**Listbox**

**Listbox 方法能在視窗物件中建立 Listbox 列表選擇框**，**建立 Listbox 後使用 insert() 方法加入選項，最後再用 pack() 方法將其加入視窗中**。

listbox = tk.Listbox(root) # 放入列表選擇框

listbox.insert(1, 'Apple') # 第一個選項

listbox.insert(2, 'Banana') # 第二個選項

listbox.insert(3, 'Orange') # 第三個選項

listbox.pack()

insert 的第一個參數為選項的位置，如果不知道位置但又希望選項按照順序插入，可以**使用 tk.END 作為順序，就會將選項依序放在最後，如果使用 0，就會從第一個位置插入**。

listbox = tk.Listbox(root) # 放入列表選擇框

listbox.insert(tk.END, 'Apple') # 第一個選項

listbox.insert(tk.END, 'Banana') # 第二個選項

listbox.insert(tk.END, 'Orange') # 第三個選項

listbox.pack()

**Listbox 參數設定**

|  |  |
| --- | --- |
| **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。 |
| width | 寬度，單位是字元數。 |
| height | 高度，單位是字元數。 |
| 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。 |

下方是 Listbox 列表選擇框行為相關的參數：

|  |  |
| --- | --- |
| **參數** | **說明** |
| listvariable | 綁定清單變數，方便動態存取 Listbox 內容。 |
| selectmode | 選取模式，可以設定 browse ( 預設，等同單選 )、single ( 單選 )、multiple ( 多選 ) 和 extended ( 等同多選，但可用滑鼠拖曳選取 )。 |
| selectbackground | 選取時的背景色。 |
| xscrollcommand | 與 set() 方法搭配，將 Listbox 加上水平捲軸。 |
| yscrollcommand | 與 set() 方法搭配，將 Listbox 加上垂直捲軸。 |

## **Listbox 操作方法**

透過下列方法，可以進行 Listbox 列表選擇框的相關操作：

|  |  |  |
| --- | --- | --- |
| **方法** | **參數** | **說明** |
| insert() | index, option | 在指定位置 ( index ) 加入選項 ( option )。 |
| delete() | first, last=None | 刪除 first 到 last 之間的項目，如果沒有 last 則只會刪除 first 的項目。 |
| get() | first, last=None | 取得 first 到 last 之間的項目，如果沒有 last 則只會取得 first 的項目。 |
| size() |  | 回傳項目總數。 |
| curselection() |  | 以 tuple 格式回傳選取的項目索引。 |
| activate() | index | 選取 index 項目。 |
| selection\_set() | index | 設定 index 項目為被選取的狀態。 |
| selection\_includes() | index | 檢查 index 項目是否有被選取。 |

**一次添加多個選項**

如果有多個項目需要添加，可以先將項目設為串列的形式，再使用 for 迴圈依序加入。

listbox = tk.Listbox(root)

menu = ['Apple','Banana','Orange','Grap','Papaya','Coconut'] # 建立選單的串列

for i in menu:

listbox.insert(tk.END, i) # 使用 for 迴圈添加選項

listbox.pack()

root.mainloop()

除了使用迴圈，也可以透過 listvariable 指定變數，當變數內容為清單項目時，就會自動將清單項目加入。

menu = tk.StringVar() # 設定 tk 文字變數

menu.set(('Apple','Banana','Orange','Grap','Papaya','Coconut')) # 加入清單內容

listbox = tk.Listbox(root, listvariable=menu) # 設定 listvariable=a

listbox.pack()

## **點擊按鈕，顯示清單選擇項目**

參考「[Label 標籤](https://steam.oxxostudio.tw/category/python/tkinter/label.html)」和「[Button 按鈕](https://steam.oxxostudio.tw/category/python/tkinter/button.html)」文章，搭配 Listbox 的 curselection() 取得選取項目以及 get() 取得項目內容，就能在點擊按鈕時，透過 Label 顯示 Listbox 選取的項目。

# 定義 Label 顯示選取內容的函式

def show():

n, = listbox.curselection() # 取得項目索引值，因為是單選，回傳 (i,)，所以使用 n, 取值

text.set(listbox.get(n)) # 設定文字變數內容為該索引值對應的內容

text = tk.StringVar() # 設定文字變數

label = tk.Label(root, textvariable=text) # 放入 Label

label.pack()

menu = tk.StringVar()

menu.set(('Apple','Banana','Orange','Grap','Papaya','Coconut'))

listbox = tk.Listbox(root, listvariable=menu, height=8)

listbox.pack()

btn = tk.Button(root, text='顯示', command=show) # 放入 Button，設定 command 參數

btn.pack()

## 

**Listbox 加入捲軸**

Listbox 要加入捲軸比較複雜，需要**先添加 Frame 頁框元件，再將 Listbox 和額外添加的 Scrollbar 捲軸元件放在 Frame 頁框裡，接著將 Listbox 和 Scrollbar 綁定**，就能實現 Listbox 捲軸效果。

frame = tk.Frame(root, width=15) # 加入頁框元件，設定寬度

frame.pack()

scrollbar = tk.Scrollbar(frame) # 在頁框中加入捲軸元件

scrollbar.pack(side='right', fill='y') # 設定捲軸的位置以及填滿方式

menu = tk.StringVar()

menu.set(('Apple','Banana','Orange','Grap','Papaya','Coconut','Pear','Nuts'))

# 在頁框中加入 Listbox 元件，設定 yscrollcommand = scrollbar.set

listbox = tk.Listbox(frame, listvariable=menu, height=6, width=15, yscrollcommand = scrollbar.set)

listbox.pack(side='left', fill='y') # 設定 Listbox 的位置以及填滿方式

scrollbar.config(command = listbox.yview) # 設定 scrollbar 的 command = listbox.yview

## **OptionMenu**

**透過 OptionMenu 方法，就能在視窗物件中建立 OptionMenu 下拉選單**，必要的參數有三個，第一個表示要加入的視窗物件，第二個是 tk 變數，表示選單選擇以及顯示的項目，第三個開始是選單內容。

optionList = ['七龍珠','海賊王','鬼滅之刃','灌籃高手','排球少年'] # 選項

value = tk.StringVar() # 取值

value.set('')

menu = tk.OptionMenu(root, value, \*optionList) # 第二個參數是取值，第三個開始是選項，使用星號展開

menu.pack()

## **OptionMenu 參數設定**

加入 OptionMenu 之後，可以透過 OptionMenu 的參數調整內容的樣式，下方列出 OptionMenu 和其他元件相同的參數：

|  |  |
| --- | --- |
| **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。 |
| width | 寬度，單位是字元數。 |
| height | 高度，單位是字元數。 |
| 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。 |

## **OptionMenu 操作方法**

OptionMenu 操作方法和其他元件的用法不同，必須要在外部建立一個變數，透過變數 trace 的方式取值，trace 有兩個參數，第一個是追蹤的模式，第二個是追蹤事件發生後要執行的函式，追蹤模式如下：

|  |  |
| --- | --- |
| **模式** | **說明** |
| w | 當選單的值被更改或選擇。 |
| r | 當選單被讀取。 |
| u | 當選單被刪除。 |

此外，也因為 OptionMenu 的參數已經提供給選單「內容」使用，**因此如果要改變樣式，則必須透過 config 的方式設定**，下方的程式執行後，會在選擇 OptionMenu 選項時，印出所選的內容。

def show(\*e):

a.set(value.get()) # Label 變數改變成選單內容，使用 get() 取值

a = tk.StringVar() # Label 變數

a.set('七龍珠')

label = tk.Label(root, textvariable=a) # 建立 Label，文字為變數

label.pack()

optionList = ['七龍珠','海賊王','鬼滅之刃','灌籃高手','排球少年']

value = tk.StringVar()

value.set('七龍珠')

menu = tk.OptionMenu(root, value, \*optionList) # 選單

**menu.config(width=10, fg='#f00') # 設定樣式**

menu.pack()

**value.trace('w', show)**  # 變數 trace 是否改變，若有改變執行 show

## **顯示 OptionMenu 選取內容**

**如果不想用 trace 的方式**，也可改用單純點擊 Button 後執行函式的做法，就可以在選單變更後，點擊按鈕顯示選擇內容。

def show(\*e):

a.set(value.get())

a = tk.StringVar()

a.set('')

label = tk.Label(root, textvariable=a)

label.pack()

optionList = ['七龍珠','海賊王','鬼滅之刃','灌籃高手','排球少年']

value = tk.StringVar()

value.set('七龍珠')

menu = tk.OptionMenu(root, value, \*optionList)

menu.config(width=10, fg='#f00')

menu.pack()

**button = tk.Button(root, text='顯示', command=show)**

**button.pack()**