Top of Form



Bottom of Form

Top of Form



**Python 2: Getting More Out of Python  
Lesson 7, Quiz 2**

Handed in: 12 Mar 2015 04:03:37PM Graded: 13 Mar 2015 04:25:16PM

**Question 1:**

What configuration item should you set on a button to establish the function to run when the button is clicked?

**Your Answer:**

already answered

**Mentor Comments:**

*none*

**Question 2:**

How would you configure a button **B** so it does not respond to clicks? **B** already exists, and has a handler function that should be left as is.

**Your Answer:**

already answered

**Mentor Comments:**

*none*

**Question 3:**

How would you specify that button **B** should appear immediately below the last object created with the same parent?

**Your Answer:**

if you are not using any 'side' constants for the pack method, packing order takes precedent:

last\_frame = Frame(self)

A = Button(last\_frame, text="A")

B = Button(last\_frame, text="B")

C = Button(last\_frame, text="C")

D = Button(last\_frame, text="D")

A.pack() # object is now placed on the frame

B.pack() # called immediately after the last object ('A') so it will follow it

D.pack() # will be placed directly under 'B'

C.pack() # will be placed directly under 'C'

last\_frame.pack()

Scenario 2:

If you do provide a constant to side, then I believe that it operates on a first come first served method, taking into account that no side argument is equivalent to TOP.

last\_frame = Frame(self)

A = Button(last\_frame, text="A")

B = Button(last\_frame, text="B")

C = Button(last\_frame, text="C")

D = Button(last\_frame, text="D")

A.pack() # no argument for side, so TOP is default, this will be first in the frame

B.pack(side=BOTTOM) # packed with BOTTOM arg, first widget to call BOTTOM, so it gets first dibs at the most bottom position

D.pack(side=TOP) # tried to call TOP arg, but 'A' beat him to it; will be placed just under 'A'

C.pack(side=BOTTOM) # tried to call BOTTOM arg, but 'B' called it already, will be placed directly under 'D', but above 'B'

last\_frame.pack()

Dictating orientation via the pack method is dangerous if you mix horizontal constants (LEFT, RIGHT) with vertical constants (TOP, BOTTOM) because resizing of the window ensures there is no sound way to know what the end layout will look like, so I didn't cover that topic. I tested this with the 'texthandler.py' code from the Lesson 7 exercises.

**Mentor Comments:**

This is an awesome, encyclopedic, and well researched answer.

**Overall Comments:**

Terrific work here, Jason. Thanks for hanging in there with me.

-Pat

**Grade:**

Great

[Take Me Back](ostreturn:)   [Previous Attempt](https://students.oreillyschool.com/student/viewassignment.php?entryid=7879&attempt=2&graded=1)

© 2014, O'Reilly Media, Inc. All rights reserved.

Bottom of Form