

1. 概述

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

JTextArea，文本区域。JTextArea 用来编辑多行的文本。JTextArea 除了允许多行编辑外，其他基本用法和 JTextField 基本一致。

JTextArea 常用构造方法:

```
1  /**
2   * 参数说明:
3   * text: 默认显示的文本
4   * rows: 默认可见的行数
5   * columns: 默认可见的列数
6   *
7   * 默认由 rows 和 columns 决定首选大小
8   */
9   JTextArea()
10  JTextArea(String text)
11  JTextArea(int rows,int columns)
12  JTextArea(String text,int rows,int columns)
```

JTextArea 常用方法:

```
1  // 是否自动换行，默认为 false
2  void setLineWrap(boolean wrap)
3  // 设置自动换行方式。如果为 true，则将在单词边界（空白）处换行；如果为 false，
   则将在字符边界处换行。默认为 false。
4  void setWrapStyleWord(boolean word)
5  // 获取文本框中的文本
6  String getText()
7  // 追加文本到文档末尾
8  void append(String str)
9  // 替换部分文本
10 void replaceRange(String str,int start,int end)
11 // 设置文本框的 文本、字体 和 字体颜色
12 void setText(String text)
13 void setFont(Font font)
14 void setForeground(Color fg)
15 // 获取内容的行数（以换行符计算，满行自动换下一行不算增加行数）
16 int getLineCount()
17 // 获取指定行（行数从0开始）的行尾（包括换行符）在全文中的偏移量
```

```
18  intg etLineEndOffset(int line)
19  // 获取指定偏移量所在的行数（行数从0开始）
20  int getLineOfOffset(int offset)
21  // 设置颜色，分别为：光标颜色、呈现选中部分的背景颜色、选中部分文本的颜色、不可用时文本的颜色
22  void setCaretColor(Color c)
23  void setSelectionColor(Color c)
24  void setSelectedTextColor(Color c)
25  void setDisabledTextColor(Color c)
26  // 设置文本框是否可编辑
27  void setEditable(boolean b)
28  /* 下面方法定义在 java.awt.Component 基类中 */
29  // 判断组件当前是否拥有焦点
30  boolean isFocusOwner()
31  // 设置组件是否可用
32  void setEnabled(boolean b)
```

JTextArea 复制粘贴相关方法:

```
1  // 设置光标开始位置, selectionStart >= 0
2  void setSelectionStart(int selectionStart)
3  // 设置光标结束位置, selectionEnd >= selectionStart
4  void setSelectionEnd(int selectionEnd)
5  // 复制选中部分文本
6  void copy()
7  // 剪切选中部分文本
8  void cut()
9  // 粘贴文本到文本框
10 void paste()
```

JTextArea 常用监听器:

```
1  // 添加焦点事件监听器
2  void addFocusListener(FocusListener listener)
3  // 添加文本框内的 文本改变 监听器
4  textField.getDocument().addDocumentListener(DocumentListener listener)
5  // 添加按键监听器
6  void addKeyListener(KeyListener listener)
```

PS: JTextArea 在使用时通常把它放到 JScrollPane 容器中来使用，以此来实现内容增多时可水平/垂直滚动的效果。

2. 代码实例

```
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 =new JFrame("测试窗口");
8         jf.setSize(250,250);
9         jf.setLocationRelativeTo(null);
10        jf.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
11        JPanel panel =new JPanel();
12        // 创建一个 5 行 10 列的文本区域
13        final JTextArea textArea =newJ TextArea(5,10);
14        // 设置自动换行
15        textArea.setLineWrap(true);
16        // 添加到内容面板
17        panel.add(textArea);
18        // 创建一个提交按钮，点击按钮获取输入文本
19        JButton btn =new JButton("提交");
20        btn.addActionListener(new ActionListener(){
21            @Override public void actionPerformed(ActionEvent e){
22                System.out.println("提交: "+ textArea.getText());
23            }
24        });
25        panel.add(btn);
26        jf.setContentPane(panel);
27        jf.setVisible(true);
28    }
```

结果展示:



result.png

输入内容，点击按钮，查看控制台输出。