



# 圖形化介面

Tkinter

# Tkinter介紹

---

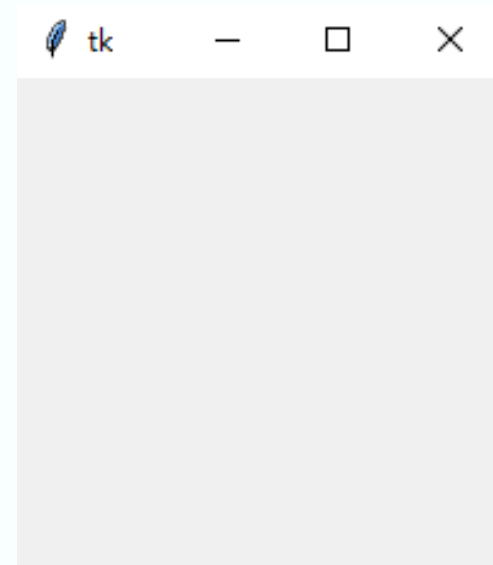
- ◆ GUI開發也叫圖形介面開發，Tkinter是Python內建的圖形介面開發庫，它簡單實用，速度很快
- ◆ Tkinter支援跨平台，可以在Windows、Mac、Linux、UNIX上開發執行
- ◆ Tkinter的全名為Tk interface
- ◆ 除了Tkinter之外，還有其他幾種GUI庫可代替，例如PyQt、Pygame、PySide等
- ◆ Python2時用Tkinter，Python3更改為tkinter

# 建立tkinter視窗

- ◆ Tkinter是Python內建函式庫，只要匯入即可使用
- ◆ 使用Tk()方法建立Tkinter視窗
- ◆ 使用mainloop()放在主迴圈中一直執行，直到關閉視窗才會停止運作

```
1 import tkinter as tk
2
3 window = tk.Tk()
4 window.mainloop()
```

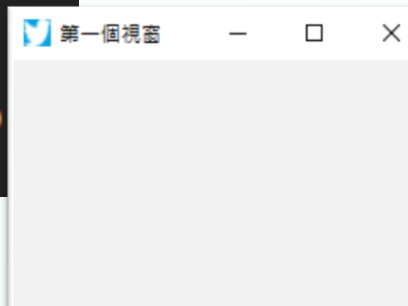
```
1 from tkinter import *
2
3 window = Tk()
4 window.mainloop()
```



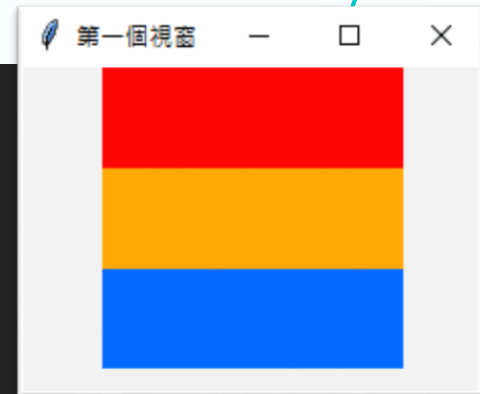
# 建立tkinter視窗

- ◆ title()：設定視窗標題
- ◆ geometry()：設定視窗大小，一定要是字串

```
1 from tkinter import *
2 from PIL import Image
3
4 win = Tk()
5 # 設定視窗標題
6 win.title("第一個視窗")
7 # 設定視窗大小為400*300
8 win.geometry("400x300")
9 # 先將png轉成ico
10 icon = Image.open("twitterlogo.png")
11 icon.save("twitterlogo.ico")
12 # 變更視窗icon
13 win.iconbitmap("twitterlogo.ico")
14 win.mainloop()
```



```
1 from tkinter import *
2
3
4 win = Tk()
5 # 設定視窗標題
6 win.title("第一個視窗")
7 # 設定視窗大小為400*300
8 win.geometry("400x300")
9 # 設定三種背景顏色
10 for color in ["red", "orange", "#0066FF"]:
11     Frame(bg=color, height=50, width=150).pack()
12
13 win.mainloop()
```

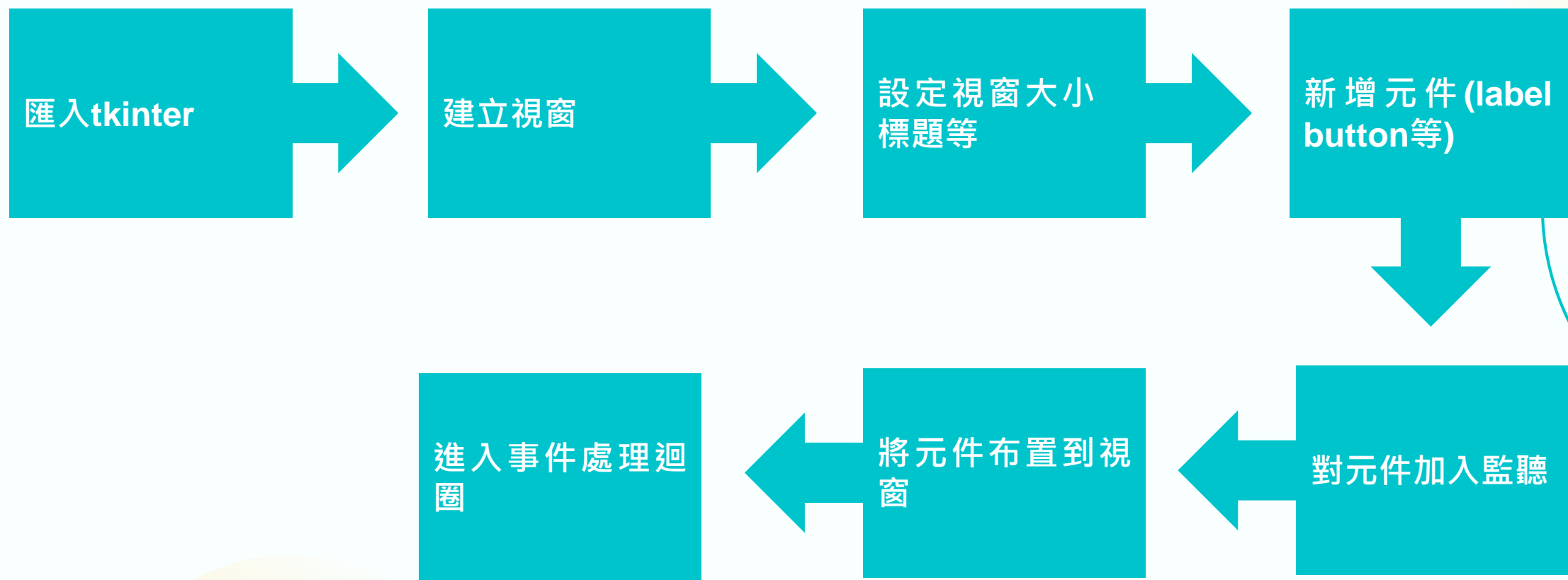


# GUI元件

元件	說明
Label	文字標籤
Button	按鈕
Entry	文字方塊
Text	文字區塊
Listbox	清單方塊
Radiobutton	選項按鈕
Checkbutton	核取按鈕
Canvas	畫布
Menu	功能表

元件	說明
Menubutton	功能表按鈕
Scrollbar	捲軸
Scale	滑杆
Spinbox	調整鈕
messagebox	對話方塊
Frame	視窗

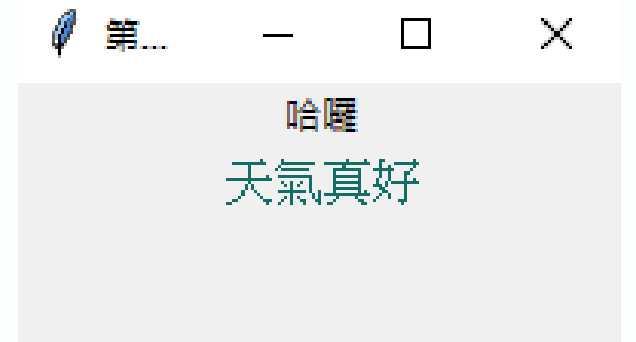
# Tkinter建立流程



# GUI元件 – label(文字標籤)

參數	說明
anchor	擺放位置，可以設定n、s、w、e，預設center
text	文字內容，可以用換行符號(\n)
width	寬度，預設0
height	高度，預設0
bg	背景顏色
fg	文字顏色
font	字型設定，包含大小(px)、粗體(bold)、斜體(italic)
padx、pady	內容和標籤左右(px)、上下(px)的間距，預設1

```
1  from tkinter import *
2
3  win = Tk()
4  # 設定視窗標題
5  win.title("第一個視窗")
6  # 設定視窗大小為200*200
7  win.geometry("200x200")
8  # 設定二個文字標籤
9  label1 = Label(win, text="哈囉")
10 label2 = Label(win, text="天氣真好", font=50, fg="#146E68")
11 # 在視窗上添加剛設定好的二個文字標籤
12 label1.pack()
13 label2.pack()
14 win.mainloop()
```



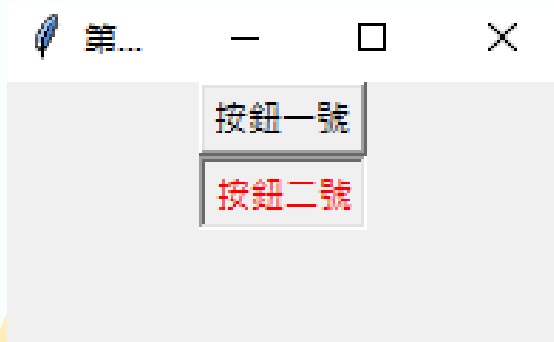
# GUI元件 – Button(按鈕)

參數	說明
anchor	擺放位置，可以設定n、s、w、e，預設center
text	文字內容，可以用換行符號(\n)
width	寬度，預設0
height	高度，預設0
bg	背景顏色
fg	文字顏色
font	字型設定，包含大小(px)、粗體(bold)、斜體(italic)
padx、pady	內容和標籤左右(px)、上下(px)的間距，預設1
image	圖片
command	按下按鈕時要執行的函式
activeforeground	按下按鈕時的按鈕顏色



# GUI元件 – Button(按鈕)

```
1 from tkinter import *
2
3 win = Tk()
4 win.title("第一個視窗")
5 win.geometry("200x200")
6 # 設定二個按鈕
7 btn1 = Button(win, text="按鈕一號")
8 btn2 = Button(win, text="按鈕二號", activeforeground="red")
9 # 在視窗上添加二個按鈕
10 btn1.pack()
11 btn2.pack()
12 win.mainloop()
```



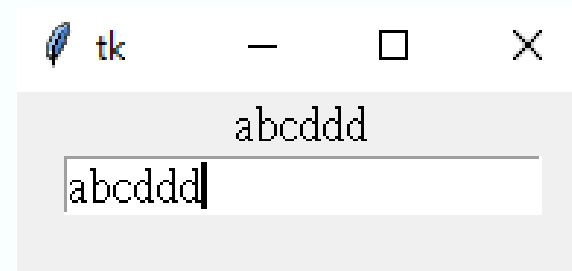
```
1 from tkinter import *
2
3 win = Tk()
4 win.title("第一個視窗")
5 win.geometry("200x200")
6
7 num = 0
8 word = StringVar() # 設定文字變數word
9 word.set(num)      # 設定word的內容
10
11 ▼ def add():
12     global num      # 將num設成全域變數
13     num += 1
14     word.set(num)
15
16 # 設定一個標籤，字體大小30，文字的變數名稱為word
17 label1 = Label(win, textvariable=word, font=30)
18
19 # 設定一個按鈕，按下後會去呼叫add函式
20 btn = Button(win, text="請按我", command=add)
21
22 # 在視窗上添加二個元件
23 label1.pack()
24 btn.pack()
25
26 win.mainloop()
```

# GUI元件 – Entry(文字方塊)

```
1  from tkinter import *
2
3  win = Tk()
4
5  # 設定一個文字標籤跟一個文字區塊
6  label1 = Label(win, text="帳號:", font=20)
7  entry = Entry(win, font=20)
8
9  # 在視窗上添加二個元件
10 label1.pack(side=LEFT)
11 entry.pack(side=RIGHT)
12
13 win.mainloop()
```



```
1  from tkinter import *
2
3  win = Tk()
4  win.geometry("200x200")
5
6  # 設定文字變數
7  word = StringVar()
8  word.set("")
9
10 # 設定一個文字標籤跟一個文字區塊
11 label1 = Label(win, textvariable=word, font=30).pack()
12 entry = Entry(win, textvariable=word, font=30).pack()
13
14 win.mainloop()
```



# GUI元件 – Entry(文字方塊)

```
1  from tkinter import *
2
3  win = Tk()
4  # win.geometry("200x200")
5
6  def add():
7      result.set(num1.get() + num2.get())
8
9  # 設定三個變數為浮數點型態
10 num1 = DoubleVar()
11 num2 = DoubleVar()
12 result = DoubleVar()
13
14 # 文字方塊，第一個輸入的值
15 Entry(win, width=10, textvariable=num1).pack(side=LEFT)
16 # 標籤，顯示「+」
17 Label(win, width=10, text="+").pack(side=LEFT)
18 # 文字方塊，第二個輸入的值
19 Entry(win, width=10, textvariable=num2).pack(side=LEFT)
20 # 標籤，顯示「=」
21 Label(win, width=10, text="=").pack(side=LEFT)
22 # 文字方塊，計算後的結果
23 Entry(win, width=10, textvariable=result).pack(side=LEFT)
24 # 按鈕
25 Button(win, text="計算", command=add).pack()
26
27 win.mainloop()
```



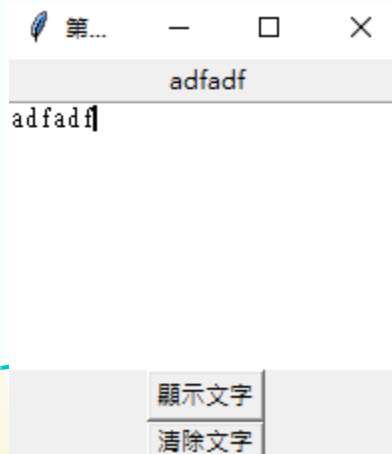
# GUI元件 – Text(文字區塊)

參數	說明
width	寬度，單位是字元數
height	高度，單位是字元數
bg	背景顏色
fg	文字顏色
font	字型設定，包含大小(px)、粗體(bold)、斜體(italic)
padx、pady	內容和標籤左右(px)、上下(px)的間距，預設1
bd	邊框粗細，預設1
relief	邊框樣式，有flat、sunken、raised、ridge、solid，預設flat
textvariable	文字內容的變數名稱

方法	參數	說明
get()	first, last=None	取得文字區塊內容，以小數點方式表示。例如0.0=第一個字元
delete()	first, last=None	刪除指定範圍的內容
see()	index	取得第幾個位置的字元
index()	index	取得指定的索引值
insert()	index,s	在指定索引值位置插入內容

# GUI元件 – Text(文字區塊)

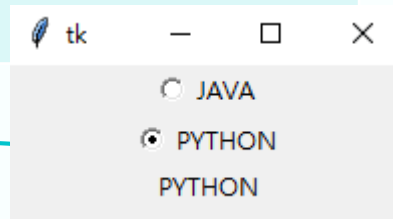
```
1 from tkinter import *
2
3 win = Tk()
4 win.title("第一個視窗")
5 win.geometry("200x200")
6
7 # 設定一個文字區塊，背景淡綠色、字體顏色藍、大小30
8 text = Text(win,bg="#CCFFCC",
9             fg="#0000FF", font=30)
10
11 # 在視窗上添加元件
12 text.pack()
13
14 win.mainloop()
```



```
1 from tkinter import *
2
3 win = Tk()
4 win.title("第一個視窗")
5 win.geometry("200x200")
6
7 word = StringVar() # 設定文字變數word
8 word.set("") # 設定word的內容
9
10 label = Label(win, textvariable=word)
11 label.pack()
12 text = Text(win, height=10)
13 text.pack()
14
15 def show():
16     word.set(text.get("1.0", "end-1c"))
17     # 1.0表示從第一個字元開始取值
18     # end-1c表示從倒數第二個字元取值
19
20 def clear():
21     word.set("")
22     text.delete("1.0", "end")
23
24 btn1 = Button(win, text="顯示文字", command=show).pack()
25 btn2 = Button(win, text="清除文字", command=clear).pack()
26
27 win.mainloop()
```

# GUI元件 – Radiobutton(選項按鈕)

參數	說明
anchor	擺放位置，可以設定n、s、w、e，預設center
text	文字內容，可以用換行符號(\n)
width	寬度，預設0
height	高度，預設0
bg	背景顏色
fg	文字顏色
font	字型設定，包含大小(px)、粗體(bold)、斜體(italic)
padx、pady	內容和標籤左右(px)、上下(px)的間距，預設1
variable	綁定變數
value	選擇按鈕時要回傳變數的值
command	按下按鈕時要執行的函式



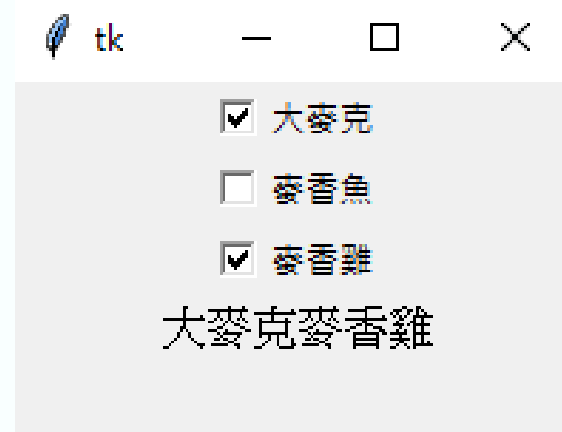
```
1  from tkinter import *
2
3  win = Tk()
4  win.geometry("200x200")
5
6  def show():
7      select = word.get()
8      label.config(text=select)
9
10 # 設定文字變數
11 word = StringVar()
12
13 # 設定第一個選項按鈕JAVA
14 radio1 = Radiobutton(win, text="JAVA", variable=word,
15                      value="JAVA", command=show)
16 # 預設勾選的為JAVA
17 radio1.select()
18 # 設定第二個選項按鈕PYTHON
19 radio2 = Radiobutton(win, text="PYTHON", variable=word,
20                      value="PYTHON", command=show)
21 # 設定一個文字標籤
22 label = Label(win)
23
24 radio1.pack()
25 radio2.pack()
26 label.pack()
27
28 win.mainloop()
```

# GUI元件 – Checkbutton(核取按鈕)

參數	說明
anchor	擺放位置，可以設定n、s、w、e，預設center
text	文字內容，可以用換行符號(\n)
width	寬度，預設0
height	高度，預設0
bg	背景顏色
fg	文字顏色
font	字型設定，包含大小(px)、粗體(bold)、斜體(italic)
padx、pady	內容和標籤左右(px)、上下(px)的間距，預設1
variable	綁定變數
onvalue	選取按鈕時回傳到變數的值，預設0
offvalue	取消選取按鈕時回傳到變數的值，預設0
command	按下按鈕時要執行的函式

# GUI元件 – Checkbutton(核取按鈕)

```
1  from tkinter import *
2
3  win = Tk()
4  win.geometry("200x200")
5
6  def show():
7      text = var1.get() + var2.get() + var3.get()
8      word.set(text)
9
10 # 設定文字變數
11 word = StringVar()
12 word.set("")
13
14 # 第一個選項
15 var1 = StringVar()
16 check1 = Checkbutton(win, text="大麥克", variable=var1,
17                      onvalue="大麥克", offvalue="", command=show)
18 check1.pack()
19
20 # 第二個選項
21 var2 = StringVar()
22 check2 = Checkbutton(win, text="麥香魚", variable=var2,
23                      onvalue="麥香魚", offvalue="", command=show)
24 check2.pack()
25
26 # 第三個選項
27 var3 = StringVar()
28 check3 = Checkbutton(win, text="麥香雞", variable=var3,
29                      onvalue="麥香雞", offvalue="", command=show)
30 check3.pack()
31
32 label = Label(win, textvariable=word, font=30)
33 label.pack()
34
35 win.mainloop()
```

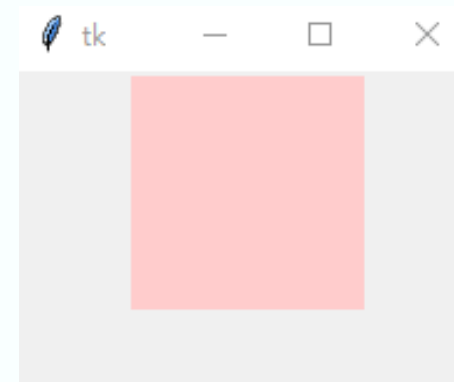




# GUI元件 – Canvas(畫布)

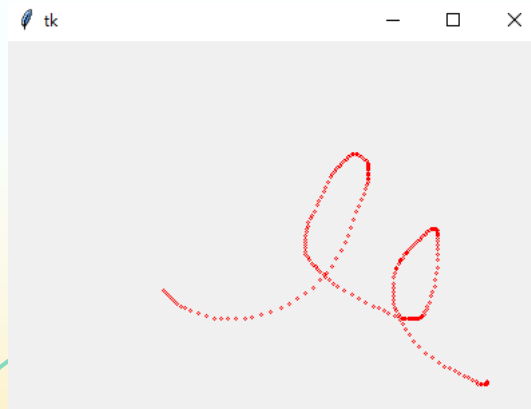
參數	說明
anchor	擺放位置，可以設定n、s、w、e，預設center
width	寬度，預設0
height	高度，預設0
bg	背景顏色
padx、pady	內容和標籤左右(px)、上下(px)的間距，預設1
relief	邊框樣式，有flat、sunken、raised、ridge、solid，預設flat
bd	邊框粗細，預設1

```
1  from tkinter import *
2
3  win = Tk()
4  win.geometry("200x200")
5
6  # 設置畫布大小100*100，顏色粉紅色
7  canvas = Canvas(win, width=100, height=100, bg="#FFCCCC")
8  canvas.pack()
9
10 win.mainloop()
```



# GUI元件 – Canvas(畫布)

參數	說明
create_text	繪製文字
create_line	繪製直線
create_rectangle()	繪製矩形
create_oval()	繪製圓形、橢圓形
create_arc()	繪製圓弧
create_polygon()	繪製多邊形
create_image()	繪製圖像



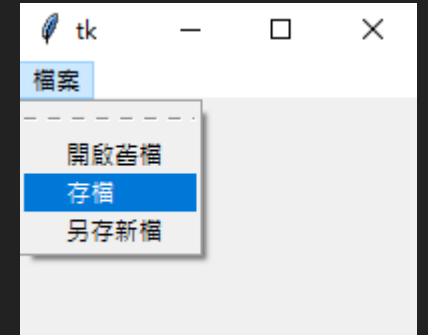
```
1  from tkinter import *
2
3  win = Tk()
4
5  # 設置畫布大小400*300
6  canvas = Canvas(win, width=400, height=300)
7  canvas.pack()
8
9  def paint(event):
10     x1, y1 = (event.x -1),(event.y -1)
11     x2, y2 = (event.x +1),(event.y +1)
12     canvas.create_oval(x1,y1,x2,y2, outline="red")
13
14  canvas.bind("<B1-Motion>", paint)
15
16  win.mainloop()
```

# GUI元件 – Menu(功能表)

```
1 from tkinter import *
2
3 win = Tk()
4 win.geometry("200x200")
5
6 # 建立主選單
7 menu = Menu(win)
8 menu.add_command(label="檔案") # 建立主選單項目
9 menu.add_command(label="編輯") # 建立主選單項目
10 menu.add_command(label="工具") # 建立主選單項目
11
12 win.config(menu=menu) # 將主選單添加到視窗上
13
14 win.mainloop()
```

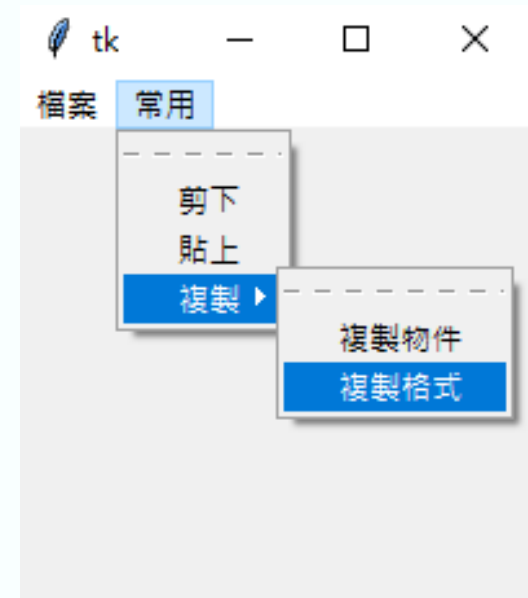


```
1 from tkinter import *
2
3 win = Tk()
4 win.geometry("200x200")
5
6 # 建立主選單
7 menu = Menu(win)
8
9 # 建立子選單
10 submenu = Menu(menu)
11 submenu.add_command(label="開啟舊檔") # 建立子選單項目
12 submenu.add_command(label="存檔") # 建立子選單項目
13 submenu.add_command(label="另存新檔") # 建立子選單項目
14
15 # 建立主選單「檔案」，內容為submenu
16 menu.add_cascade(label="檔案", menu=submenu)
17 win.config(menu=menu) # 將主選單添加到視窗上
18
19 win.mainloop()
```



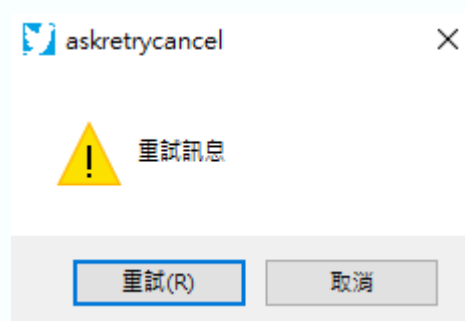
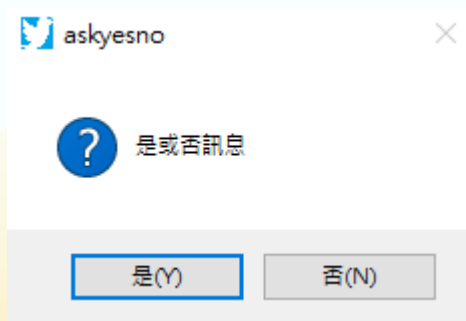
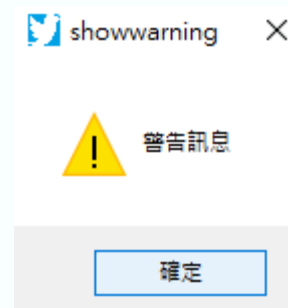
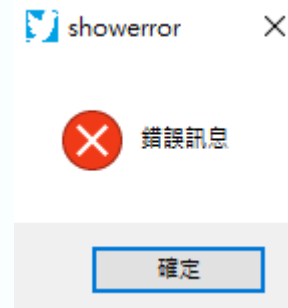
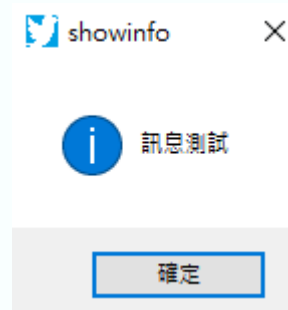
# GUI元件 – Menu(功能表)

```
1  from tkinter import *
2
3  win = Tk()
4  win.geometry("200x200")
5
6  menu = Menu(win)
7
8  # 建立第一個選單
9  menu1 = Menu(menu)
10 menu1.add_command(label="開啟舊檔")      # 建立第一個選單項目
11 menu1.add_command(label="存檔")          # 建立第一個選單項目
12 menu1.add_command(label="另存新檔")      # 建立第一個選單項目
13 menu.add_cascade(label="檔案", menu=menu1) # 將第一個選單項目綁定在「檔案」下
14
15 # 建立第二個選單
16 menu2 = Menu(menu)
17 menu2.add_command(label="剪下")           # 建立第二個選單項目
18 menu2.add_command(label="貼上")          # 建立第二個選單項目
19 menu.add_cascade(label="常用", menu=menu2) # 將第二個選單項目綁定在「常用」下
20
21 # 建立子選單的子選單
22 menu2_more = Menu(menu2)
23 menu2_more.add_command(label="複製物件")  # 建立第二個選單項目的子選單
24 menu2_more.add_command(label="複製格式")  # 建立第二個選單項目的子選單
25 menu2.add_cascade(label="複製", menu=menu2_more) # 將子選單綁定在「複製」下
26
27 win.config(menu=menu)    # 將主選單添加到視窗上
28
29 win.mainloop()
```



# GUI元件 – messagebox(對話方塊)

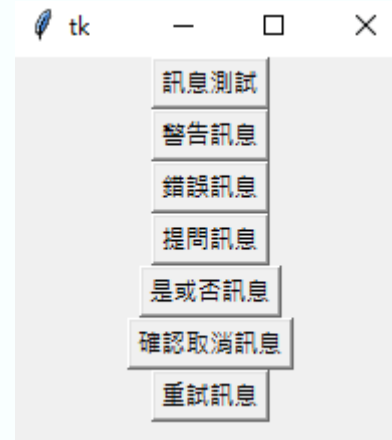
方法種類	說明
showinfo	普通訊息
showwarning	警告訊息
showerror	錯誤訊息
askquestion	提問(yes/no)
askyesno	是或否(yes/no)
askokcancel	確認或取消(ok/cancel)
askretrycancel	重試(retry/cancel)



# GUI元件 – messagebox(對話方塊)

```
1 from tkinter import *
2 from tkinter import messagebox
3
4 win = Tk()
5 win.geometry("200x200")
6
7 def showinfo():
8     messagebox.showinfo("showinfo", "訊息測試")
9
10 btn_showinfo = Button(win, text="訊息測試", command=showinfo)
11 btn_showinfo.pack()
12
13 def showwarning():
14     messagebox.showwarning("showwarning", "警告訊息")
15
16 btn_showwarning = Button(win, text="警告訊息", command=showwarning)
17 btn_showwarning.pack()
18
19 def showerror():
20     messagebox.showerror("showerror", "錯誤訊息")
21
22 btn_showerror = Button(win, text="錯誤訊息", command=showerror)
23 btn_showerror.pack()
24
25 def askquestion():
26     messagebox.askquestion("askquestion", "提問訊息")
27
28 btn_askquestion = Button(win, text="提問訊息", command=askquestion)
29 btn_askquestion.pack()
```

```
31 def askyesno():
32     messagebox.askyesno("askyesno", "是或否訊息")
33
34 btn_askyesno = Button(win, text="是或否訊息", command=askyesno)
35 btn_askyesno.pack()
36
37 def askokcancel():
38     messagebox.askokcancel("askokcancel", "確認取消訊息")
39
40 btn_askokcancel = Button(win, text="確認取消訊息", command=askokcancel)
41 btn_askokcancel.pack()
42
43 def askretrycancel():
44     messagebox.askretrycancel("askretrycancel", "重試訊息")
45
46 btn_askretrycancel = Button(win, text="重試訊息", command=askretrycancel)
47 btn_askretrycancel.pack()
48
49 win.mainloop()
```

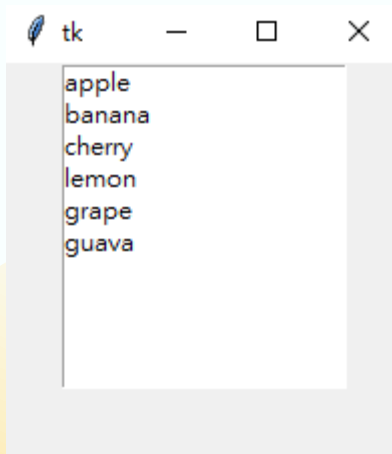


# GUI元件 – Listbox(列表)

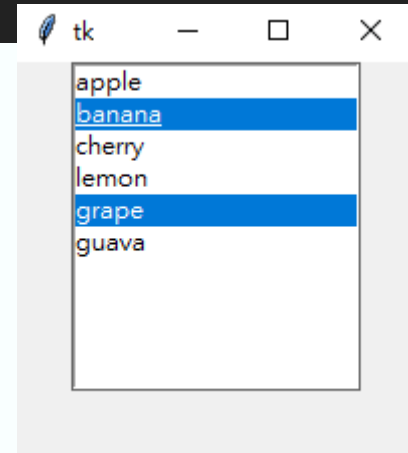
參數	說明
anchor	擺放位置，可以設定n、s、w、e，預設center
width	寬度，預設0
height	高度，預設0
bg	背景顏色
fg	文字顏色
font	字型設定，包含大小(px)、粗體(bold)、斜體(italic)
relief	邊框樣式，有flat、sunken、raised、ridge、solid，預設flat
bd	邊框粗細，預設1
listvariable	綁定清單變數，可動態存取listbox內容
selectmode	選取模式，可以設browse(預設，等同單選)、single(單選)、multiple(多選)、extended(等同多選，可以用滑鼠拖曳選取)

# GUI元件 – Listbox(列表)

```
1 from tkinter import *
2
3 win = Tk()
4 win.geometry("200x200")
5
6 # 設定文字變數
7 word = StringVar()
8 word.set(("apple", "banana", "cherry", "lemon", "grape", "guava"))
9
10 # 設定列表
11 listbox = Listbox(win, listvariable = word)
12 listbox.pack()
13
14 win.mainloop()
```



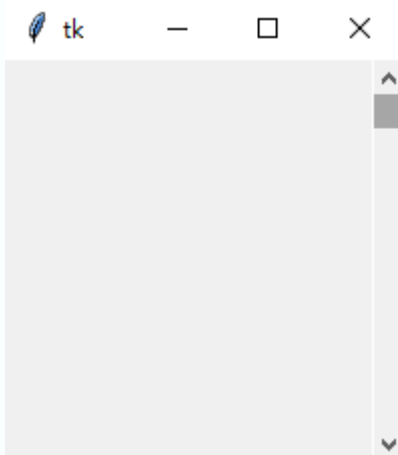
```
1 from tkinter import *
2
3 win = Tk()
4 win.geometry("200x200")
5
6 # 設定文字變數
7 word = StringVar()
8 word.set(("apple", "banana", "cherry", "lemon", "grape", "guava"))
9
10 # 單選模式
11 # listbox = Listbox(win, listvariable=word, selectmode="single")
12 # 多選模式
13 # listbox = Listbox(win, listvariable=word, selectmode="multiple")
14 # 多選模式(可用滑鼠拖曳)
15 listbox = Listbox(win, listvariable=word, selectmode="extended")
16 listbox.pack()
17
18 win.mainloop()
```



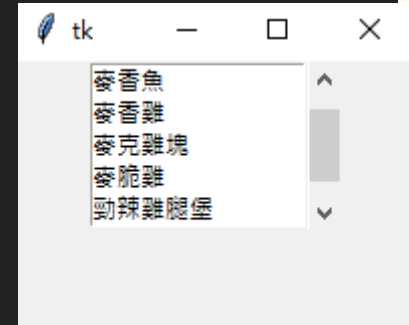


# GUI元件- Scrollbar(捲軸)

```
1 from tkinter import *
2
3 win = Tk()
4 win.geometry("200x200")
5
6 # 建立捲軸
7 scrollbar = Scrollbar(win)
8 # 將捲軸設定在視窗右邊，垂直填滿
9 scrollbar.pack(side="right", fill="y")
10
11 win.mainloop()
```



```
1 from tkinter import *
2
3 win = Tk()
4 win.geometry("200x200")
5
6 # 建立frame
7 frame = Frame(win, width=15)
8 frame.pack()
9
10 # 在frame裡放入捲軸
11 scrollbar = Scrollbar(frame)
12 # 將捲軸設定在frame右邊，垂直填滿
13 scrollbar.pack(side="right", fill="y")
14
15 # 設定列表清單
16 word = StringVar()
17 word.set(("大麥克", "麥香魚", "麥香雞", "麥克雞塊",
18          "麥脆雞", "勁辣雞腿堡", "牛肉吉事堡"))
19
20 # 列表設定
21 listbox = Listbox(frame, listvariable=word, height=5,
22                   width=15, yscrollcommand=scrollbar.set)
23 listbox.pack(side="left", fill="y")
24 scrollbar.config(command=listbox.yview)
25
26 win.mainloop()
```



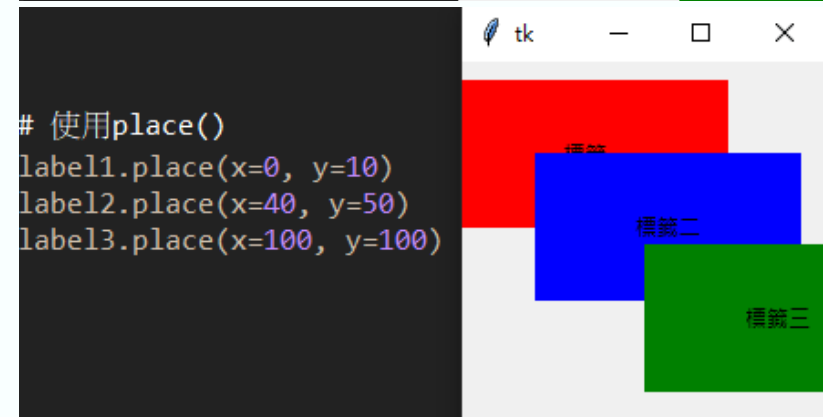
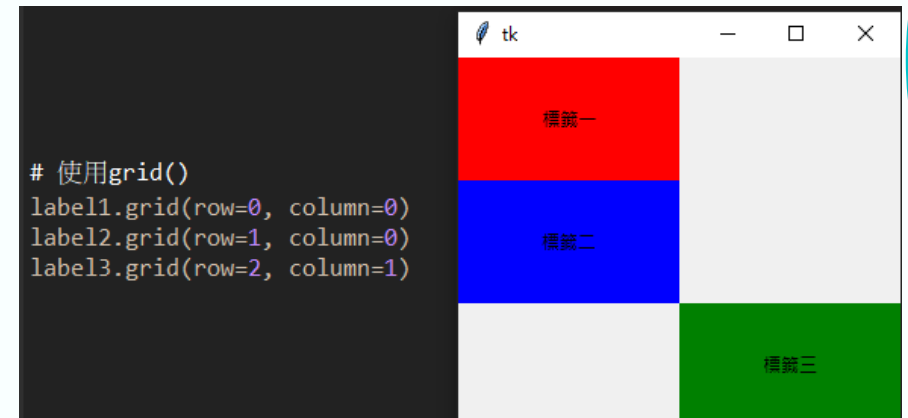
# GUI元件 – 布局

## ◆ 元件布局有三種方法

◆ `pack(side=XXXX)` : xxx可以是TOP、LEFT、BOTTOM、RIGHT

◆ `grid(row=x, column=y)` : 以行列布局

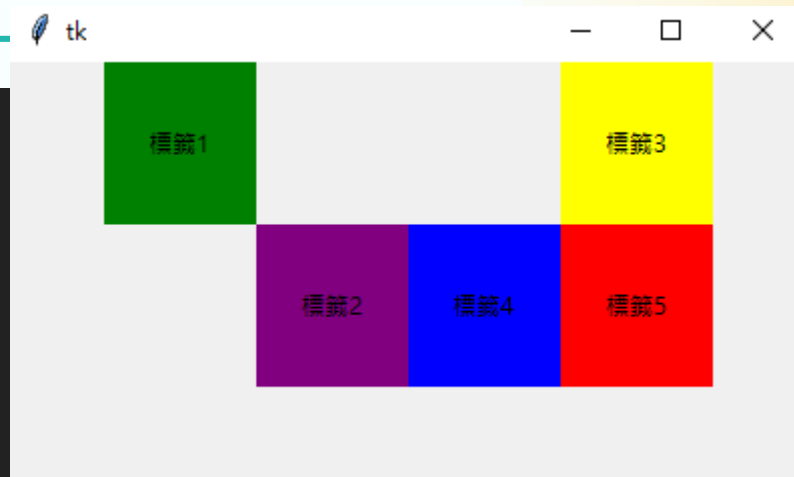
◆ `place(x=10,y=50)` : 以坐標方式布局



# GUI元件 – 布局管理元件

- ◆ Tkinter提供了幾種布局管理元件，主要作用為為其他元件提供容器，將主視窗劃分成多個區域，進而方便管理
- ◆ Frame：Frame可以放在主視窗內，Frame內還能再放Frame
- ◆ LabelFrame：為Frame的進階版，功能屬性跟Frame相似
- ◆ PanedWindow：在Tkinter8.4版後新增功能，和Frame不同之處在於PanedWindow可以讓使用者自訂每個區塊的大小

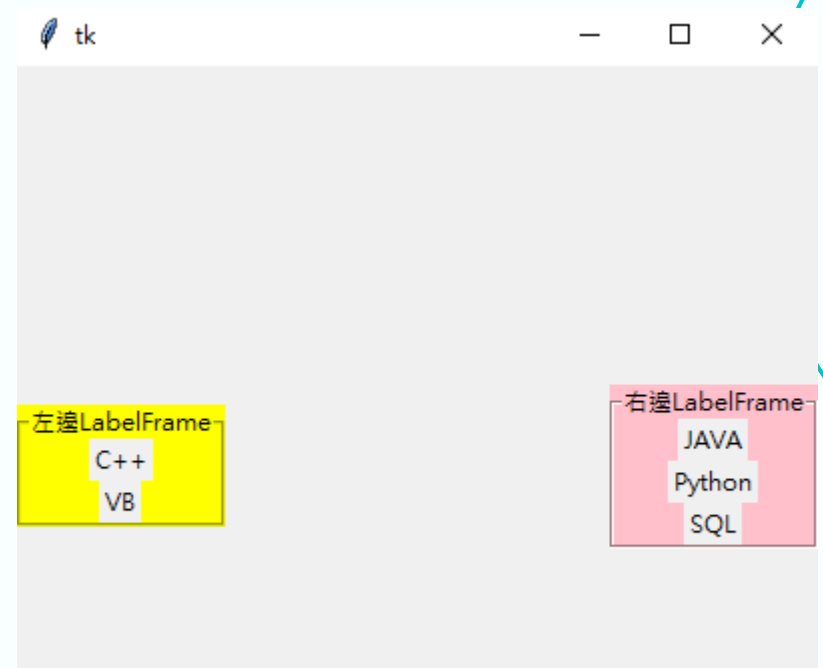
# 布局管理元件 – Frame



```
1  from tkinter import *
2
3  win = Tk()
4  # 視窗大小400*400，距離螢幕左邊及上邊各500
5  win.geometry("400x400+500+500")
6
7  # 在主視窗上添加Frame元件
8  frame = Frame(win)
9  frame.pack()
10
11 # 在Frame元件上再添加兩個Frame，分別放在左跟右
12 frame_left = Frame(frame)
13 Label(frame_left, text="標籤1", bg="green", width=10, height=5).grid(row=0, column=0)
14 Label(frame_left, text="標籤2", bg="purple", width=10, height=5).grid(row=2, column=2)
15 frame_left.pack(side=LEFT)
16
17 frame_right = Frame(frame)
18 Label(frame_right, text="標籤3", bg="yellow", width=10, height=5).grid(row=0, column=1)
19 Label(frame_right, text="標籤4", bg="blue", width=10, height=5).grid(row=1, column=0)
20 Label(frame_right, text="標籤5", bg="red", width=10, height=5).grid(row=1, column=1)
21 frame_right.pack(side=RIGHT)
22
23 win.mainloop()
```

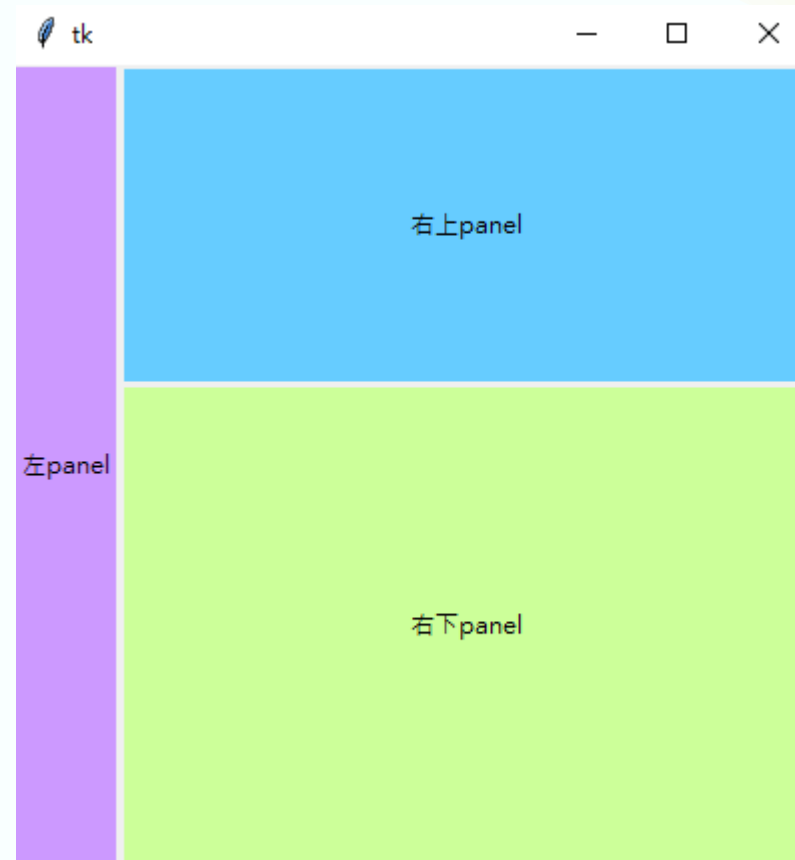
# 布局管理元件 – LabelFrame

```
1  from tkinter import *
2
3  win = Tk()
4  # 視窗大小400*400，距離螢幕左邊及上邊各500
5  win.geometry("400x400+500+500")
6
7  # 設定左邊LabelFrame，裡面有2個label
8  frame_left = LabelFrame(win, text="左邊LabelFrame",bg="yellow")
9  frame_left.pack(side=LEFT)
10 label_1 = Label(frame_left, text="C++")
11 label_1.pack()
12 label_2 = Label(frame_left, text="VB")
13 label_2.pack()
14
15 # 設定右邊LabelFrame，裡面有3個label
16 frame_right = LabelFrame(win, text="右邊LabelFrame",bg="pink")
17 frame_right.pack(side=RIGHT)
18 label_3 = Label(frame_right, text="JAVA")
19 label_3.pack()
20 label_4 = Label(frame_right, text="Python")
21 label_4.pack()
22 label_5 = Label(frame_right, text="SQL")
23 label_5.pack()
24
25 win.mainloop()
```



# 布局管理元件 – PanedWindow

```
1  from tkinter import *
2
3  win = Tk()
4  # 視窗大小400*400，距離螢幕左邊及上邊各500
5  win.geometry("400x400+500+500")
6
7  # 創建一個水平方向的panel
8  left_panel = PanedWindow(win)
9  left_panel.pack(fill=BOTH, expand=1)
10
11 # 在左邊panel裡添加一個label
12 left = Label(left_panel, text="左panel", bg="#CC99FF")
13 left_panel.add(left)
14
15 # 創建一個垂直方向的panel
16 right_panel = PanedWindow(orient=VERTICAL)
17 # 將垂直方向panel添加至水平方向panel內
18 left_panel.add(right_panel)
19 # 設置垂直方向panel內的2個label
20 top = Label(right_panel, text="右上panel", bg="#66CCFF", height=10)
21 bottom = Label(right_panel, text="右下panel", bg="#CCFF99")
22 right_panel.add(top)
23 right_panel.add(bottom)
24
25 win.mainloop()
```



# Tkinter事件處理

- ◆ 我們希望在按下滑鼠按鈕或是拖曳滑鼠時會觸發事件，此時就需要使用bind方法

```
1  from tkinter import *
2
3  win = Tk()
4  win.geometry("200x200")
5
6  def mouseClick(event):
7      print("滑鼠點擊")
8
9  label = Label(win, text="請按我", font=30)
10 label.pack()
11 label.bind("<Button>", mouseClick)
12
13 win.mainloop()
```

```
滑鼠點擊
滑鼠點擊
滑鼠點擊
滑鼠點擊
滑鼠點擊
```

# Tkinter事件處理

- ◆ bind("<Button>",函式名)的Button是指使用者點擊滑鼠時所觸發的事件
  - ◆ Button-1：滑鼠左鍵
  - ◆ Button-2：滑鼠中鍵
  - ◆ Button-3：滑鼠右鍵
  - ◆ Button-4：滾輪向上滾
  - ◆ Button-5：滾輪向下滾

```
1  from tkinter import *
2
3  win = Tk()
4  win.geometry("200x200")
5
6  def mouseCall(event):
7      print("滑鼠位置：%s,%s" %(event.x, event.y))
8
9  win.bind("<Button-1>", mouseCall)
10
11 win.mainloop()
```

滑鼠位置：57,26

滑鼠位置：116,98

滑鼠位置：62,121

滑鼠位置：42,152



# Tkinter事件處理

- ◆ `bind("<Motion>", 函式名)` 的 Motion 是指使用者移動滑鼠時所觸發的事件

```
1  from tkinter import *
2
3  win = Tk()
4  win.geometry("200x200")
5
6  ▼ def mouseMove(event):
7      print("滑鼠位置：%s,%s" %(event.x, event.y))
8
9  win.bind("<Motion>", mouseMove)
10
11 win.mainloop()
```

```
滑鼠位置：143,85
滑鼠位置：149,79
滑鼠位置：157,70
滑鼠位置：165,61
滑鼠位置：174,52
滑鼠位置：183,45
滑鼠位置：192,37
滑鼠位置：193,15
```