

# 圖形化介面開發 - Tkinter

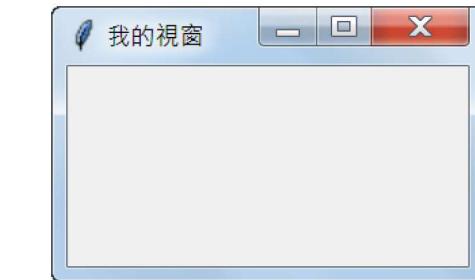
Python Implementation



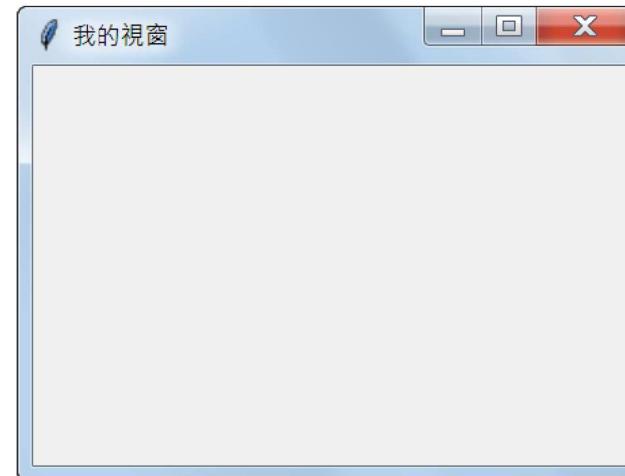
# Tkinter 套件

- ◆ Tkinter (唸做Tk-inter) 是一個跨平台的GUI套件，能夠在UNIX、Linux、Windows、Mac等平台開發GUI程式，下面是一個例子。

```
from tkinter import *\n\nwindow = Tk()\n\nwindow.title('我的視窗')\n\nwindow.geometry('200x100')\n\nwindow.maxsize(300, 200)\n\nwindow.mainloop()
```



一般 (geometry)



放大 (maxsize)

# GUI元件

---

◆ Tkinter提供了許多GUI元件 (widget)，例如：

- Frame (視窗區域)
- LabelFrame (標籤式視窗區域)
- Label (文字標籤)
- Entry (文字方塊)
- Text (文字區域)
- Button (按鈕)
- Checkbutton (核取按鈕)
- Radiobutton (選項按鈕)
- Listbox (清單方塊)
- Menu (功能表)
- Menubutton (功能表按鈕)
- Scrollbar (捲軸)
- Scale (滑桿)
- Spinbox (調整鈕)
- messagebox (對話方塊)
- PhotoImage (圖形)

# GUI元件

---

- ◆ Tkinter提供的元件都有各自的類別，如欲在視窗上面放置元件，只要根據元件類別建立物件即可，其語法如下：

元件類別(父物件, 選擇性參數1 = 值1, ...)

- ◆ 例如：

```
btn1 = Button(window, text = '確定', bg = 'yellow')
```

```
btn1['fg'] = 'red'
```

```
btn1['bg'] = 'blue'
```

```
btn1.config(fg = "red", bg = "blue")
```

```
btn1.pack()
```

# GUI元件

## ◆ Label (文字標籤)

Label (文字標籤) 可以用來顯示無法由使用者編輯的文字，我們可以使用如下語法建立文字標籤：

Label(父物件, 選擇性參數1 = 值1, ...)

## ◆ 例如：

```
01  from tkinter import *
02  window = Tk()
03  label1 = Label(window, text = "文字標籤1", width = 30, bg = "lightyellow")
04  label2 = Label(window, text = "文字標籤2", width = 30, bg = "lightblue")
05  label3 = Label(window, text = "文字標籤3", width = 30, bg = "lightgray")
06  label1.pack()
07  label2.pack()
08  label3.pack()
09  window.mainloop()
```



# GUI元件

---

◆ 常用的選擇性參數如下：

- **text**：文字
- **width**：寬度
- **height**：高度
- **bg**或**background**：背景色彩
- **fg**或**foreground**：前景色彩
- **bd**或**borderwidth**：框線寬度
- **padx**：水平間距，預設值為1
- **pady**：垂直間距，預設值為1
- **justify**：對齊方式，有**left**、**right**、**center**三種

# GUI元件

## ◆ 設定位置的第一種方法 - pack()

將前一個例子第06 ~ 08行改寫成如下：

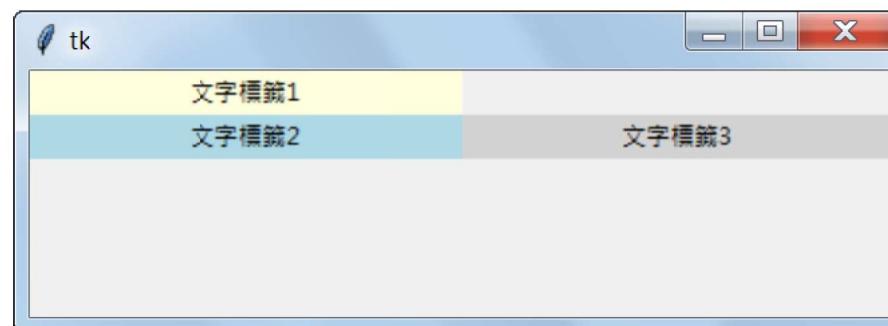
```
06     label1.pack(side = BOTTOM)  
07     label2.pack(side = BOTTOM)  
08     label3.pack(side = BOTTOM)
```



## ◆ 設定位置的第二種方法 - grid()

將前一個例子第06 ~ 08行改寫成如下：

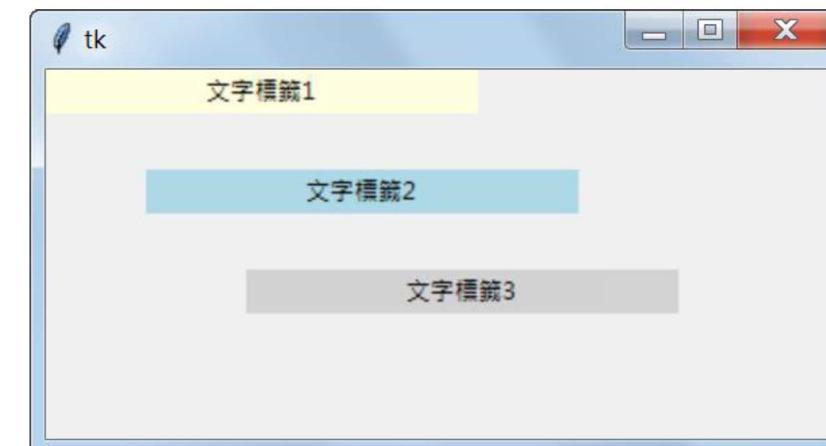
```
06     label1.grid(row = 0, column = 0)  
07     label2.grid(row = 1, column = 0)  
08     label3.grid(row = 1, column = 1)
```



## ◆ 設定位置的第三種方法 - place()

將前一個例子第06 ~ 08行改寫成如下：

```
06     label1.place(x = 0, y = 0)  
07     label2.place(x = 50, y = 50)  
08     label3.place(x = 100, y = 100)
```



# GUI元件

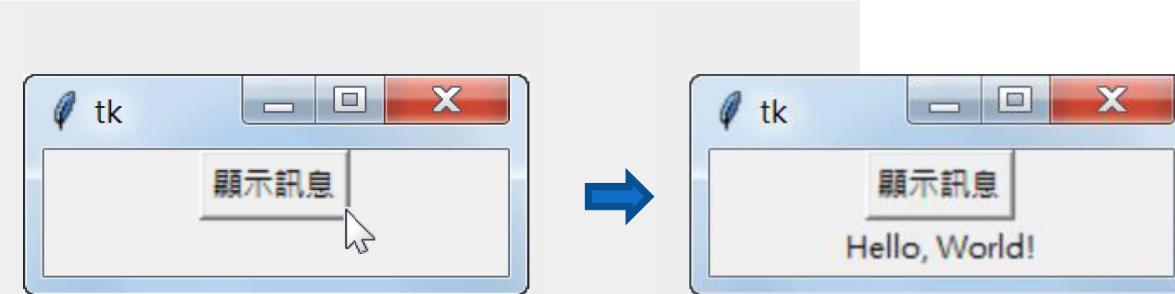
## ◆ Button (按鈕)

Button (按鈕) 可以用來執行、終止或中斷動作，我們可以使用如下語法建立按鈕：

**Button(父物件, 選擇性參數1 = 值1, ...)**

例如：

```
from tkinter import *
def showMsg():
    label1['text'] = 'Hello, World!'
    window = Tk()
    btn1 = Button(window, text = "顯示訊息", command = showMsg)
    label1 = Label(window)
    btn1.pack()
    label1.pack()
    window.mainloop()
```



# GUI元件

---

## ◆ 常用的選擇性參數如下：

- **text**：文字
- **width**：寬度
- **height**：高度
- **bg**或**background**：背景色彩
- **fg**或**foreground**：前景色彩
- **bd**或**borderwidth**：框線寬度
- **padx**：水平間距，預設值為1
- **pady**：垂直間距，預設值為1
- **justify**：對齊方式，有LEFT、RIGHT、CENTER三種。
- **Image**：按鈕上面的圖形
- **textvariable**：文字變數，用來取得或設定按鈕的文字
- **underline**：加底線的字元，預設值為-1，表示全部不加底線，0表示第一個字元，1表示第二個字元，依此類推
- **Command**：當使用者按一下按鈕時，會叫此參數所指定的函式

# GUI元件

## ◆ Entry (文字方塊)

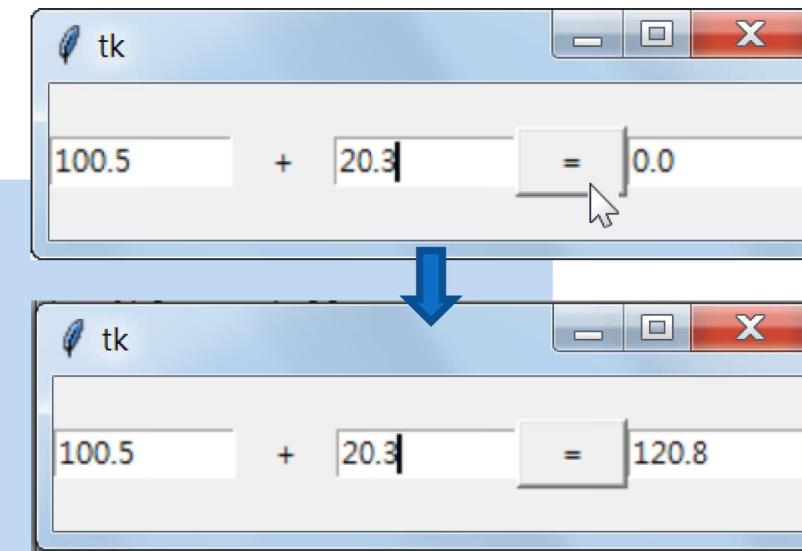
Entry (文字方塊) 可以用來取得使用者輸入的資料，我們可以使用如下語法建立文字方塊：

Entry(父物件, 選擇性參數1 = 值1, ...)

例如：

```
from tkinter import *
def add():
    result.set(num1.get() + num2.get())

    window = Tk()
    num1 = DoubleVar()
    num2 = DoubleVar()
    result = DoubleVar()
    Entry(window, width = 10, textvariable = num1).pack(side = LEFT)
    Label(window, width = 5, text = "+").pack(side = LEFT)
    Entry(window, width = 10, textvariable = num2).pack(side = LEFT)
    Button(window, width = 5, text = "=", command = add).pack(side = LEFT)
    Entry(window, width = 10, textvariable = result).pack(side = LEFT)
    window.mainloop()
```



# GUI元件

---

◆ 常用的選擇性參數如下：

- `width`：寬度
- `bg`或`background`：背景色彩
- `fg`或`foreground`：前景色彩
- `state`：輸入狀態，預設值為`NORMAL`，而`DISABLED`表示無法輸入。
- `show`：顯示的字元，例如`show='*'`會顯示星號。
- `textvariable`：文字變數，用來取得或設定文字方塊的資料。

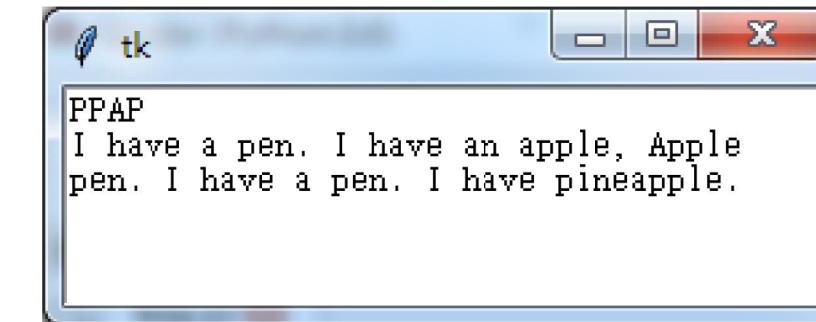
# GUI元件

## ◆ (文字區域)

Text (文字區域) 可以用來顯示與編輯具有格式的文字，我們可以使用如下語法建立文字區域：

Text(父物件, 選擇性參數1 = 值1, ...)

例如：



```
import tkinter

song = 'I have a pen. I have an apple, Apple pen. I have a pen. I have pineapple.'

window = Tk()
text1 = Text(window, width = 40, height = 6, wrap = WORD)
text1.insert(END, 'PPAP\n')
text1.insert(END, song)
text1.pack()
window.mainloop()
```

# GUI元件

---

## ◆ 常用的選擇性參數如下：

- **width**：寬度
- **height**：高度
- **bg**或**background**：背景色彩
- **fg**或**foreground**：前景色彩
- **bd**或**borderwidth**：框線寬度
- **padx**：水平間距，預設值為1
- **pady**：垂直間距，預設值為1
- **state**：輸入狀態，預設值為NORMAL，而DISABLED表示無法輸入。
- **wrap**：換行，預設值為CHAR，表示當一行的長度超過文字區域的寬度時，會將單字換行，而WORD不會切斷單字，將整個單字換行，NONE則是不換行。
- **Xscrollcommand**：水平捲軸
- **Yscrollcommand**：垂直捲軸

# GUI元件

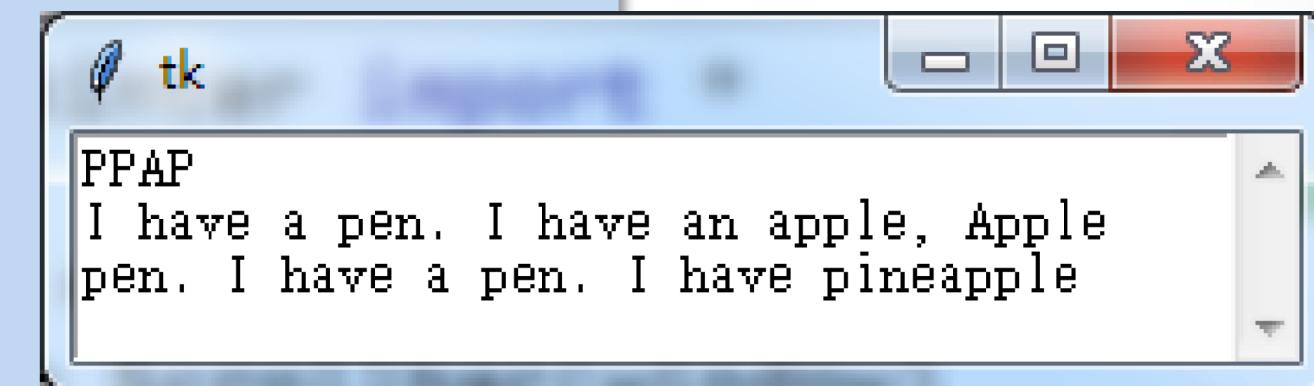
## ◆ Scrollbar (捲軸)

Scrollbar (捲軸) 可以用來在Text (文字區域)、Listbox (清單方塊) 顯示捲軸，我們可以使用如下語法建立捲軸：

**Scrollbar(父物件, 選擇性參數1 = 值1, ...)**

例如：

```
from tkinter import *
song = 'I have a pen. I have an apple, Apple pen. I have a pen. I have pineapple'
window = Tk()
sbar1 = Scrollbar(window)
text1 = Text(window, width = 40, height = 4, wrap = WORD)
text1.insert(END, "PPAP\n")
text1.insert(END, song)
sbar1.pack(side = RIGHT, fill = Y)
text1.pack(side = LEFT, fill = Y)
sbar1['command'] = text1.yview
text1['yscrollcommand'] = sbar1.set
window.mainloop()
```



# GUI元件

---

◆ 常用的選擇性參數如下：

- width : 寬度
- bg或background : 背景色彩
- bd或borderwidth : 框線寬度
- highlightbackground : 反白背景色彩
- highlightcolor : 反白色彩
- activebackground : 當移動到捲軸時，捲軸與箭頭的色彩
- orient : 預設值為VERTICAL，表示垂直捲軸；HORIZONTAL表示水平捲軸
- command : 當移動捲軸時，會呼叫此參數所指定的函式

# GUI元件

## ◆ messagebox (對話方塊)

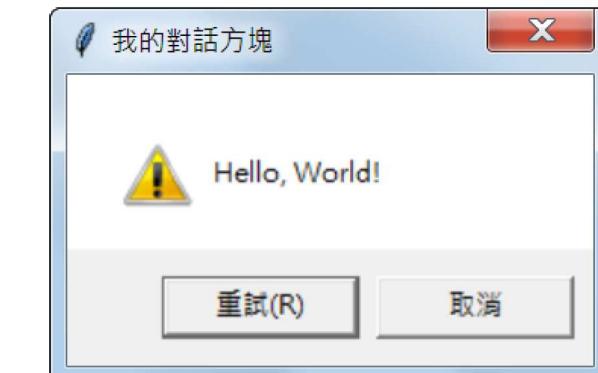
tkinter套件的messagebox模組提供下列數個方法可以用來顯示對話方塊：

- askokcancel(*title*, *message*, *options*)
- askquestion(*title*, *message*, *options*)
- askretrycancel(*title*, *message*, *options*)
- askyesno(*title*, *message*, *options*)
- showerror(*title*, *message*, *options*)
- showinfo(*title*, *message*, *options*)
- showwarning(*title*, *message*, *options*)

例如：

```
messagebox.askokcancel('我的對話方塊', 'Hello, World!')
```

```
messagebox.askretrycancel('我的對話方塊', 'Hello, World!')
```



# GUI元件

## ◆ Checkbutton (核取按鈕)

Checkbutton (核取按鈕) 允許使用者核取多個選項，我們可以使用如下語法建立核取按鈕：

**Checkbutton (父物件, 選擇性參數1 = 值1, ...)**

## ◆ 常用的選擇性參數如下：

- **text** : 文字
- **width** : 寬度
- **height** : 高度
- **bg或background** : 背景色彩
- **textvariable** : 取得或設定核取按鈕的文字
- **variable** : 取得或設定核取按鈕的目前狀態
- **command** : 當核取按鈕的狀態改變時，會呼叫此參數所指定的函式



# GUI元件

```
from tkinter import *

def showMsg():

    result = ""

    for i in checkvalue:

        if checkvalue[i].get() == True:

            result = result + dessert[i] + '\t'

    messagebox.showinfo("核取結果", result)

window = Tk()

label1 = Label(window, text = "請核取您喜歡的甜點 : ").pack()

dessert = {0 : "馬卡龍", 1 : "舒芙蕾", 2 : "草莓塔", 3 : "蘋果派"}

checkvalue = {}

for i in range(len(dessert)):

    checkvalue[i] = BooleanVar()

    Checkbutton(window, variable = checkvalue[i], text = dessert[i]).pack()

Button(window, text = "確定", command = showMsg).pack()

window.mainloop()
```



# GUI元件

## ◆ Radiobutton (選項按鈕)

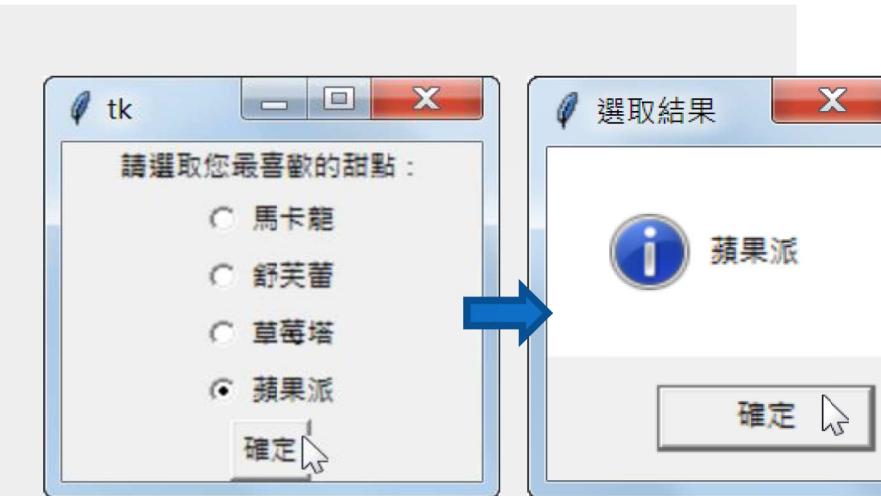
Radiobutton (選項按鈕) 允許使用者選取一個選項，我們可以使用如下語法建立選項按鈕：

**Radiobutton (父物件, 選擇性參數1 = 值1, ...)**

```
from tkinter import *

def showMsg():
    i = radiovalue.get()
    messagebox.showinfo("選取結果", dessert[i])

window = Tk()
label1 = Label(window, text = "請選取您最喜歡的甜點 :").pack()
dessert = {0 : "馬卡龍", 1 : "舒芙蕾", 2 : "草莓塔", 3 : "蘋果派"}
radiovalue = IntVar()
radiovalue.set(0)
for i in range(len(dessert)):
    Radiobutton(window, text = dessert[i], variable = radiovalue, value = i).pack()
Button(window, text = "確定", command = showMsg).pack()
window.mainloop()
```



# GUI元件

---

選擇性參數與方法	設定說明
<b>text</b>	文字
<b>width</b>	寬度
<b>height</b>	高度
<b>textvariable</b>	取得或設定核取按鈕的文字
<b>value</b>	選項按鈕的值，用來區分不同的選項按鈕
<b>variable</b>	取得或設定核取按鈕的目前狀態
<b>command</b>	當核取按鈕的狀態改變時，會呼叫此參數所指定的函式

# GUI元件

## ◆ Menu (功能表)

我們可以使用如下語法建立Menu (功能表) :

Menu (父物件, 選擇性參數1 = 值1, ...)



```
from tkinter import *
def newFile():
    messagebox.showinfo("開啟檔案", "在此撰寫開啟檔案的敘述")
def openFile():
    messagebox.showinfo("開啟舊檔", "在此撰寫開啟舊檔的敘述")
def about():
    messagebox.showinfo("關於我們", "在此撰寫關於我們的敘述")
window = Tk()
menu = Menu(window)
window["menu"] = menu
filemenu = Menu(menu, tearoff=0)
menu.add_cascade(label = "檔案", menu = filemenu)
filemenu.add_command(label = "開啟檔案...", command = newFile)
filemenu.add_command(label="開啟舊檔...", command = openFile)
filemenu.add_separator()
filemenu.add_command(label="離開", command = window.destroy)
helpmenu = Menu(menu)
menu.add_cascade(label = "說明", menu = helpmenu)
helpmenu.add_command(label="關於我們...", command = about)
window.mainloop()
```

# GUI元件

選擇性參數與方法	設定說明
<code>bg / background</code>	背景色彩
<code>fg / foreground</code>	前景色彩
<code>bd / borderwidth</code>	框線寬度
<code>activebackground</code>	當指標移到項目上面時的反白色彩
<code>tearoff</code>	第一個項目上面的分隔線， 若不要顯示，可以加上 <code>tearoff=0</code>
<code>add_cascade</code>	加入子功能表，參數 <code>options</code> 為選擇性參數， 例如 <code>label</code> 用來指定子功能表的文字， <code>menu</code> 用來指定子功能表與哪個 <code>Menu</code> 元件產生關聯
<code>add_command</code>	加入項目，參數 <code>options</code> 為選擇性參數， 例如 <code>label</code> 用來指定子功能表的文字， <code>command</code> 用來指定當按一下項目時所要呼叫的函式
<code>add_separator()</code>	加入分隔線

# GUI元件

## ◆ PhotoImage (圖形)

我們可以使用PhotoImage (圖形) 類別在視窗加入圖形，其語法如下：

**PhotoImage(file = "GIF或PGM/PPM圖檔"))**

```
from tkinter import *
def showMsg():
    i = radiovalue.get()
    if i == 0:
        messagebox.showinfo("選取結果", "繡球花")
    else:
        messagebox.showinfo("選取結果", "鬱金香")
window = Tk()
image1 = PhotoImage(file = "flower1.gif")
image2 = PhotoImage(file = "flower2.gif")
label1 = Label(window, text = "請選取您最喜歡的花 :").pack()
radiovalue = IntVar()
radiovalue.set(0)
Radiobutton(window, image = image1, variable = radiovalue, value = 0).pack()
Radiobutton(window, image = image2, variable = radiovalue, value = 1).pack()
Button(window, text = "確定", command = showMsg).pack()
window.mainloop()
```



# Q & A