| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/FlowLayout.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/FileDialog.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/FocusTraversalPolicy.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/FlowLayout.html)    [**NO FRAMES**](http://docs.google.com/FlowLayout.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#lnxbz9) | [METHOD](#2jxsxqh) |

## **java.awt**

Class FlowLayout

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.awt.FlowLayout**

**All Implemented Interfaces:** [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

public class **FlowLayout**extends [Object](http://docs.google.com/java/lang/Object.html)implements [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

A flow layout arranges components in a directional flow, much like lines of text in a paragraph. The flow direction is determined by the container's componentOrientation property and may be one of two values:

* ComponentOrientation.LEFT\_TO\_RIGHT
* ComponentOrientation.RIGHT\_TO\_LEFT

Flow layouts are typically used to arrange buttons in a panel. It arranges buttons horizontally until no more buttons fit on the same line. The line alignment is determined by the align property. The possible values are:

* [LEFT](http://docs.google.com/java/awt/FlowLayout.html#LEFT)
* [RIGHT](http://docs.google.com/java/awt/FlowLayout.html#RIGHT)
* [CENTER](http://docs.google.com/java/awt/FlowLayout.html#CENTER)
* [LEADING](http://docs.google.com/java/awt/FlowLayout.html#LEADING)
* [TRAILING](http://docs.google.com/java/awt/FlowLayout.html#TRAILING)

For example, the following picture shows an applet using the flow layout manager (its default layout manager) to position three buttons:



Here is the code for this applet:

import java.awt.\*;  
 import java.applet.Applet;  
  
 public class myButtons extends Applet {  
 Button button1, button2, button3;  
 public void init() {  
 button1 = new Button("Ok");  
 button2 = new Button("Open");  
 button3 = new Button("Close");  
 add(button1);  
 add(button2);  
 add(button3);  
 }  
 }

A flow layout lets each component assume its natural (preferred) size.

**Since:** JDK1.0 **See Also:**[ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.FlowLayout)

| **Field Summary** | |
| --- | --- |
| static int | [**CENTER**](http://docs.google.com/java/awt/FlowLayout.html#CENTER)            This value indicates that each row of components should be centered. |
| static int | [**LEADING**](http://docs.google.com/java/awt/FlowLayout.html#LEADING)            This value indicates that each row of components should be justified to the leading edge of the container's orientation, for example, to the left in left-to-right orientations. |
| static int | [**LEFT**](http://docs.google.com/java/awt/FlowLayout.html#LEFT)            This value indicates that each row of components should be left-justified. |
| static int | [**RIGHT**](http://docs.google.com/java/awt/FlowLayout.html#RIGHT)            This value indicates that each row of components should be right-justified. |
| static int | [**TRAILING**](http://docs.google.com/java/awt/FlowLayout.html#TRAILING)            This value indicates that each row of components should be justified to the trailing edge of the container's orientation, for example, to the right in left-to-right orientations. |

| **Constructor Summary** | |
| --- | --- |
| [**FlowLayout**](http://docs.google.com/java/awt/FlowLayout.html#FlowLayout())()            Constructs a new FlowLayout with a centered alignment and a default 5-unit horizontal and vertical gap. |
| [**FlowLayout**](http://docs.google.com/java/awt/FlowLayout.html#FlowLayout(int))(int align)            Constructs a new FlowLayout with the specified alignment and a default 5-unit horizontal and vertical gap. |
| [**FlowLayout**](http://docs.google.com/java/awt/FlowLayout.html#FlowLayout(int,%20int,%20int))(int align, int hgap, int vgap)            Creates a new flow layout manager with the indicated alignment and the indicated horizontal and vertical gaps. |

| **Method Summary** | |
| --- | --- |
| void | [**addLayoutComponent**](http://docs.google.com/java/awt/FlowLayout.html#addLayoutComponent(java.lang.String,%20java.awt.Component))([String](http://docs.google.com/java/lang/String.html) name, [Component](http://docs.google.com/java/awt/Component.html) comp)            Adds the specified component to the layout. |
| int | [**getAlignment**](http://docs.google.com/java/awt/FlowLayout.html#getAlignment())()            Gets the alignment for this layout. |
| boolean | [**getAlignOnBaseline**](http://docs.google.com/java/awt/FlowLayout.html#getAlignOnBaseline())()            Returns true if components are to be vertically aligned along their baseline. |
| int | [**getHgap**](http://docs.google.com/java/awt/FlowLayout.html#getHgap())()            Gets the horizontal gap between components and between the components and the borders of the Container |
| int | [**getVgap**](http://docs.google.com/java/awt/FlowLayout.html#getVgap())()            Gets the vertical gap between components and between the components and the borders of the Container. |
| void | [**layoutContainer**](http://docs.google.com/java/awt/FlowLayout.html#layoutContainer(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Lays out the container. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumLayoutSize**](http://docs.google.com/java/awt/FlowLayout.html#minimumLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Returns the minimum dimensions needed to layout the *visible* components contained in the specified target container. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredLayoutSize**](http://docs.google.com/java/awt/FlowLayout.html#preferredLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) target)            Returns the preferred dimensions for this layout given the *visible* components in the specified target container. |
| void | [**removeLayoutComponent**](http://docs.google.com/java/awt/FlowLayout.html#removeLayoutComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) comp)            Removes the specified component from the layout. |
| void | [**setAlignment**](http://docs.google.com/java/awt/FlowLayout.html#setAlignment(int))(int align)            Sets the alignment for this layout. |
| void | [**setAlignOnBaseline**](http://docs.google.com/java/awt/FlowLayout.html#setAlignOnBaseline(boolean))(boolean alignOnBaseline)            Sets whether or not components should be vertically aligned along their baseline. |
| void | [**setHgap**](http://docs.google.com/java/awt/FlowLayout.html#setHgap(int))(int hgap)            Sets the horizontal gap between components and between the components and the borders of the Container. |
| void | [**setVgap**](http://docs.google.com/java/awt/FlowLayout.html#setVgap(int))(int vgap)            Sets the vertical gap between components and between the components and the borders of the Container. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/awt/FlowLayout.html#toString())()            Returns a string representation of this FlowLayout object and its values. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### LEFT

public static final int **LEFT**

This value indicates that each row of components should be left-justified.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.FlowLayout.LEFT)

### CENTER

public static final int **CENTER**

This value indicates that each row of components should be centered.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.FlowLayout.CENTER)

### RIGHT

public static final int **RIGHT**

This value indicates that each row of components should be right-justified.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.FlowLayout.RIGHT)

### LEADING

public static final int **LEADING**

This value indicates that each row of components should be justified to the leading edge of the container's orientation, for example, to the left in left-to-right orientations.

**Since:** 1.2 **See Also:**[Component.getComponentOrientation()](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.FlowLayout.LEADING)

### TRAILING

public static final int **TRAILING**

This value indicates that each row of components should be justified to the trailing edge of the container's orientation, for example, to the right in left-to-right orientations.

**Since:** 1.2 **See Also:**[Component.getComponentOrientation()](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.FlowLayout.TRAILING)

| **Constructor Detail** |
| --- |

### FlowLayout

public **FlowLayout**()

Constructs a new FlowLayout with a centered alignment and a default 5-unit horizontal and vertical gap.

### FlowLayout

public **FlowLayout**(int align)

Constructs a new FlowLayout with the specified alignment and a default 5-unit horizontal and vertical gap. The value of the alignment argument must be one of FlowLayout.LEFT, FlowLayout.RIGHT, FlowLayout.CENTER, FlowLayout.LEADING, or FlowLayout.TRAILING.

**Parameters:**align - the alignment value

### FlowLayout

public **FlowLayout**(int align,  
 int hgap,  
 int vgap)

Creates a new flow layout manager with the indicated alignment and the indicated horizontal and vertical gaps.

The value of the alignment argument must be one of FlowLayout.LEFT, FlowLayout.RIGHT, FlowLayout.CENTER, FlowLayout.LEADING, or FlowLayout.TRAILING.

**Parameters:**align - the alignment valuehgap - the horizontal gap between components and between the components and the borders of the Containervgap - the vertical gap between components and between the components and the borders of the Container

| **Method Detail** |
| --- |

### getAlignment

public int **getAlignment**()

Gets the alignment for this layout. Possible values are FlowLayout.LEFT, FlowLayout.RIGHT, FlowLayout.CENTER, FlowLayout.LEADING, or FlowLayout.TRAILING.

**Returns:**the alignment value for this layout**Since:** JDK1.1 **See Also:**[setAlignment(int)](http://docs.google.com/java/awt/FlowLayout.html#setAlignment(int))

### setAlignment

public void **setAlignment**(int align)

Sets the alignment for this layout. Possible values are

* FlowLayout.LEFT
* FlowLayout.RIGHT
* FlowLayout.CENTER
* FlowLayout.LEADING
* FlowLayout.TRAILING

**Parameters:**align - one of the alignment values shown above**Since:** JDK1.1 **See Also:**[getAlignment()](http://docs.google.com/java/awt/FlowLayout.html#getAlignment())

### getHgap

public int **getHgap**()

Gets the horizontal gap between components and between the components and the borders of the Container

**Returns:**the horizontal gap between components and between the components and the borders of the Container**Since:** JDK1.1 **See Also:**[setHgap(int)](http://docs.google.com/java/awt/FlowLayout.html#setHgap(int))

### setHgap

public void **setHgap**(int hgap)

Sets the horizontal gap between components and between the components and the borders of the Container.

**Parameters:**hgap - the horizontal gap between components and between the components and the borders of the Container**Since:** JDK1.1 **See Also:**[getHgap()](http://docs.google.com/java/awt/FlowLayout.html#getHgap())

### getVgap

public int **getVgap**()

Gets the vertical gap between components and between the components and the borders of the Container.

**Returns:**the vertical gap between components and between the components and the borders of the Container**Since:** JDK1.1 **See Also:**[setVgap(int)](http://docs.google.com/java/awt/FlowLayout.html#setVgap(int))

### setVgap

public void **setVgap**(int vgap)

Sets the vertical gap between components and between the components and the borders of the Container.

**Parameters:**vgap - the vertical gap between components and between the components and the borders of the Container**Since:** JDK1.1 **See Also:**[getVgap()](http://docs.google.com/java/awt/FlowLayout.html#getVgap())

### setAlignOnBaseline

public void **setAlignOnBaseline**(boolean alignOnBaseline)

Sets whether or not components should be vertically aligned along their baseline. Components that do not have a baseline will be centered. The default is false.

**Parameters:**alignOnBaseline - whether or not components should be vertically aligned on their baseline**Since:** 1.6

### getAlignOnBaseline

public boolean **getAlignOnBaseline**()

Returns true if components are to be vertically aligned along their baseline. The default is false.

**Returns:**true if components are to be vertically aligned along their baseline**Since:** 1.6

### addLayoutComponent

public void **addLayoutComponent**([String](http://docs.google.com/java/lang/String.html) name,  
 [Component](http://docs.google.com/java/awt/Component.html) comp)

Adds the specified component to the layout. Not used by this class.

**Specified by:**[addLayoutComponent](http://docs.google.com/java/awt/LayoutManager.html#addLayoutComponent(java.lang.String,%20java.awt.Component)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**name - the name of the componentcomp - the component to be added

### removeLayoutComponent

public void **removeLayoutComponent**([Component](http://docs.google.com/java/awt/Component.html) comp)

Removes the specified component from the layout. Not used by this class.

**Specified by:**[removeLayoutComponent](http://docs.google.com/java/awt/LayoutManager.html#removeLayoutComponent(java.awt.Component)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**comp - the component to remove**See Also:**[Container.removeAll()](http://docs.google.com/java/awt/Container.html#removeAll())

### preferredLayoutSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **preferredLayoutSize**([Container](http://docs.google.com/java/awt/Container.html) target)

Returns the preferred dimensions for this layout given the *visible* components in the specified target container.

**Specified by:**[preferredLayoutSize](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**target - the container that needs to be laid out **Returns:**the preferred dimensions to lay out the subcomponents of the specified container**See Also:**[Container](http://docs.google.com/java/awt/Container.html), [minimumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/FlowLayout.html#minimumLayoutSize(java.awt.Container)), [Container.getPreferredSize()](http://docs.google.com/java/awt/Container.html#getPreferredSize())

### minimumLayoutSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **minimumLayoutSize**([Container](http://docs.google.com/java/awt/Container.html) target)

Returns the minimum dimensions needed to layout the *visible* components contained in the specified target container.

**Specified by:**[minimumLayoutSize](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**target - the container that needs to be laid out **Returns:**the minimum dimensions to lay out the subcomponents of the specified container**See Also:**[preferredLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/FlowLayout.html#preferredLayoutSize(java.awt.Container)), [Container](http://docs.google.com/java/awt/Container.html), [Container.doLayout()](http://docs.google.com/java/awt/Container.html#doLayout())

### layoutContainer

public void **layoutContainer**([Container](http://docs.google.com/java/awt/Container.html) target)

Lays out the container. This method lets each *visible* component take its preferred size by reshaping the components in the target container in order to satisfy the alignment of this FlowLayout object.

**Specified by:**[layoutContainer](http://docs.google.com/java/awt/LayoutManager.html#layoutContainer(java.awt.Container)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**target - the specified component being laid out**See Also:**[Container](http://docs.google.com/java/awt/Container.html), [Container.doLayout()](http://docs.google.com/java/awt/Container.html#doLayout())

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**()

Returns a string representation of this FlowLayout object and its values.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of this layout

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/FlowLayout.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/FileDialog.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/FocusTraversalPolicy.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/FlowLayout.html)    [**NO FRAMES**](http://docs.google.com/FlowLayout.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#lnxbz9) | [METHOD](#2jxsxqh) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).