

SWING - JTEXTAREA CLASS

http://www.tutorialspoint.com/swing/swing_jtextarea.htm

Copyright © tutorialspoint.com

Introduction

The class **JTextArea** is a multi-line area to display plain text.

Class declaration

Following is the declaration for **javax.swing.JTextArea** class –

```
public class JTextArea
    extends JTextComponent
```

Class constructors

S.N.	Constructor & Description
1	JTextArea Constructs a new TextArea.
2	JTextAreaDocumentdoc Constructs a new JTextArea with the given document model, and defaults for all of the other arguments <i>null</i> , 0, 0.
3	JTextAreaDocumentdoc, Stringtext, introws, intcolumns Constructs a new JTextArea with the specified number of rows and columns, and the given model.
4	JTextAreaintrows, intcolumns Constructs a new empty TextArea with the specified number of rows and columns.
5	JTextAreaStringtext Constructs a new TextArea with the specified text displayed.
6	JTextAreaStringtext, introws, intcolumns Constructs a new TextArea with the specified text and number of rows and columns.

Class methods

S.N.	Method & Description
1	void appendStringstr Appends the given text to the end of the document.
2	protected Document createDefaultModel

Creates the default implementation of the model to be used at construction if one isn't explicitly given.

3 **AccessibleContext getAccessibleContext**

Gets the AccessibleContext associated with this JTextArea.

4 **int getColumns**

Returns the number of columns in the TextArea.

5 **protected int getColumnWidth**

Gets column width.

6 **int getLineCount**

Determines the number of lines contained in the area.

7 **int getLineEndOffset***intline*

Determines the offset of the end of the given line.

8 **int getLineOfOffset***intoffset*

Translates an offset into the components text to a line number.

9 **int getLineStartOffset***intline*

Determines the offset of the start of the given line.

10 **boolean getLineWrap**

Gets the line-wrapping policy of the text area.

11 **Dimension getPreferredSizeScrollableViewportSize**

Returns the preferred size of the viewport if this component is embedded in a JScrollPane.

12 **Dimension getPreferredSize**

Returns the preferred size of the TextArea.

13 **protected int getRowHeight**

Defines the meaning of the height of a row.

14 **int getRows**

Returns the number of rows in the TextArea.

15 **boolean getScrollableTracksViewportWidth**

Returns true if a viewport should always force the width of this Scrollable to match the width of the viewport.

- 16 **int getScrollableUnitIncrement***RectanglevisibleRect, intorientation, intdirection*
Components that display logical rows or columns should compute the scroll increment that will completely expose one new row or column, depending on the value of orientation.
- 17 **int getTabSize**
Gets the number of characters used to expand tabs.
- 18 **String getUIClassID**
Returns the class ID for the UI.
- 19 **boolean getWrapStyleWord**
Gets the style of wrapping used if the text area is wrapping lines.
- 20 **void insert***Stringstr, intpos*
Inserts the specified text at the specified position.
- 21 **protected String paramString**
Returns a string representation of this JTextArea.
- 22 **void replaceRange***Stringstr, intstart, intend*
Replaces text from the indicated start to end position with the new text specified.
- 23 **void setColumns***intcolumns*
Sets the number of columns for this TextArea.
- 24 **void setFont***Fontf*
Sets the current font.
- 25 **void setLineWrap***booleanwrap*
Sets the line-wrapping policy of the text area.
- 26 **void setRows***introws*
Sets the number of rows for this TextArea.
- 27 **void setTabSize***intsize*
Sets the number of characters to expand tabs to.
- 28 **void setWrapStyleWord***booleanword*
Sets the style of wrapping used if the text area is wrapping lines.

Methods inherited

This class inherits methods from the following classes:

- javax.swing.text.JTextComponent
- javax.swing.JComponent
- java.awt.Container
- java.awt.Component
- java.lang.Object

JTextArea Example

Create the following java program using any editor of your choice in say **D:/ > SWING > com > tutorialspoint > gui >**

SwingControlDemo.java

```
package com.tutorialspoint.gui;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class SwingControlDemo {

    private JFrame mainFrame;
    private JLabel headerLabel;
    private JLabel statusLabel;
    private JPanel controlPanel;

    public SwingControlDemo(){
        prepareGUI();
    }

    public static void main(String[] args){
        SwingControlDemo swingControlDemo = new SwingControlDemo();
        swingControlDemo.showTextAreaDemo();
    }

    private void prepareGUI(){
        mainFrame = new JFrame("Java Swing Examples");
        mainFrame.setSize(400,400);
        mainFrame.setLayout(new GridLayout(3, 1));
        mainFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent){
                System.exit(0);
            }
        });
        headerLabel = new JLabel("", JLabel.CENTER);
        statusLabel = new JLabel("",JLabel.CENTER);

        statusLabel.setSize(350,100);

        controlPanel = new JPanel();
        controlPanel.setLayout(new FlowLayout());

        mainFrame.add(headerLabel);
        mainFrame.add(controlPanel);
        mainFrame.add(statusLabel);
        mainFrame.setVisible(true);
    }

    private void showTextAreaDemo(){
        headerLabel.setText("Control in action: JTextArea");

        JLabel commentlabel= new JLabel("Comments: ", JLabel.RIGHT);
```

```

final JTextArea commentTextArea =
    new JTextArea("This is a Swing tutorial "
        +"to make GUI application in Java.",5,20);

JScrollPane scrollPane = new JScrollPane(commentTextArea);

JButton showButton = new JButton("Show");

showButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        statusLabel.setText( commentTextArea.getText());
    }
});

controlPanel.add(commentLabel);
controlPanel.add(scrollPane);
controlPanel.add(showButton);
mainFrame.setVisible(true);
}
}

```

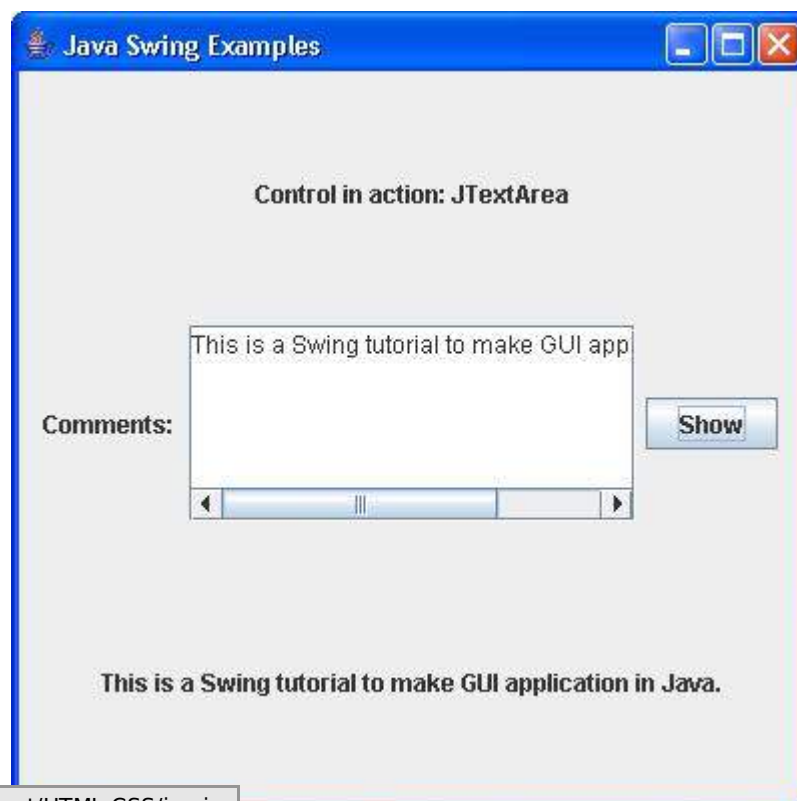
Compile the program using command prompt. Go to **D:/ > SWING** and type the following command.

```
D:\SWING>javac com\tutorialspoint\gui\SwingControlDemo.java
```

If no error comes that means compilation is successful. Run the program using following command.

```
D:\SWING>java com.tutorialspoint.gui.SwingControlDemo
```

Verify the following output



Loading [Mathjax]/jax/output/HTML-CSS/jax.js