back next

The Tkinter PanedWindow Widget

(new in Tk 8.4) The **PanedWindow** widget is a geometry manager widget, which can contain one or more child widgets (“panes”). The child widgets can be resized by the user, by moving separator lines (“sashes”) using the mouse.

When to use the PanedWindow Widget

The PanedWindow widget can be used to implement common 2-pane and 3-pane layouts.

Patterns

Here’s how to create a 2-pane widget:

```
from Tkinter import *

m = PanedWindow(orient=VERTICAL)
m.pack(fill=BOTH, expand=1)

top = Label(m, text="top pane")
m.add(top)

bottom = Label(m, text="bottom pane")
m.add(bottom)

mainloop()
```

Here’s how to create a 3-pane widget:

```
from Tkinter import *

m1 = PanedWindow()
m1.pack(fill=BOTH, expand=1)

left = Label(m1, text="left pane")
m1.add(left)

m2 = PanedWindow(m1, orient=VERTICAL)
m1.add(m2)

top = Label(m2, text="top pane")
m2.add(top)

bottom = Label(m2, text="bottom pane")
m2.add(bottom)

mainloop()
```

Reference

PanedWindow(master=None, **options) (class) [#]

A paned window manager widget. This widget manages one or more child widgets, and allows the user to resize these widgets, by moving the separators between them.

master
Parent widget.

***options*
Widget options. See the description of the config method for a list of available options.

add(child, **options) [#]

Adds a child window to the paned window.

config(options)** [#]

Modifies one or more widget options. If no options are given, the method returns a dictionary containing all current option values.

***options*
: Widget options.

background=
: Default is system specific. (background/Background)

bd=
: Same as **borderwidth**.

bg=
: Same as **background**.

borderwidth=
: Default is 2. (borderWidth/BorderWidth)

cursor=
: No default value. (cursor/Cursor)

handlepad=
: Default is 8. (handlePad/HandlePad)

handlesize=
: Default is 8. (handleSize/HandleSize)

height=
: No default value. (height/Height)

opaqueresize=
: No default value. (opaqueResize/OpaqueResize)

orient=
: Default is HORIZONTAL. (orient/Orient)

relief=
: Default is FLAT. (relief/Relief)

sashcursor=
: No default value. (sashCursor/Cursor)

sashpad=
: Default is 2. (sashPad/SashPad)

sashrelief=
: Default is RAISED. (sashRelief/Relief)

sashwidth=
: Default is 2. (sashWidth/Width)

showhandle=
: No default value. (showHandle/ShowHandle)

width=
: No default value. (width/Width)

forget(child) [#]

Removes a child window.

identify(x, y) [#]

Identifies the widget element at the given position.

panecget(child, option) [#]

Gets a child window option.

paneconfig(child, **options) [#]

Same as **paneconfigure**.

paneconfigure(child, **options) [#]

Set child window configuration options.

child
: Child window.

***options*
: Child window options.

after=
: Insert after this widget.

before=
: Insert before this widget.

height=
: Widget height.

minsize=
: Minimal size (width for horizontal panes, height for vertical panes).

padx=
: Horizontal padding.

pady=
: Vertical padding.

sticky=
: Defines how to expand a child widget if the resulting pane is larger than the widget itself. This can be any combination of the constants **S**, **N**, **E**, and **W**, or **NW**, **NE**, **SW**, and **SE**.

width=
: Widget width.

panes() [#]

Returns a list of child widgets.

Returns:
: A list of widgets.

proxy_coord() [#]

Gets the most recent proxy position.

proxy_forget() [#]

Removes the proxy.

proxy_place(x, y) [#]

Places the proxy at the given position.

remove(child) [#]

Same as **forget**.

sash_coord(index) [#]

Gets the current position for a sash (separator).

index
: Sash index (0..n).

Returns:
: The upper left corner of the sash, given as a 2-tuple (x, y).

sash_dragto(index, x, y) [#]

Drag the sash (separator) to a new position, relative to the mark. Together with **sash_mark**, this method is used by the widget bindings to move a sash by dragging the mouse. The **mark** method is called when the mouse is pressed over a sash (you can use **identify** to figure out which sash to mark), and the **dragto** method is called repeatedly when the mouse pointer is moved.

Note that this method is missing from the Tkinter bindings in Python 2.3. You can use **sash("dragto", index, x, y)** instead.

index
: Sash index (0..n).

Returns:
: The upper left corner of the sash, given as a 2-tuple (x, y).

sash_mark(index, x, y) [#]

Registers the current mouse position. See **sash_dragto** for more information.

Note that this method only takes a single argument in the Tkinter bindings shipped with in Python 2.3. To pass in all three arguments, use **sash("mark", index, x, y)**.

index
Sash index (0..n).
x
Start position.
y
Start position.

sash_place(index, x, y) [#]

Moves the sash (separator) to a given position.

index
Sash index (0..n).
x
Sash position.
y
Sash position.

back next

rendered by a django application. hosted by webfaction.