้อมูลที่ต้องการวาด
- ใช้ DataFrame ในการวาดกราฟบน Figure

```
1 from tkinter import *
2 import pandas as pd
3 import matplotlib.pyplot as plt
4 from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
5
6 df1 = pd.DataFrame({
7     'Major': ['CS', 'IT', 'CAG', 'CA', 'GIS'],
8     'Students': [250, 180, 200, 220, 150]})
9
10 df2 = pd.DataFrame({
11     'Year': [2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020],
12     'F_students': [20, 15, 18, 10, 5, 6, 13, 5, 19, 25],
13     'A_students': [10, 20, 30, 40, 50, 40, 30, 20, 10, 10]})
```

```

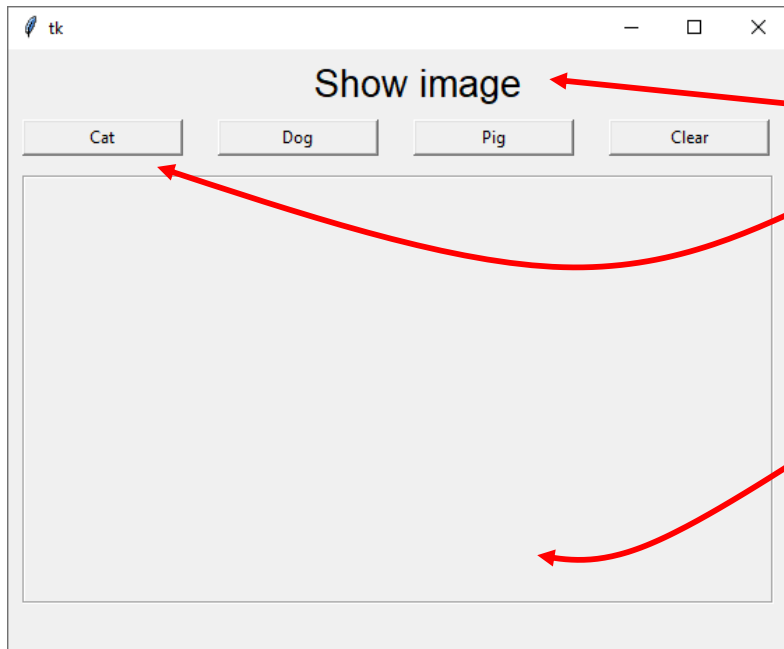
15 root= Tk()
16 root.title("Demo Graph from DataFrame")
17
18 fig = plt.Figure(figsize=(8,8), dpi=100)
19
20 plot1 = fig.add_subplot(221)
21 data = df1[['Major', 'Students']].groupby('Major').sum()
22 data.plot(kind='bar', color='green', legend=True, ax=plot1)
23 plot1.set_title('Students in Major.')
24
25 plot2 = fig.add_subplot(222)
26 data = df2[['Year', 'F_students']].groupby('Year').sum()
27 data.plot(kind='line', color='red', legend=True, ax=plot2)
28 plot2.set_title('F students.')
29
30 plot3 = fig.add_subplot(223)
31 data = df2[['Year', 'A_students']].groupby('Year').sum()
32 plot3.scatter(df2['Year'], df2['A_students'], color='green')
33 plot3.set_title('A_students.')
34
35 plot4 = fig.add_subplot(224)
36 data_a = df2[['Year', 'A_students']].groupby('Year').sum()
37 data_f = df2[['Year', 'F_students']].groupby('Year').sum()
38 data_a.plot(kind='line', color='red', legend=True, ax=plot4)
39 data_f.plot(kind='line', color='green', legend=True, ax=plot4)
40 plot4.set_title('A and F students.')
41
42 canvas = FigureCanvasTkAgg(fig, master = root)
43 canvas.get_tk_widget().pack()
44
45 root.geometry('800x800')
46 root.mainloop()

```



8.2 แบบฝึกหัด

1. สร้าง GUI ประกอบด้วย widget ต่างๆ ตามภาพ *จะใช้ GUI นี้เพื่อทำข้อ 3. *



Label แสดงข้อความ Show image

Button 4 ตัว ตั้งชื่อ

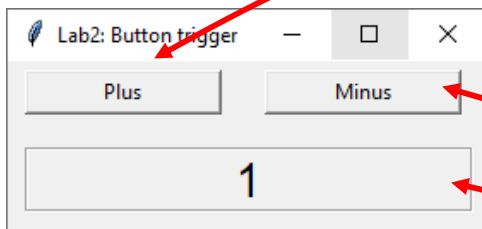
catButton, dogButton, pigButton
และ clearButton

Label ตั้งชื่อ outputLabel

วาง widgets ด้วยคำสั่ง place(x, y)

2. สร้าง GUI ประกอบด้วย widget และการทำงานตามภาพ

- การแปลง int --> string: `number = (int)(text)`
- การแปลง string --> int: `text = str(number)`

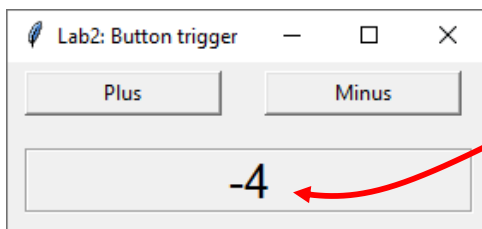


plusButton

minusButton

outputLabel เริ่มต้นแสดงเลข 1

เมื่อกดปุ่ม plusButton หรือ
minusButton ให้เพิ่มหรือลบค่าตัวเลขบน
outputLabel



3. จาก GUI ใน Lab 1. ให้ใส่ trigger ให้ button โดย

- เมื่อคลิก button ให้แสดงภาพของสัตว์ตามชื่อ button บน outputLabel
- เมื่อคลิก clear button ให้ลบรูปออกจาก outputLabel
- ใช้ lambda ในการ pass parameter เพื่อใช้ว่า Button ตัวใดถูกกด

การใช้ lambda เพื่อ pass parameter เพราะปุ่ม cat, dog และ pig ต้องการ trigger function เดียวกัน คือเมื่อกดปุ่มแล้วอ่าน text บนปุ่ม และแสดงรูปภาพของปุ่มนั้น ดังนั้นเราต้องส่ง parameter เป็น string ข้อความบนปุ่ม

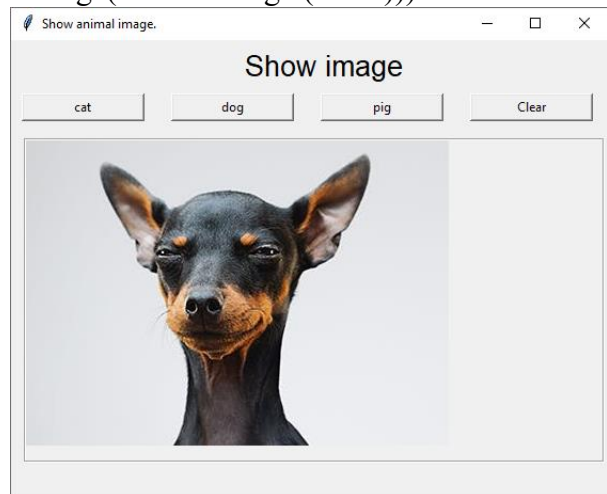
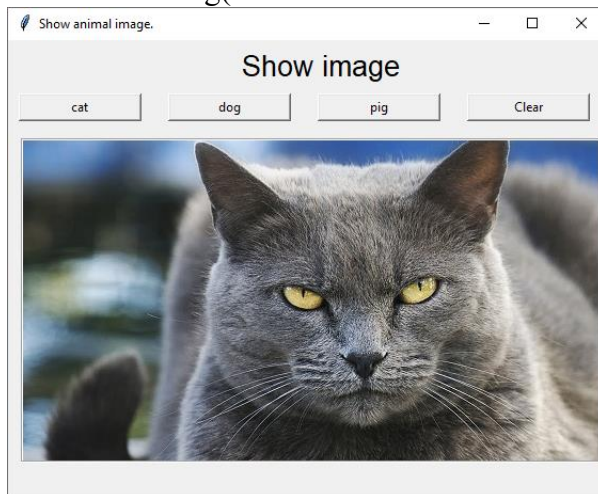
การ **define callback function**

```
def showImage(image_name):
```

```
.....
```

การสั่ง **trigger** ให้ **Button**

```
catButton.config(command=lambda m= 'cat':showImage(catButton.cget('text')))
```



4. อ่านข้อมูล smart phone shared market จากไฟล์ phone.csv แสดงข้อมูลการขายเป็นกราฟเส้นของแต่ละยี่ห้อ และ pie graph แสดง share market รวมทุกปีของแต่ละยี่ห้อ

ข้อมูลในไฟล์ smartphone.csv

Year	Samsung	Huawei	Xiaomi	Apple	Oppo
2016	22	11	8	14	7
2017	19	15	9	11	8
2018	18	15	6	17	8
2019	18	14	8	18	8
2020	22	14	13	11	8

การแสดงผล

