**import tkinter as tk**

#視窗相關

root.**title**('pythontk') # 設定標題

root.**iconbitmap**('favicon.ico') # 設定 icon ( 格式限定 .ico )

width = 600

height = 400

left = 0

top = 0

root.**geometry**(f'{width}x{height}+{left}+{top}') # 定義視窗的尺寸和位置 (注意格式)

root.**minsize**(200, 200) # 設定視窗最小為 200x200

root.**maxsize**(500, 500) # 設定視窗最大為 500x500

root.**resizable**(False, False) # 設定 x 方向和 y 方向都不能縮放

root.**configure**(background='#000') # 設定背景色黑色

**root = tk.Tk()** # 建立 tkinter 視窗物件

**root.mainloop()** # 放在主迴圈中

#按鈕相關

| **參數** | **說明** |
| --- | --- |
| anchor | 擺放位置，可以設定 n、s、w、e、ne、nw、sw、se、center  ( e 右，w 左，s 下，n 上 )，預設 center。 |
| text | 文字內容，可以使用換行符 ( \n )。 |
| width | 寬度，單位是字元數，預設 0。 |
| height | 高度，單位是字元數，預設 0。 |
| padx | 內容和標籤左右邊界的間距 ( px )，預設 1。 |
| pady | 內容和標籤上下邊界的間距 ( px )，預設 1。 |
| bg/background | 背景顏色，可以使用十六進位色碼或顏色名稱。 |
| fg/foreground | 文字顏色，可以使用十六進位色碼或顏色名稱。 |
| font | 字型設定，包含字體、大小 ( px )、粗體 ( bold )、斜體 ( italic )。 |
| justify | 多行文字的對齊方式，可以設定 left、right、center，預設 center。 |
| cursor | 滑鼠移動到標籤的樣式，可以設定 arrow、circle、cross、plus...等，預設 arrow。 |
| relief | 邊框樣式，可以設定 flat、sunken、raised、groove、ridge、solid，預設 flat。 |
| bd/borderwidth | 邊框粗細，預設 1。 |
| textvariable | 文字內容的變數名稱，如果變數被修改，文字就會發生變化。 |
| underline | 第幾個字元開始加底線 ( 0 為第一個字元, 預設 -1 不加底線 )。 |
| wraplength | 一段文字超過多少寬度 ( px ) 會換行。 |
| image | 圖片內容。 |
| bitmap | 使用 bitmap 圖示作為標籤內容。 |
| compound | 設定文字與圖片排列的方式，可以設定 none、center、top、bottom、left、right，預設 none。 |

| **參數** | **說明** |
| --- | --- |
| command | 按下按鈕要執行的函式。 |
| activeforeground | 按下按鈕時的按鈕文字顏色。 |
| state | 按鈕狀態，tk.NORMAL 可使用 ( 預設 )、tk.DISABLED 不可使用。 |

btn = tk.Button(root,

text='我是按鈕',

font=('Arial',30,'bold'),

padx=10,

pady=10,

**activeforeground='#f00'**

)

btn.**pack()** #加入視窗中

#標籤相關

| **參數** | **說明** |
| --- | --- |
| anchor | 擺放位置，可以設定 n、s、w、e、ne、nw、sw、se、center  ( e 右，w 左，s 下，n 上 )，預設 center。 |
| text | 文字內容，可以使用換行符 ( \n )。 |
| width | 寬度，單位是字元數，預設 0。 |
| height | 高度，單位是字元數，預設 0。 |
| padx | 內容和標籤左右邊界的間距 ( px )，預設 1。 |
| pady | 內容和標籤上下邊界的間距 ( px )，預設 1。 |
| bg/background | 背景顏色，可以使用十六進位色碼或顏色名稱。 |
| fg/foreground | 文字顏色，可以使用十六進位色碼或顏色名稱。 |
| font | 字型設定，包含字體、大小 ( px )、粗體 ( bold )、斜體 ( italic )。 |
| justify | 多行文字的對齊方式，可以設定 left、right、center，預設 center。 |
| cursor | 滑鼠移動到標籤的樣式，可以設定 arrow、circle、cross、plus...等，預設 arrow。 |
| relief | 邊框樣式，可以設定 flat、sunken、raised、groove、ridge、solid，預設 flat。 |
| bd/borderwidth | 邊框粗細，預設 1。 |
| textvariable | 文字內容的變數名稱，如果變數被修改，文字就會發生變化。 |
| underline | 第幾個字元開始加底線 ( 0 為第一個字元, 預設 -1 不加底線 )。 |
| wraplength | 一段文字超過多少寬度 ( px ) 會換行。 |
| image | 圖片內容。 |
| bitmap | 使用 bitmap 圖示作為標籤內容。 |
| compound | 設定文字與圖片排列的方式，可以設定 none、center、top、bottom、left、right，預設 none。 |

mylabel = tk.Label(root,

text='hello world\n大家好',

font=('Arial',20,'bold'),

fg='#fff',

bg='#f50',

relief='groove',

pady=20,

padx=20)

mylabel.pack()

作業練習: 加入 bitmap 圖示

#加入文字變數

a = tk.StringVar() # 建立文字變數

a.set('hello') # 設定內容

mylabel = tk.Label(root, textvariable=a, font=('Arial',20)) # 內容是 a 變數

mylabe.pack()

#混合練習

n = 0 # n 等於 0

a = tk.StringVar() # 設定 a 為文字變數

a.set(n) # 設定 a 的內容

**def add():**

**global n # n 是全域變數**

**n = n + 1 # 每次執行 add 時將 n 增加 1**

**a.set(n) # 設定 a 的內容**

mylabel = tk.Label(root, textvariable=a, font=('Arial',20)) # 放入標籤

mylabel.pack()

# Button 設定 command 參數

btn = tk.Button(root,

text='我是按鈕',

font=('Arial',30,'bold'),

**command=add**

)

btn.pack()

#單選按鈕

radio\_btn1 = tk.Radiobutton(root, text='Apple') # 放入第一個單選按鈕

radio\_btn1.pack()

radio\_btn2 = tk.Radiobutton(root, text='Banana') # 放入第二個單選按鈕

radio\_btn2.pack()

| **參數** | **說明** |
| --- | --- |
| **variable** | **綁定哪個變數。** |
| **value** | **選擇該按鈕時回傳到變數的值。** |
| command | 按下按鈕要執行的函式。 |
| state | 按鈕狀態，tk.NORMAL 可使用 ( 預設 )、tk.DISABLED 不可使用。 |

**val = tk.StringVar()**

radio\_btn1 = tk.Radiobutton(root, text='Apple',variable=val, value='Apple')

radio\_btn1.pack()

radio\_btn1.select() # 搭配 select() 方法選取 radio\_btn1

radio\_btn2 = tk.Radiobutton(root, text='Banana',variable=val, value='Banana')

radio\_btn2.pack()

# 放入 Label 標籤，設定 textvariable=val

mylabel = tk.Label(root, textvariable=val, font=('Arial',30), fg='#f00')

mylabel.pack()

#複選按鈕

check\_btn1 = tk.Checkbutton(root, text='Apple') # 放入第一個按鈕

check\_btn1.pack()

check\_btn2 = tk.Checkbutton(root, text='Banana') # 放入第二個按鈕

check\_btn2.pack()

#應用

def show():

print(var1.get(), var2.get()) # 使用 get() 方法取得變數內容

var1 = tk.StringVar() # 設定文字變數，並綁定第一個 Checkbutton

check\_btn1 = tk.Checkbutton(root, text='Apple',

variable=var1, **onvalue='Apple', offvalue='--',**

command=show)

check\_btn1.pack()

check\_btn1.deselect() # 開始時不要勾選

var2 = tk.StringVar() # 設定文字變數，並綁定第二個 Checkbutton

check\_btn2 = tk.Checkbutton(root, text='Banana',

variable=var2, **onvalue='Banana', offvalue='--',**

command=show)

check\_btn2.pack()

check\_btn2.deselect() # 開始時不要勾選

**補充 label或button加入圖片**

**方法 1：使用 PhotoImage**

適用於 .png 和 .gif 格式：

# 加載圖片

image = tk.PhotoImage(file="image.png") # 替換成你的圖片路徑

# 在 Label 上顯示圖片

label = tk.Label(root, image=image)

label.pack(pady=10)

# 在 Button 上顯示圖片

button = tk.Button(root, image=image)

button.pack(pady=10)

**方法 2：使用 PIL（支援 .jpg）**

如果要顯示 .jpg、.png、.bmp 等格式，可以使用 PIL 的 ImageTk：

python

複製編輯

import tkinter as tk

from PIL import Image, ImageTk  
**(py –m pip install Pillow)**

root = tk.Tk()

root.title("顯示圖片")

# 讀取圖片並轉換為 Tkinter 格式

image = Image.open("image.jpg") # 替換成你的圖片路徑

image = image.resize((100, 100)) # 可調整大小

tk\_image = ImageTk.PhotoImage(image)

# 在 Label 上顯示圖片

label = tk.Label(root, image=tk\_image)

label.pack(pady=10)

# 在 Button 上顯示圖片

button = tk.Button(root, image=tk\_image)

button.pack(pady=10)

root.mainloop()