

# 1. 概述

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

**JButton**，按钮。

## JButton 常用构造方法:

```
1 // 创建不带有设置文本或图标的按钮
2 JButton()
3 // 创建一个带文本的按钮
4 JButton(String text)
5 // 创建一个带图标的按钮
6 JButton(Icon icon)
```

## JButton 常用方法:

```
1 // 设置按钮的 文本、字体 和 字体颜色
2 void setText(String text)
3 void setFont(Font font)
4 void setForeground(Color fg)
5 /* 以下方法定义在 javax.swing.AbstractButton 基类中 */
6 // 设置按钮是否可用
7 void setEnabled(boolean enable)
8 // 设置按钮在 默认、按下、不可用 时显示的图片
9 void setIcon(Icon defaultIcon)
10 void setPressedIcon(Icon pressedIcon)
11 void setDisabledIcon(Icon disabledIcon)
12 // 是否绘制边框（设置不绘制边框，再结合上面三个设置按钮状态图片的方法，可实现自定义按钮）
13 void setBorderPainted(boolean b);
```

## JButton 常用监听器:

```
1 // 添加 和 移除 按钮的点击事件
2 void addActionListener(ActionListener listener)
3 void removeActionListener(ActionListener listener)
```

# 2. 代码实例: 默认按钮

```
1 package com.xiets.swing;
2 import javax.swing.*;
3 import java.awt.event.ActionEvent;
```

```

4 import java.awt.event.ActionListener;
5 public class Main{
6     public static void main(String[] args){
7         JFrame jf =newJFrame("测试窗口");
8         jf.setSize(200,200);
9         jf.setLocationRelativeTo(null);
10        jf.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE); J
11        Panel panel =newJPanel();
12        // 创建一个按钮final JButton btn =newJButton("测试按钮");
13        // 添加按钮的点击事件监听器
14        btn.addActionListener(newActionListener(){
15            @Override
16            public void actionPerformed(ActionEvent e){
17                // 获取到的事件源就是按钮本身
18                // JButton btn = (JButton) e.getSource();
19                System.out.println("按钮被点击");
20            }
21        });
22        panel.add(btn);
23        jf.setContentPane(panel);
24        jf.setVisible(true);
25    }
26 }

```

结果展示:



result\_01.png

点击按钮，查看控制台输出。

### 3. 代码实例: 自定义图片按钮

使用以下两张图片分别表示按钮 *正常显示* 和 *被按下时* 显示 的图片:

```

1 package com.xiets.swing;

```

```
2 import javax.swing.*;
3 import java.awt.*;
4 import java.awt.event.ActionEvent;
5 import java.awt.event.ActionListener;
6 public class Main{
7     public static void main(String[] args)throws AWTException {
8         JFrame jf =newJFrame("测试窗口");
9         jf.setSize(200,200);
10        jf.setLocationRelativeTo(null);
11        jf.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
12        JPanel panel =newJPanel();
13        final JButton btn =newJButton();
14        // 设置按钮的默认图片
15        btn.setIcon(newImageIcon("button_normal.png"));
16        // 设置按钮被点击时的图片
17        btn.setPressedIcon(newImageIcon("button_press.png"));
18        // 不绘制边框
19        btn.setBorderPainted(false);
20        // 添加按钮点击事件监听器
21        btn.addActionListener(
22            new ActionListener(){
23                @Override
24                public void actionPerformed(ActionEvent e){
25                    System.out.println("按钮被点击了");
26                }
27            });
28        panel.add(btn);
29        jf.setContentPane(panel);
30        jf.setVisible(true);
31    }
32 }
```

结果展示:



result\_02.gif

点击按钮，查看控制台输出。