

Hướng dẫn tạo giao diện Tkinter với method Pack

Giáo viên: Nguyễn Hùng Cường

Ví dụ 1: Viết mã để tạo giao diện

Trong bài này, ta tạo một form cho phép hiển thị 3 Frame thể hiện 3 vùng có màu nền khác nhau được hiển thị trên một form giao diện của Tkinter

```
import tkinter as tk

win = tk.Tk()
win.title("Demo Pack method")
win.geometry("300x200")

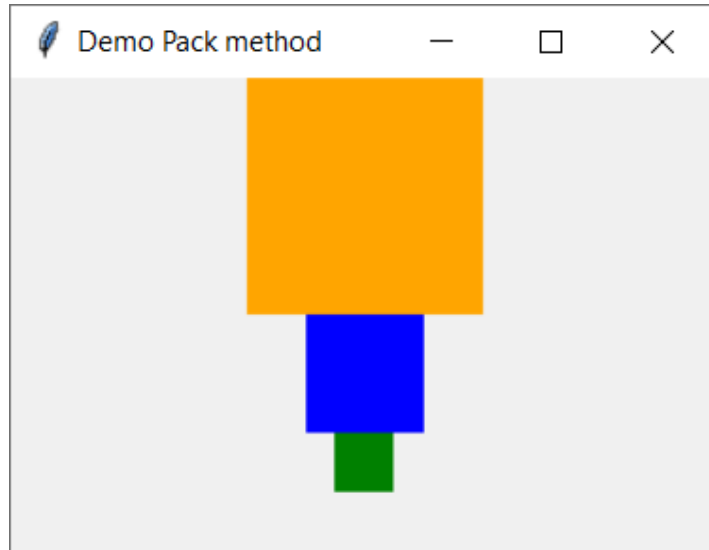
# add an orange frame
frame1 = tk.Frame(master=win, width=100, height=100, bg="orange")
frame1.pack()

# add blue frame
frame2 = tk.Frame(master=win, width=50, height=50, bg="blue")
frame2.pack()

# add green frame
frame3 = tk.Frame(master=win, width=25, height=25, bg="green")
frame3.pack()

win.mainloop()
```

Sau đó, thực thi chương trình ta có thể thấy, form đã được hiển thị với 3 vùng (Frame) có màu nền khác nhau, sử dụng method pack() đúng như mong muốn.



Ví dụ 2: Hiển thị các vùng trên form giao diện trong đó có sử dụng thuộc tính Fill

Trong ví dụ này, ta đã tạo ra một form bao gồm 3 vùng (Frame) có màu nền khác nhau, nhưng đồng thời ta cũng cho phép giãn 3 vùng này theo chiều ngang thông qua thuộc tính fill=X như sau

```
import tkinter as tk

win = tk.Tk()
win.title("Demo Pack method")
win.geometry("300x200")

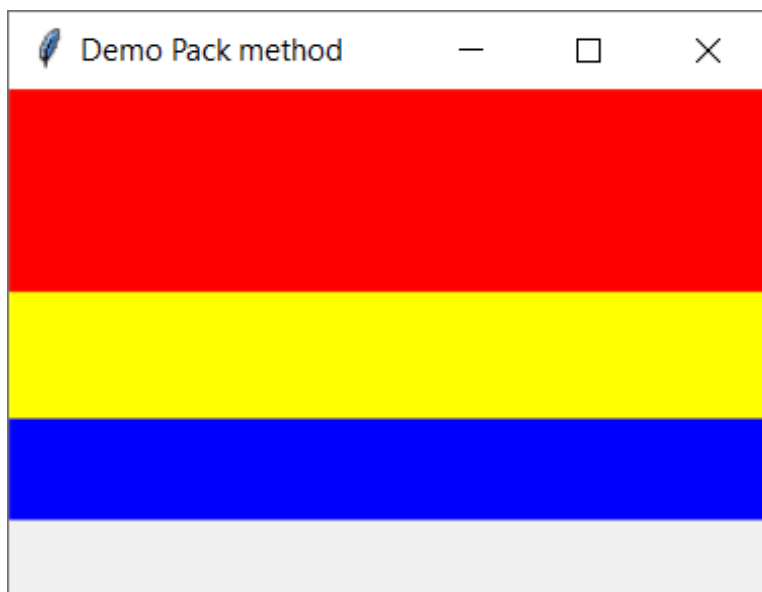
frame1 = tk.Frame(master=win, height=80, bg="red")
# adding the fill argument with
# horizontal fill value
frame1.pack(fill=tk.X)

frame2 = tk.Frame(master=win, height=50, bg="yellow")
frame2.pack(fill=tk.X)

frame3 = tk.Frame(master=win, height=40, bg="blue")
frame3.pack(fill=tk.X)

win.mainloop()
```

Sau khi thực thi chương trình, ta có thể thấy, chương trình đã được hiển thị đúng như mong muốn.



Ví dụ 3: Tạo form giao diện và căn các phần tử trên form với method pack()

Trong ví dụ này, ta sẽ tạo một form gồm nhiều phần tử giao diện, đồng thời ta sẽ căn các phần tử hiển thị theo một cạnh nào đó của form như sau.

```
import tkinter as tk
from tkinter import ttk

root = tk.Tk()
root.title('Pack Demo')
root.geometry("300x200")

# place widgets top down
label1 = tk.Label(
    root,
    text='Box 1',
    bg="red",
    fg="white"
)

label1.pack(
    padx=10,
    pady=10,
    fill='x'
)
```

```
label2 = tk.Label(
    root,
    text='Box 2',
    bg="green",
    fg="white"
)

label2.pack(
    padx=10,
    pady=10,
    fill='x'
)

label3 = tk.Label(
    root,
    text='Box 3',
    bg="blue",
    fg="white"
)

label3.pack(
    padx=10,
    pady=10,
    fill='x'
)
```

```
# place widgets side by side

label4 = tk.Label(
    root,
    text='Left',
    bg="cyan",
    fg="black"
)

label4.pack(
    padx=10,
    pady=10,
    expand=True,
    fill='both',
    side='left'
)
```

```
label5 = tk.Label(
    root,
    text='Center',
    bg="magenta",
    fg="black"
)

label5.pack(
    padx=10,
    pady=10,
    expand=True,
    fill='both',
    side='left'
)

label6 = tk.Label(
    root,
    text='Right',
    bg="yellow",
    fg="black"
)
```

```
label6.pack(  
    ipadx=10,  
    ipady=10,  
    expand=True,  
    fill='both',  
    side='left'  
)  
  
root.mainloop()
```

Sau đó, khi thực thi chương trình, ta có thể thấy, giao diện đã được hiển thị đúng như mong muốn. Các phần tử đã được căn và fill trên giao diện như hình bên dưới.

