Pack配置

pack() 後就會按照執行的順序，由上而下放置

pack() 的 fill 參數預設使用元件寬度，設定 fill='x' 表示和放置的父元件同寬 ( 必須搭配 side 參數設定預設值、top、bottom 或 expand 參數 )，設定 fill='y' 表示和放置的父元件同高 ( 必須搭配 side 參數設定 left、right 或 expand 參數 )，設定 fill='both' 表示同時撐滿水平和垂直方向。

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

a.pack(fill='x')

b.pack(fill='y', side='left')

**expand 參數**

如果要讓兩個元件一個一半的撐滿某個方向，可以使用 fill 參數配 expand 參數，expand 預設為 0 表示使用元件本身長寬，若設定為 1 則會展開。

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

a.pack(fill='y', expand=1)

b.pack(fill='both', expand=1)

**padx、pady、ipadx、ipady 參數**

pack() 的 padx 參數表示左右外邊距，pady 表示上下外邊距，ipadx 表示左右內邊距，ipady 表示上下內邊距，四個參數預設值均為 0，下方的程式碼執行後，會表現出不同參數設定的結果。

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

c = tk.Label(root, text='CCC', background='#fc0')

d = tk.Label(root, text='DDD', background='#f9c')

e = tk.Label(root, text='EEE', background='#aaa')

a.pack(fill='x', padx=20)

b.pack(ipadx=20)

c.pack(fill='x', ipady=20)

d.pack(ipady=20)

e.pack()

**side 參數**

pack() 的 side 參數會設定該元件放置在 left 左邊、right 右邊、top 上面、bottom 下面，執行的順序決定放置的順序。

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

c = tk.Label(root, text='CCC', background='#fc0')

d = tk.Label(root, text='DDD', background='#f9c')

e = tk.Label(root, text='EEE', background='#aaa')

a.pack(side='left')

b.pack(side='left')

c.pack(side='top')

d.pack(side='bottom')

e.pack(side='right')

**Grid 格狀版面佈局**

grid() 方法，採用類似「表格」的方式放置元件。

**注意，在同一個視窗元件中，不能同時使用 pack() 和 grid()。**

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

c = tk.Label(root, text='CCC', background='#fc0')

d = tk.Label(root, text='DDD', background='#0c9')

a.grid(column=0, row=0) # 放在 (0,0)

b.grid(column=1, row=0) # 放在 (1,0)

c.grid(column=0, row=1) # 放在 (0,1)

d.grid(column=1, row=1) # 放在 (1,1)

**column、row 參數**

grid() 的 column 和 row 參數表示放置的元件位置，column 表示水平方向，左上角起始為 0，row 表示垂直方向，左上角起始為 0，同一個視窗元件 ( 父元件 ) 中只能使用一個 grid，放置後會根據所有元件的 column 和 row 產生對應的表格佈局，以上方的例子為例，會產生 2x2 的佈局，如果使用下方的程式碼，則會產生 3x2 的佈局 ( 因為只有四個元件，所以有兩格會留空 )。

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

c = tk.Label(root, text='CCC', background='#fc0')

d = tk.Label(root, text='DDD', background='#0c9')

a.grid(column=0, row=0) # 放在 (0,0)

b.grid(column=2, row=0) # 放在 (2,0)

c.grid(column=0, row=1) # 放在 (0,1)

d.grid(column=1, row=1) # 放在 (1,1)

**columnspan、rowspan 參數**

如果要讓「多格可以合併成一格」，可以使用 columnspan 或 rowspan 參數，參數預設值為 1，若設定大於 1 的數字表示要合併幾格。

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

c = tk.Label(root, text='CCC', background='#fc0')

d = tk.Label(root, text='DDD', background='#0c9')

e = tk.Label(root, text='EEE', background='#ccc')

a.grid(column=0, row=0, columnspan=2)

b.grid(column=2, row=0, rowspan=2)

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

c.grid(column=0, row=2)

d.grid(column=1, row=2)

**padx、pady、ipadx、ipady 參數**

grid() 的 padx 參數表示左右外邊距，pady 表示上下外邊距，ipadx 表示左右內邊距，ipady 表示上下內邊距，四個參數預設值均為 0，下方的程式碼執行後，會表現出不同參數設定的結果。

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

c = tk.Label(root, text='CCC', background='#fc0')

d = tk.Label(root, text='DDD', background='#0c9')

e = tk.Label(root, text='EEE', background='#ccc')

a.grid(column=0, row=0, padx=20)

b.grid(column=1, row=0, pady=20)

c.grid(column=0, row=1, ipadx=20)

d.grid(column=1, row=1, ipady=20)

**sticky 參數**

grid() 的 sticky 參數會元件放置的位置，設定方式預設有四個：tk.W 右邊、tk.E 左邊、tk.N 上面、tk.S 下面，除了上下左右，也可使用「+」號連結，例如 tk.E+tk.N 表示放在左上，tk.S+tk.N+tk.W 表示靠右垂直置中，下方的程式碼執行後，會表現出不同參數設定的結果。

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

c = tk.Label(root, text='CCC', background='#fc0')

d = tk.Label(root, text='DDD', background='#0c9')

e = tk.Label(root, text='EEE', background='#ccc')

a.grid(column=0, row=0, sticky=tk.W+tk.S)

b.grid(column=1, row=0, ipady=20)

c.grid(column=0, row=1, sticky=tk.E+tk.N)

d.grid(column=1, row=1, ipady=20)

e.grid(column=0, row=2, ipadx=20)

**同時使用 pack() 和 grid()**

因為在同一個視窗元件 ( 父元件 ) 中，不能同時使用 pack() 和 grid()，所以如果要同時使用，必須搭配 frame 來進行放置，下方的例子會將 abcd 四個 Label 元件使用 grid 的方式**放在一個 Frame 元件中，再將這個 Frame 與另外一個 Label 放在 root 視窗裡**。

a = tk.Label(frame, text='AAA', background='#f90')

b = tk.Label(frame, text='BBB', background='#09c')

c = tk.Label(frame, text='CCC', background='#fc0')

d = tk.Label(frame, text='DDD', background='#0c9')

a.grid(column=0, row=0)

b.grid(column=1, row=0)

c.grid(column=0, row=1)

d.grid(column=1, row=1)

**place配置方法**

place() 的佈局方法，讓使用者可以定義「絕對位置」和「相對位置」的座標進行元件定位

**注意，使用 place() 方法只能放在主視窗 root 裡，不能放在其他元件 ( 如 Frame 中 )。**

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

c = tk.Label(root, text='CCC', background='#fc0')

d = tk.Label(root, text='DDD', background='#0c9')

a.place(x=0, y=0) # 放在 (0,0)

b.place(x=50, y=50) # 放在 (50,50)

c.place(x=100, y=100) # 放在 (100,100)

d.place(x=150, y=150) # 放在 (150,150)

**x、y 參數**

place() 的 x 和 y 參數表示放置的元件在主視窗的絕對位置，視窗左上角為 (0,0)，往右為正，往下為正，下方的程式碼執行後，會有四個 Label 放在指定的位置上。

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

c = tk.Label(root, text='CCC', background='#fc0')

d = tk.Label(root, text='DDD', background='#0c9')

a.place(x=0, y=0) # 放在 (0,0)

b.place(x=50, y=50) # 放在 (50,50)

c.place(x=100, y=100) # 放在 (100,100)

d.place(x=150, y=150) # 放在 (150,150)

**relx、rely 參數**

place() 的 relx 和 rely 參數表示「比例」，也就是元件左上角座標在視窗中，按照比例算出來的座標，數值範圍 0～1，例如視窗寬度如果是 100，relx 設定 0.5，x 座標就會在 100x0.5=50 的位置。

a = tk.Label(root, text='AAA', background='#f90')

b = tk.Label(root, text='BBB', background='#09c')

c = tk.Label(root, text='CCC', background='#fc0')

d = tk.Label(root, text='DDD', background='#0c9')

a.place(relx=0.1, rely=0.1)

b.place(relx=0.3, rely=0.3)

c.place(relx=0.5, rely=0.5)

d.place(relx=0.7, rely=0.7)