

GWT - LAYOUT PANELS

Layout Panels can contain other widgets. These panels controls the way widgets to be shown on User Interface. Every Panel widget inherits properties from Panel class which in turn inherits properties from Widget class and which in turn inherits properties from UIObject class.

S.N. Widget & Description

1

[GWT UIObject Class](#)

This widget contains text, not interpreted as HTML using a `<div>` element, causing it to be displayed with block layout.

2

[GWT Widget Class](#)

This widget can contain HTML text and displays the html content using a `<div>` element, causing it to be displayed with block layout.

3

[GWT Panel Class](#)

This is an is the abstract base class for all panels, which are widgets that can contain other widgets.

Layout Panels

Following are few important *Layout Panels*:

S.N. Widget & Description

1

[FlowPanel](#)

This widget represents a panel that formats its child widgets using the default HTML layout behavior.

2

[HorizontalPanel](#)

This widget represents a panel that lays all of its widgets out in a single horizontal column.

3

[VerticalPanel](#)

This widget represents a panel that lays all of its widgets out in a single vertical column.

4

[HorizontalSplitPanel](#)

This widget represents a panel that arranges two widgets in a single horizontal row and allows the user to interactively change the proportion of the width dedicated to each of the two widgets. Widgets contained within a `HorizontalSplitPanel` will be automatically

decorated with scrollbars when necessary.

5

[VerticalSplitPanel](#)

This widget represents a A panel that arranges two widgets in a single vertical column and allows the user to interactively change the proportion of the height dedicated to each of the two widgets. Widgets contained within a VerticalSplitPanel will be automatically decorated with scrollbars when necessary.

6

[FlexTable](#)

This widget represents a flexible table that creates cells on demand. It can be jagged *that is, each row can contain a different number of cells* and individual cells can be set to span multiple rows or columns.

7

[Grid](#)

This widget represents a A rectangular grid that can contain text, html, or a child Widget within its cells. It must be resized explicitly to the desired number of rows and columns.

8

[DeckPanel](#)

panel that displays all of its child widgets in a 'deck', where only one can be visible at a time. It is used by TabPanel.

9

[DockPanel](#)

This widget represents a panel that lays its child widgets out "docked" at its outer edges, and allows its last widget to take up the remaining space in its center.

10

[HTMLPanel](#)

This widget represents a panel that contains HTML, and which can attach child widgets to identified elements within that HTML.

11

[TabPanel](#)

This widget represents a panel that represents a tabbed set of pages, each of which contains another widget. Its child widgets are shown as the user selects the various tabs associated with them. The tabs can contain arbitrary HTML.

12

[Composite](#)

This widget represents a type of widget that can wrap another widget, hiding the wrapped widget's methods. When added to a panel, a composite behaves exactly as if the widget it wraps had been added.

13

[SimplePanel](#)

This widget represents a Base class for panels that contain only one widget.

14

[ScrollPane](#)

This widget represents a simple panel that wraps its contents in a scrollable area

15

[FocusPanel](#)

This widget represents a simple panel that makes its contents focusable, and adds the ability to catch mouse and keyboard events.

16

[FormPanel](#)

This widget represents a panel that wraps its contents in an HTML <FORM> element.

17

[PopupPanel](#)

This widget represents a panel that can **pop up** over other widgets. It overlays the browser's client area *and any previously – created popups*.

18

[DialogBox](#)

This widget represents a form of popup that has a caption area at the top and can be dragged by the user. Unlike a PopupPanel, calls to PopupPanel.setWidthString and PopupPanel.setHeightString will set the width and height of the dialog box itself, even if a widget has not been added as yet.

>Loading [MathJax]/jax/output/HTML-CSS/fonts/TeX/fontdata.js