

1. 概述

官方JavaDocsApi: [javax.swing.GroupLayout](#)

GroupLayout，分组布局管理器。它将组建按层次分组，以决定它们在 Container 中的位置。GroupLayout 主要供生成器使用（生成 并行组 和 串行组）。分组由**GroupLayout.Group**类的实例来完成，每个组可以包含任意数量的元素（Group、Component 或 Gap）。

GroupLayout支持两种组：

- 串行组 (sequential group)：按顺序沿指定方向（水平/垂直）逐个放置元素。
- 并行组 (parallel group)：沿指定方向（水平/垂直）并行排列元素，能够以四种不同方式对齐其子元素。

PS: 串行和并行根据不同的方向（水平/垂直方向），元素的排列方式也不同（从上到下 或 从左到右）。

Group的创建以及元素的添加：

```
1 // 先创建分组布局，并关联容器
2 GroupLayout layout =newGroupLayout(panel);
3 // 创建 串行组
4 GroupLayout.SequentialGroup seqGroup = layout.createSequentialGroup();
5 // 添加 组元素
6 seqGroup.addGroup(group);
7 // 添加 组件元素
8 seqGroup.addComponent(component);
9 // 添加 间隙
10 seqGroup.addGap(size);
11 // 创建 并行组
12 GroupLayout.ParallelGroup parGroup = layout.createParallelGroup();
13 // 添加 组元素（可指定对齐方式）
14 parGroup.addGroup([alignment,] group);
15 // 添加 组件元素（可指定对齐方式）
16 parGroup.addComponent(component,[alignment]);
17 // 添加 间隙
18 parGroup.addGap(size);
```

GroupLayout 对 X轴（水平方向）和 Y轴（竖直方向）单独对待，因此需要给 GroupLayout 在 **水平 和 竖直 方向上分别指定一个组**，来分别确定组件在 水平

方向和 竖直方向 上的位置。

设置 GroupLayout 两个方向上的组:

```
1 // 指定布局的水平组（确定组件在 X轴 方向上的位置）
2 GroupLayout.setHorizontalGroup(GroupLayout.Group group);
3 // 指定布局的垂直组（确定组件在 Y轴 方向上的位置）
4 GroupLayout.setVerticalGroup(GroupLayout.Group group);
```

2. 代码实例

```
1 package com.xiets.swing;
2 import javax.swing.*.*;
3 public class Main{
4     public static void main(String[] args){
5         JFrame jf =newJFrame("测试窗口");
6         jf.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
7         // 创建内容面板容器
8         JPanel panel =newJPanel();
9         // 创建分组布局，并关联容器
10        GroupLayout layout =newGroupLayout(panel);
11        // 设置容器的布局
12        panel.setLayout(layout);
13        // 创建组件
14        JButton btn01 =newJButton("Button01");
15        JButton btn02 =newJButton("Button02");
16        JButton btn03 =newJButton("Button03");
17        JButton btn04 =newJButton("Button04");
18        JButton btn05 =newJButton("Button05");
19        // 自动创建组件之间的间隙
20        layout.setAutoCreateGaps(true);
21        // 自动创建容器与触到容器边框的组件之间的间隙
22        layout.setAutoCreateContainerGaps(true);
23        /** 水平组（仅确定 X 轴方向的坐标/排列方式）
24         *
25         * 水平串行：水平排列（左右排列）
26         * 水平并行：垂直排列（上下排列）
27         */
28        // 水平并行（上下） btn01 和 btn02
29        GroupLayout.ParallelGroup hParallelGroup01 =
        layout.createParallelGroup().addComponent(btn01).addComponent(btn02);
```

```

30 // 水平并行（上下） btn03 和 btn04
31 GroupLayout.ParallelGroup hParallelGroup02 =
layout.createParallelGroup().addComponent(btn03).addComponent(btn04);
32 // 水平串行（左右） hParallelGroup01 和 hParallelGroup02
33 GroupLayout.SequentialGroup hSeqGroup =
layout.createSequentialGroup().addGroup(hParallelGroup01).addGroup(hParallelGroup
02);
34 // 水平并行（上下） hSeqGroup 和 btn05
35 GroupLayout.ParallelGroup hParallelGroup = layout.createParallelGroup().ad
dGroup(hSeqGroup).addComponent(btn05, GroupLayout.Alignment.CENTER);
36 layout.setHorizontalGroup(hParallelGroup);
37 // 指定布局的 水平组（水平坐标）
38 /** 垂直组（仅确定 Y 轴方向的坐标/排列方式）
39 *
40 * 垂直串行：垂直排列（上下排列）
41 * 垂直并行：水平排列（左右排列）
42 */
43 // 垂直并行（左右） btn01 和 btn03
44 GroupLayout.ParallelGroup vParallelGroup01 =
layout.createParallelGroup().addComponent(btn01).addComponent(btn03);
45 // 垂直并行（左右） btn02 和 btn04
46 GroupLayout.ParallelGroup vParallelGroup02 =
layout.createParallelGroup().addComponent(btn02).addComponent(btn04);
47 // 垂直串行（上下） vParallelGroup01, vParallelGroup02 和 btn05
48 GroupLayout.SequentialGroup vSeqGroup =
layout.createSequentialGroup().addGroup(vParallelGroup01).addGroup(vParallelGroup
02).addComponent(btn05);
49 layout.setVerticalGroup(vSeqGroup);
50 // 指定布局的 垂直组（垂直坐标）
51 jf.setContentPane(panel);
52 jf.pack();
53 jf.setLocationRelativeTo(null);
54 jf.setVisible(true);
55 }
56 }
57

```

结果展示:



result.png

5 个 Button 的排列说明:

沿 X轴 方向 (水平方向) :

1. Button01 和 Button02 并行 成 Group01
2. Button03 和 Button04 并行 成 Group02
3. 将 Group01 和 Group02 串行 成 Group03
4. 将 Group03 和 Butotn05 并行

沿 Y轴 方向 (竖直方向) :

1. Button01 和 Button03 并行 成 Group01
2. Button02 和 Button04 并行 成 Group02
3. 将 Group01、Group02 和 Button05 串行