

Java GridLayout

 javatpoint.com/GridLayout

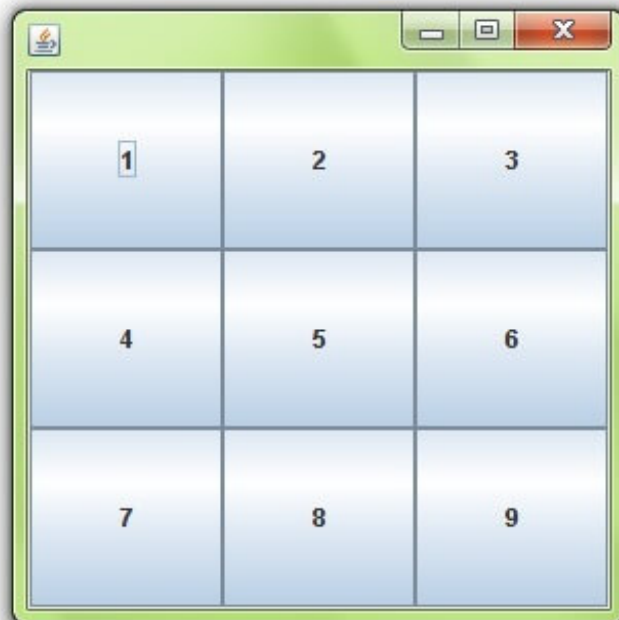
The GridLayout is used to arrange the components in rectangular grid. One component is displayed in each rectangle.

Constructors of GridLayout class

1. **GridLayout():** creates a grid layout with one column per component in a row.
2. **GridLayout(int rows, int columns):** creates a grid layout with the given rows and columns but no gaps between the components.
3. **GridLayout(int rows, int columns, int hgap, int vgap):** creates a grid layout with the given rows and columns alongwith given horizontal and vertical gaps.

Example of GridLayout class

```
1. import java.awt.*;  
2. import javax.swing.*;  
3. publicclass MyGridLayout{  
4.     JFrame f;  
5.     MyGridLayout(){  
6.         f=new JFrame();
```



```
7.         JButton b1=new JButton("1");  
8.         JButton b2=new JButton("2");  
9.         JButton b3=new JButton("3");  
10.        JButton b4=new JButton("4");  
11.        JButton b5=new JButton("5");  
12.        JButton b6=new JButton("6");  
13.        JButton b7=new JButton("7");  
14.        JButton b8=new JButton("8");  
15.        JButton b9=new JButton("9");  
16.        f.add(b1);f.add(b2);f.add(b3);f.add(b4);f.add(b5);  
17.        f.add(b6);f.add(b7);f.add(b8);f.add(b9);  
18.        f.setLayout(new GridLayout(3,3));  
19.        f.setSize(300,300);  
20.        f.setVisible(true);  
21.    }  
22.    publicstaticvoid main(String[] args) {  
23.        new MyGridLayout();  
24.    }  
25. }
```

