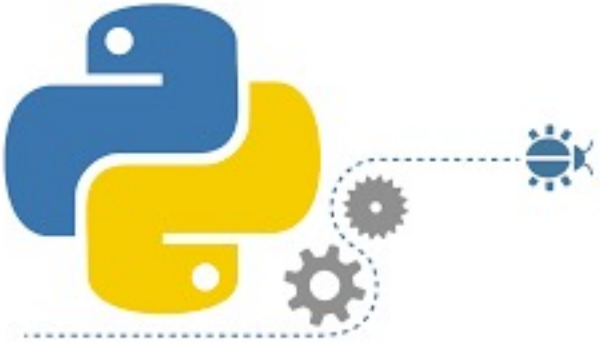
Python 的逆襲
- GUI 使用 tkinter

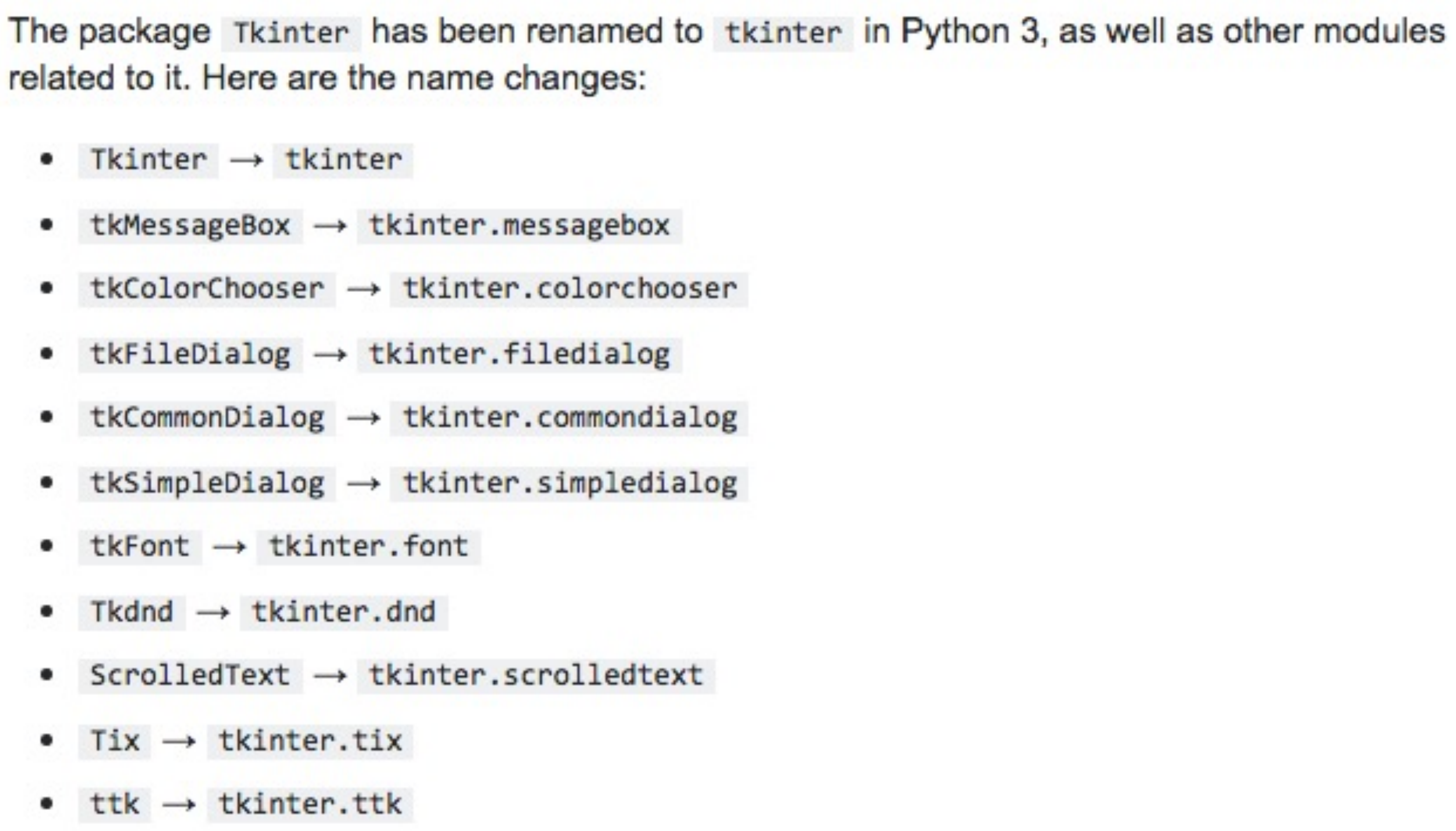


段維瀚 老師

GUI

- Tkinter 是 Python 的模組
  - 使用 tkinter 可以在 Python 中建立完整的 GUI 視窗程式
    - import tkinter (Python 3)
    - import Tkinter (Python 2.x)

Python 2.x 與 3 宣告對應

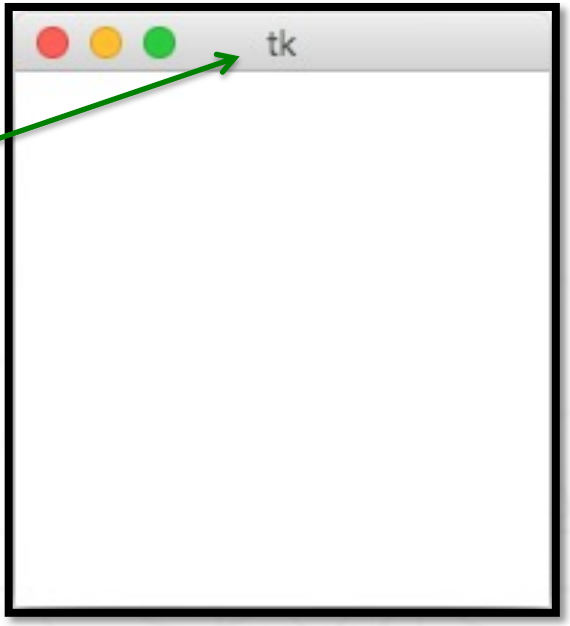


The package `Tkinter` has been renamed to `tkinter` in Python 3, as well as other modules related to it. Here are the name changes:

- `Tkinter` → `tkinter`
- `tkMessageBox` → `tkinter.messagebox`
- `tkColorChooser` → `tkinter.colorchooser`
- `tkFileDialog` → `tkinter.filedialog`
- `tkCommonDialog` → `tkinter.commondialog`
- `tkSimpleDialog` → `tkinter.simpledialog`
- `tkFont` → `tkinter.font`
- `Tkdnd` → `tkinter.dnd`
- `ScrolledText` → `tkinter.scrolledtext`
- `Tix` → `tkinter.tix`
- `ttk` → `tkinter.ttk`

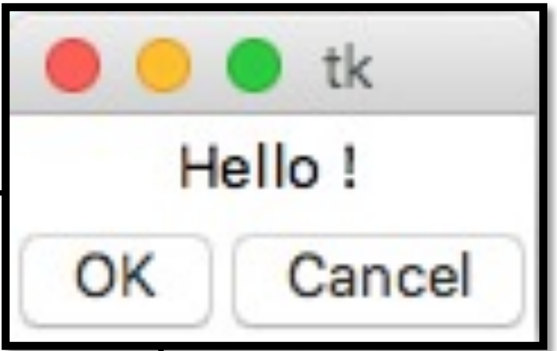
GUI-簡單的tkinter視窗

```
import tkinter
win = tkinter.Tk()

# 視窗抬頭
win.title("tk")

# 視窗大小
win.geometry("200x200")

win.mainloop()
```



GUI-加入元件

```
import tkinter

win = tkinter.Tk()

label = tkinter.Label(win, text="Hello !")
label.pack()

button1 = tkinter.Button(win, text="OK")
button1.pack(side=tkinter.LEFT)

button2 = tkinter.Button(win, text="Cancel")
button2.pack(side=tkinter.RIGHT)

win.mainloop()
```



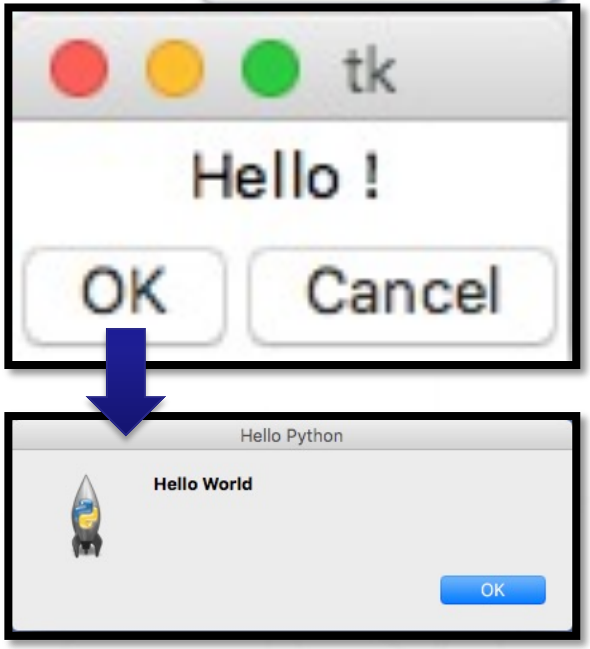
其他屬性

```
label = tkinter.Label(
     win,
     text=Hello ! ', # 標籤的文字
     bg='green', # 背景顏色
     font=('Arial', 12), # 字型和字體大小
     width=15, height=2 # 標籤長寬)
```

```
# 字型大小
label.config(font=("Courier", 44))
```

GUI-Command事件

```
import tkinter
from tkinter import messagebox

def helloCallBack():
   messagebox.showinfo( "Hello Python", "Hello World")

def quit():
   win.quit()

win = tkinter.Tk()

label = tkinter.Label(win, text="Hello !")
label.pack()

button1 = tkinter.Button(win, text="OK", command=helloCallBack)
button1.pack(side=tkinter.LEFT)

button2 = tkinter.Button(win, text="Cancel", command=quit)
button2.pack(side=tkinter.RIGHT)

win.mainloop()
```



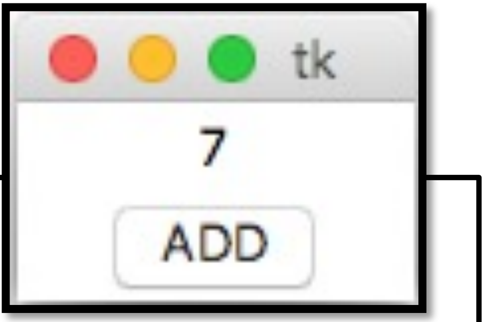
tk

Hello !

Hello Python

Hello World

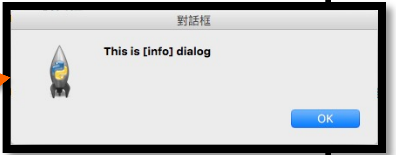
全域變數與字串參照



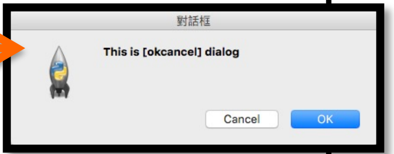
```
import tkinter

value = 0

def update():
    global value  # 把宣告在第 3 行的 value 視為全域變數
     value += 1
    var.set(str(value))

win = tkinter.Tk()
win.geometry("100x50")

var = tkinter.StringVar()   # 字串參照物件
label = tkinter.Label(win, textvariable=var)
var.set("0")
label.pack()

button1 = tkinter.Button(win, text="ADD", command=update)
button1.pack()

win.mainloop()
```

GUI-messagebox

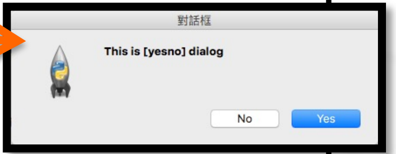
```
import tkinter
import tkinter.messagebox

def show_info():
   r = tkinter.messagebox.showinfo('對話框', 'This is [info] dialog')
   print(r) # 按下 Ok --> ok

def show_askokcancel():
   r = tkinter.messagebox.askokcancel('對話框', 'This is [okcancel] dialog')
   print(r) # 按下 ok --> True, cancel --> False

def show_askyesno():
   r = tkinter.messagebox.askyesno('對話框', 'This is [yesno] dialog')
   print(r) # 按下 yes --> True, no --> False

def show_askyesnocancel():
   r = tkinter.messagebox.askyesnocancel('對話框', 'This is [yesnocancel] dialog')
   print(r) # 按下 yes --> True, no --> False, cancel --> None

win = tkinter.Tk()

button1 = tkinter.Button(win, text="Click me", command=show_info)
button1.pack()

win.mainloop()
```


對話框

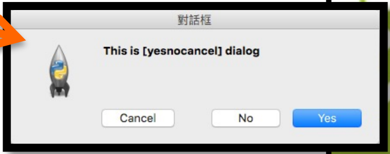
This is [info] dialog


對話框

This is [okcancel] dialog


對話框

This is [yesno] dialog


對話框

This is [yesnocancel] dialog

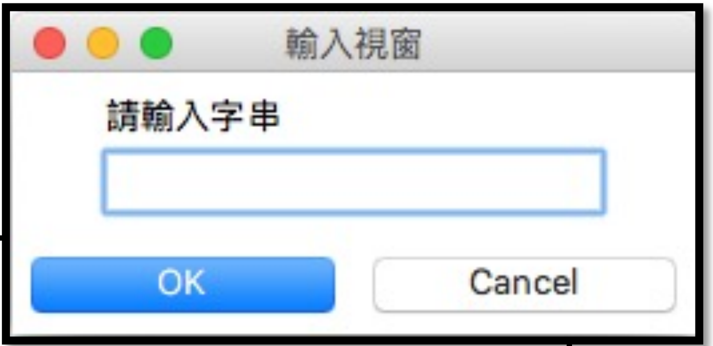
GUI-Entry

- Entry
  - 輸入文字框
    - entry = tkinter.Entry(root, justify=tkinter.CENTER)
      - justify=tkinter.CENTER #文字置中
      - entry.get() #取得文字內容
      - entry.delete(0, tkinter.END)
entry.insert(0, '文字訊息') #設定文字內容

GUI-Entry

```
import random
import tkinter
from tkinter import messagebox

def get():
    messagebox.showinfo("Hello Python", entry.get())

def set():
    entry.delete(0, tkinter.END)
    entry.insert(0, str(random.randint(1, 100)))

win = tkinter.Tk()

entry = tkinter.Entry(root, justify=tkinter.CENTER)
entry.insert(0, '0')
entry.pack()

button1 = tkinter.Button(root, text="Get", command=get)
button1.pack(side=tkinter.LEFT)

button2 = tkinter.Button(root, text="Set", command=set)
button2.pack(side=tkinter.RIGHT)

win.mainloop()
```

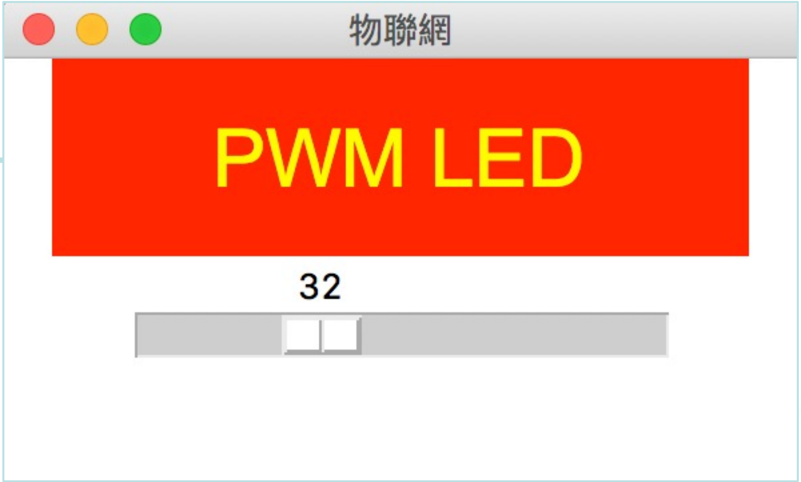
GUI-simpledialog



```
import tkinter
import tkinter.simpledialog

def setStr():
   r = tkinter.simpledialog.askstring('輸入視窗', '請輸入字串', initialvalue='')
   label.config(text=r)

def setInt():
   r = tkinter.simpledialog.askinteger('輸入視窗', '請輸入整數', initialvalue='')
   label.config(text=str(r))

def setFloat():
   r = tkinter.simpledialog.askfloat('輸入視窗', '請輸入浮點數', initialvalue='')
   label.config(text=str(r))

win = tkinter.Tk()

label = tkinter.Label(win, text="您輸入的是...")
label.pack()

button1 = tkinter.Button(win, text="輸入字串", command=setStr)
button1.pack(side=tkinter.LEFT)

button2 = tkinter.Button(win, text="輸入整數", command=setInt)
button2.pack(side=tkinter.LEFT)

button3 = tkinter.Button(win, text="輸入浮點數", command=setFloat)
button3.pack(side=tkinter.RIGHT)

win.mainloop()
```

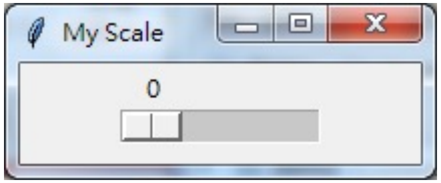
Scale

```
import tkinter

def getValue(args):
    value = scale.get() # 得到 float
    print(str(value/100))

win = tkinter.Tk()
win.title('物聯網')
win.geometry("300x200")

# add Label
label = tkinter.Label(win, text='PWM LED')
label.config(fg='yellow', bg='red', font=('Arial', 30), width=15, height=2)
label.pack()

scale = tkinter.Scale(
    win,
    width=15,
    length=200,
    from_=0, to=100,
    orient=tkinter.HORIZONTAL,
    showvalue=True,
    command=getValue)

scale.pack()
win.mainloop()
```



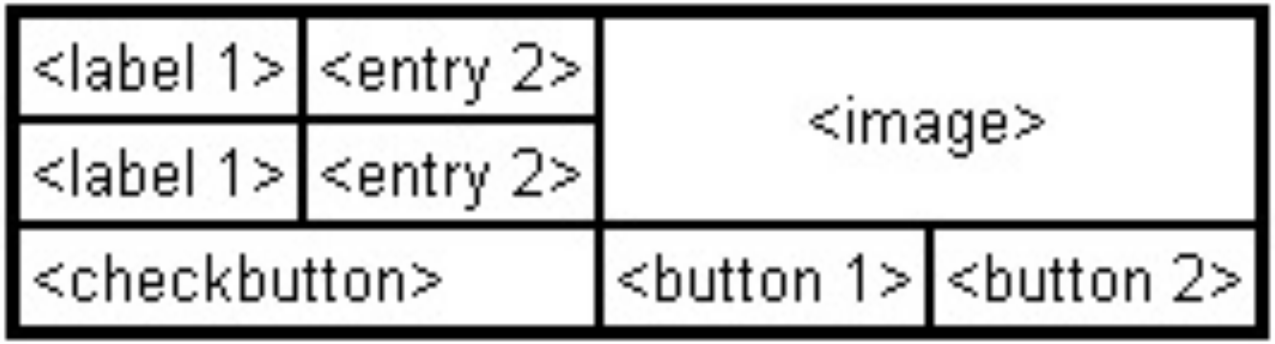
Scale

```
import tkinter

class MyScale:
    def getValue(self, args):
        value = args  # 得到 string
        #value = self.scale.get() # 得到 float
        print(value)

    def __init__(self) :
        self.value = 0
        self.root = tkinter.Tk()
        self.root.wm_title("My Scale")
        self.root.geometry("200x50")

        self.scale = tkinter.Scale(
            self.root,
            from_=0, to=255,
            orient=tkinter.HORIZONTAL,
            showvalue=True,
            command=self.getValue)

        self.scale.pack()
        self.root.mainloop()

scale = MyScale()
```

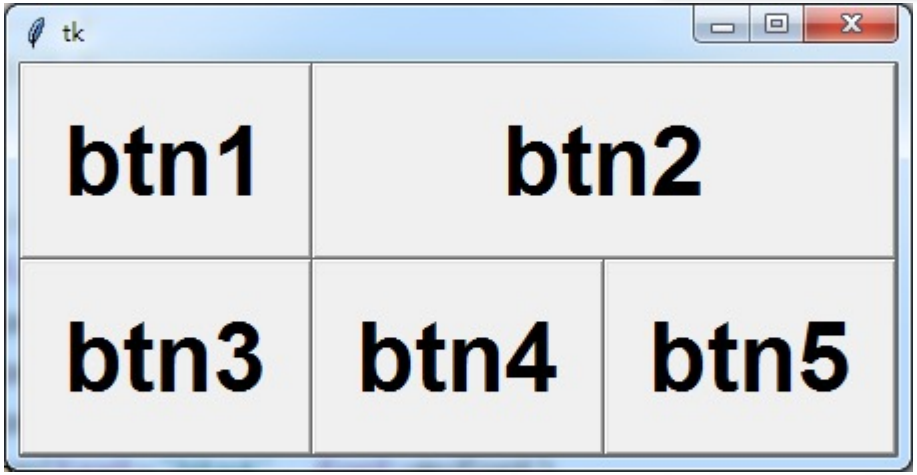


GUI-Grid 佈局管理器

- 範例
  - http://effbot.org/tkinterbook/grid.htm

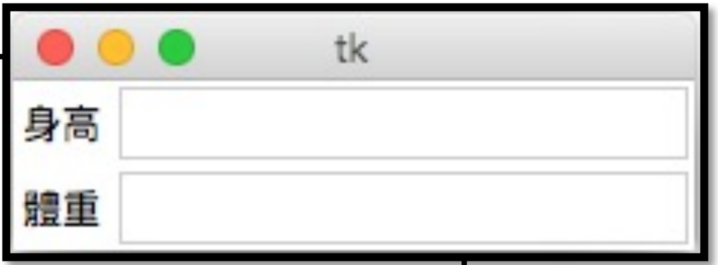


<label 1>	<entry 2>	<image>	
<label 1>	<entry 2>		
<checkbutton>		<button 1>	<button 2>



```
import tkinter
from tkinter import font
# using grid
# +------+-------------+
# | btn1 |    btn2     |
# +------+------+------+
# | btn3 | btn3 | btn4 |
# +-------------+------+
root = tkinter.Tk()
# tkFont.BOLD == 'bold'
myfont = font.Font(family='Helvetica', size=36, weight='bold')
btn1 = tkinter.Button(text='btn1', font=myfont)
btn2 = tkinter.Button(text='btn2', font=myfont)
btn3 = tkinter.Button(text='btn3', font=myfont)
btn4 = tkinter.Button(text='btn4', font=myfont)
btn5 = tkinter.Button(text='btn5', font=myfont)

root.rowconfigure((0,1), weight=1)  # 列 0, 列 1 同步放大縮小
root.columnconfigure((0,1,2), weight=1)  # 欄 0, 欄 1, 欄 2 同步放大縮小

btn1.grid(row=0, column=0, columnspan=1, sticky='EWNS') # sticky='EWNS' 無縫填滿
btn2.grid(row=0, column=1, columnspan=2, sticky='EWNS')
btn3.grid(row=1, column=0, columnspan=1, sticky='EWNS')
btn4.grid(row=1, column=1, columnspan=1, sticky='EWNS')
btn5.grid(row=1, column=2, columnspan=1, sticky='EWNS')

root.mainloop()
```

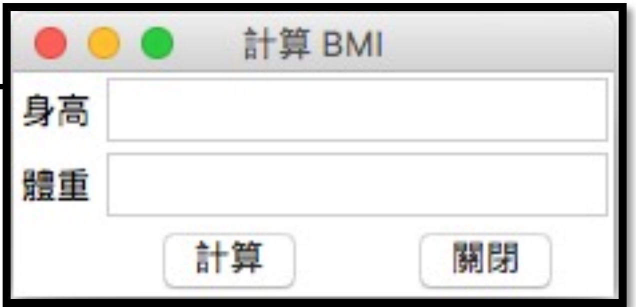
GUI-Grid 佈局管理器



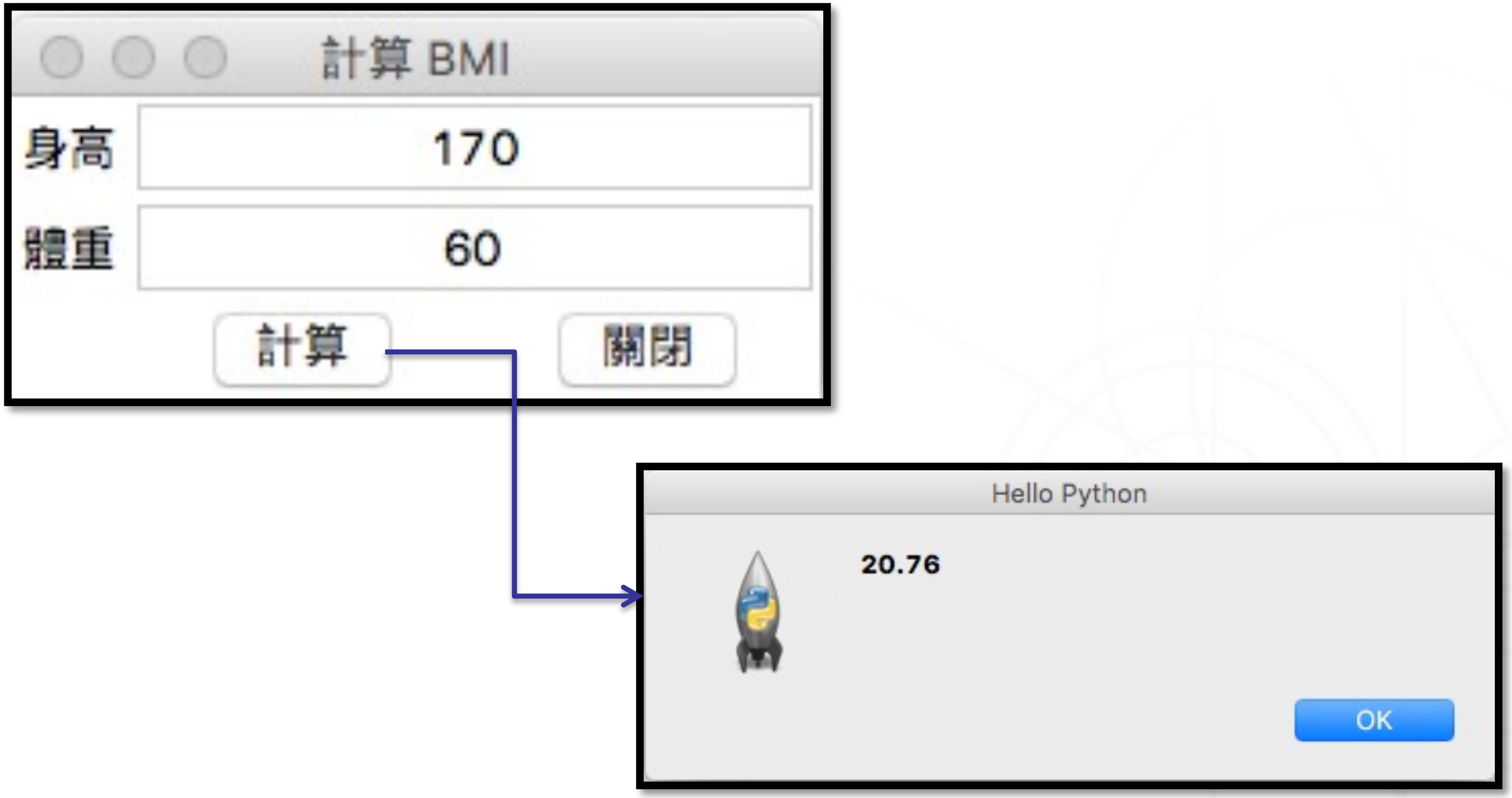
```
import tkinter

win = tkinter.Tk()

tkinter.Label(root, text="身高").grid(row=0)
tkinter.Label(root, text="體重").grid(row=1)

height = tkinter.Entry(win)
weight = tkinter.Entry(win)

height.grid(row=0, column=1)
weight.grid(row=1, column=1)

win.mainloop()
```

GUI-Grid 佈局管理器

```
import tkinter

win = tkinter.Tk()
win.title("計算 BMI")

tkinter.Label(win, text="身高").grid(row=0)
tkinter.Label(win, text="體重").grid(row=1)

height = tkinter.Entry(win, justify=tkinter.CENTER)
weight = tkinter.Entry(win, justify=tkinter.CENTER)

height.grid(row=0, column=1, columnspan=2)
weight.grid(row=1, column=1, columnspan=2)

button1 = tkinter.Button(win, text="計算")
button1.grid(row=2, column=1)

button2 = tkinter.Button(win, text="關閉")
button2.grid(row=2, column=2)

win.mainloop()
```



GUI-Lab



計算 BMI

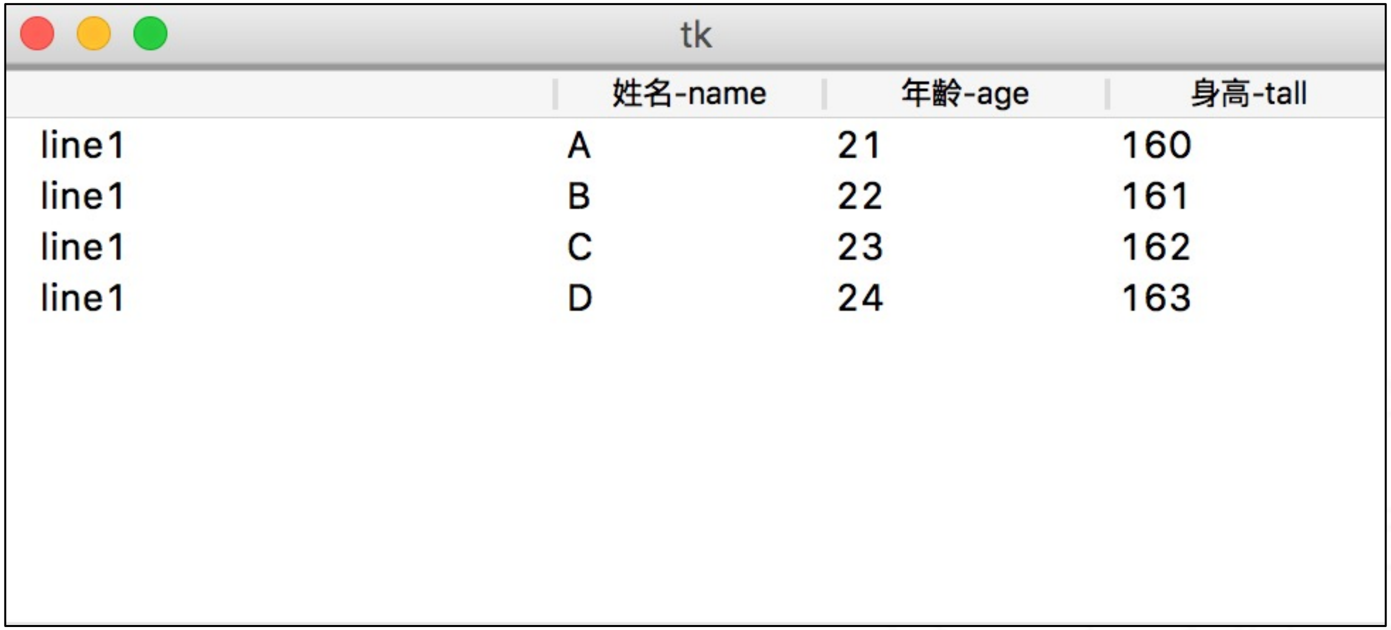
身高 170

體重 60

Hello Python

20.76

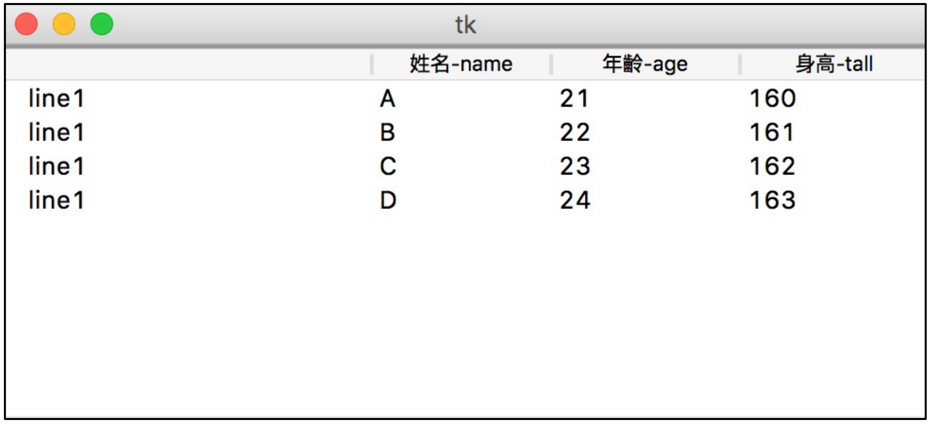
TreeView



	姓名-name	年齡-age	身高-tall
line1	A	21	160
line1	B	22	161
line1	C	23	162
line1	D	24	163

TreeView

```
import tkinter
from  tkinter import ttk  # 匯入資源

win = tkinter.Tk()
tree = ttk.Treeview(win) #表格

tree["columns"]=("姓名", "年齡", "身高")
tree.column("姓名", width=100)   # 欄位
tree.column("年齡", width=100)
tree.column("身高", width=100)

tree.heading("姓名", text="姓名-name")  # Title
tree.heading("年齡", text="年齡-age")
tree.heading("身高", text="身高-tall")

# insert(parent, index, iid=None, **kw)
# parent is the item ID of the parent item, or the empty string to create a new top-level item
tree.insert("", 0, text="line1" ,values=("A","21","160")) #插入資料，
tree.insert("", 1, text="line1" ,values=("B","22","161"))
tree.insert("", 2, text="line1" ,values=("C","23","162"))
tree.insert("", 3, text="line1" ,values=("D","24","163"))
tree.pack()
win.mainloop()
```



tk

	姓名-name	年齡-age	身高-tall
line1	A	21	160
line1	B	22	161
line1	C	23	162
line1	D	24	163

TreeView

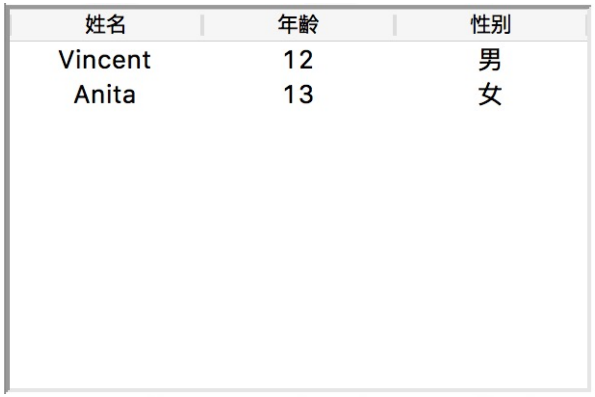
- 常用事件

事件	代碼
左鍵按下	ButtonPress-1
左鍵放開	ButtonRelease-1
左鍵雙擊	Double-Button-1
鍵盤事件	key
獲得鍵盤焦點	FocusIn
失去鍵盤焦點	FocusOut
滑鼠移動	B1-Motion
滑鼠移動到區域	Enter
滑鼠離開區域	Leave

TreeView

```
def click(event):  # Click
    item = tree.selection()[0]
    item_text = tree.item(item, "values")
    print(item_text[0]) # 第一列的資料

def keydown(event):  # Click
    print(event)
    item = tree.selection()[0]
    if event.char == 'd':  # 刪除
        tree.delete(item)
    elif event.char == 'e':  # 修改
        tree.item(item, text="line1", values=("z", "30", "170"))
    elif event.char == 'c':  # 清除
        for item in tree.get_children():
            tree.delete(item)


# ButtonPress-1 、Double-Button-1
tree.bind('<ButtonRelease-1>', click)  # 綁定事件
tree.bind('<Key>', keydown) # 綁定事件
```

TreeView-簡易版



姓名	年齡	性別
Vincent	12	男
Anita	13	女

```
data1 = ['Vincent','12','男']
data2 = ['Anita','13','女']

tree = ttk.Treeview(win, columns=['1','2','3'], show='headings')
tree.column('1',width=100, anchor='center')
tree.column('2',width=100, anchor='center')
tree.column('3',width=100, anchor='center')
tree.heading('1', text='姓名')
tree.heading('2', text='年齡')
tree.heading('3', text='性別')
tree.insert('', 'end', values=data1)  # end 最後一筆，0 第一筆
tree.insert('', 'end', values=data2)  # end 最後一筆，0 第一筆
tree.grid()
```