Java KeyListener Interface

The **Java KeyListener is notified whenever you change the state of key.** It is notified against KeyEvent. The KeyListener interface is found in java.awt.event package, and it has three methods.

Interface declaration

Following is the declaration for **java.awt.event.KeyListener** interface:

public interface KeyListener **extends** EventListener

Methods of KeyListener interface

The signature of 3 methods found in KeyListener interface are given below:

Sr. no.	Method name	Description
1.	public abstract void keyPressed (KeyEvent e);	It is invoked when a key has been pressed.
2.	public abstract void keyReleased (KeyEvent e);	It is invoked when a key has been released.
3.	public abstract void keyTyped (KeyEvent e);	It is invoked when a key has been typed.

Methods inherited

This interface inherits methods from the following interface:

- java.awt.EventListener

Java KeyListener Example

In the following example, we are implementing the methods of the KeyListener interface.
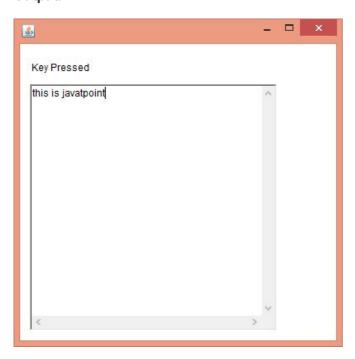
KeyListenerExample.java

```
// importing awt libraries
```

⇧ SCROLL TO TOP

```java
import java.awt.event.*;
// class which inherits Frame class and implements KeyListener interface
public class KeyListenerExample extends Frame implements KeyListener {
// creating object of Label class   and TextArea class
 Label l;
    TextArea area;
// class constructor
    KeyListenerExample() {
        // creating the label
        l = new Label();
// setting the location of the label in frame
        l.setBounds (20, 50, 100, 20);
// creating the text area
        area = new TextArea();
// setting the location of text area
        area.setBounds (20, 80, 300, 300);
// adding the KeyListener to the text area
        area.addKeyListener(this);
// adding the label and text area to the frame
        add(l);
add(area);
// setting the size, layout and visibility of frame
        setSize (400, 400);
        setLayout (null);
        setVisible (true);
    }
// overriding the keyPressed() method of KeyListener interface where we set the text of the label when key is presse
    public void keyPressed (KeyEvent e) {
        l.setText ("Key Pressed");
    }
// overriding the keyReleased() method of KeyListener interface where we set the text of the label when key is releas
    public void keyReleased (KeyEvent e) {
        l.setText ("Key Released");
    }
// overriding the keyTyped() method of KeyListener interface where we set the text of the label when a key is typed
    public void keyTyped (KeyEvent e) {
        l.setText ("Key Typed");
    }
 // main method
    public static void main(String[] args) {
        new KeyListenerExample();
    }
```

⇧ SCROLL TO TOP

Output:

Java KeyListener Example 2: Count Words & Characters

In the following example, we are printing the count of words and characters of the string. Here, the string is fetched from the TextArea and uses the KeyReleased() method of KeyListener interface.

KeyListenerExample2.java

```java
// importing the necessary libraries
import java.awt.*;
import java.awt.event.*;
// class which inherits Frame class and implements KeyListener interface
public class KeyListenerExample2 extends Frame implements KeyListener {
// object of Label and TextArea
    Label l;
    TextArea area;
// class constructor
    KeyListenerExample2() {
        // creating the label
        l = new Label();
// setting the location of label
        l.setBounds (20, 50, 200, 20);
// creating the text area
        area = new TextArea();
// setting location of text area
        area.setBounds (20, 80, 300, 300);
// adding KeyListener to the text area
        area.addKeyListener(this);
```

⇧ SCROLL TO TOP | text area to frame

```java
    add(l);
add(area);
// setting size, layout and visibility of frame
    setSize (400, 400);
    setLayout (null);
    setVisible (true);
    }
// even if we do not define the interface methods, we need to override them
    public void keyPressed(KeyEvent e) {}
// overriding the keyReleased() method of KeyListener interface
    public void keyReleased (KeyEvent e) {
// defining a string which is fetched by the getText() method of TextArea class
    String text = area.getText();
// splitting the string in words
    String words[] = text.split ("\\s");
// printing the number of words and characters of the string
    l.setText ("Words: " + words.length + " Characters:" + text.length());
    }
    public void keyTyped(KeyEvent e) {}
 // main method
    public static void main(String[] args) {
        new KeyListenerExample2();
    }
}
```

Output:

Words: 3 Characters:18

This is javatpoint

⇧ SCROLL TO TOP

Next →