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. It has three methods.

Methods of KeyListener interface

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

1. **public** **abstract** **void** keyPressed(KeyEvent e);
2. **public** **abstract** **void** keyReleased(KeyEvent e);
3. **public** **abstract** **void** keyTyped(KeyEvent e);

Java KeyListener Example

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** KeyListenerExample **extends** Frame **implements** KeyListener{

    Label l;

    TextArea area;

    KeyListenerExample(){

        l=**new** Label();

        l.setBounds(20,50,100,20);

        area=**new** TextArea();

        area.setBounds(20,80,300, 300);

        area.addKeyListener(**this**);

        add(l);add(area);

        setSize(400,400);

        setLayout(**null**);

        setVisible(**true**);

    }

**public** **void** keyPressed(KeyEvent e) {

        l.setText("Key Pressed");

    }

**public** **void** keyReleased(KeyEvent e) {

        l.setText("Key Released");

    }

**public** **void** keyTyped(KeyEvent e) {

        l.setText("Key Typed");

    }

**public** **static** **void** main(String[] args) {

**new** KeyListenerExample();

    }

}

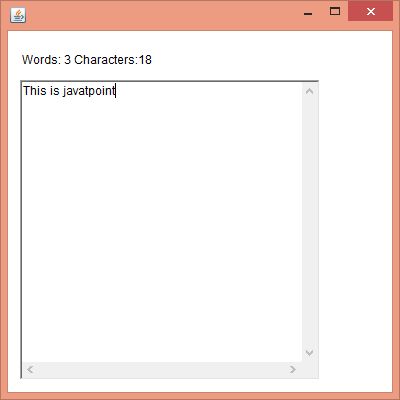
Output:



Java KeyListener Example 2: Count Words & Characters

1. **import** java.awt.\*;
2. **import** java.awt.event.\*;
3. **public** **class** KeyListenerExample **extends** Frame **implements** KeyListener{
4. Label l;
5. TextArea area;
6. KeyListenerExample(){
8. l=**new** Label();
9. l.setBounds(20,50,200,20);
10. area=**new** TextArea();
11. area.setBounds(20,80,300, 300);
12. area.addKeyListener(**this**);
14. add(l);add(area);
15. setSize(400,400);
16. setLayout(**null**);
17. setVisible(**true**);
18. }
19. **public** **void** keyPressed(KeyEvent e) {}
20. **public** **void** keyReleased(KeyEvent e) {
21. String text=area.getText();
22. String words[]=text.split("\\s");
23. l.setText("Words: "+words.length+" Characters:"+text.length());
24. }
25. **public** **void** keyTyped(KeyEvent e) {}
27. **public** **static** **void** main(String[] args) {
28. **new** KeyListenerExample();
29. }
30. }

Output



Java ActionListener Interface

The Java ActionListener is notified whenever you click on the button or menu item. It is notified against ActionEvent. The ActionListener interface is found in java.awt.event [package](https://www.javatpoint.com/package). It has only one method: actionPerformed().

actionPerformed() method

The actionPerformed() method is invoked automatically whenever you click on the registered component.

1. **public** **abstract** **void** actionPerformed(ActionEvent e);

How to write ActionListener

The common approach is to implement the ActionListener. If you implement the ActionListener class, you need to follow 3 steps:

1) Implement the ActionListener interface in the class:

1. **public** **class** ActionListenerExample Implements ActionListener

2) Register the component with the Listener:

1. component.addActionListener(instanceOfListenerclass);

3) Override the actionPerformed() method:

1. **public** **void** actionPerformed(ActionEvent e){
2. //Write the code here
3. }

Java ActionListener Example: On Button click

**import** java.awt.\*;

**import** java.awt.event.\*;

//1st step

**public** **class** ActionListenerExample **implements** ActionListener{

**public** **static** **void** main(String[] args) {

    Frame f=**new** Frame("ActionListener Example");

**final** TextField tf=**new** TextField();

    tf.setBounds(50,50, 150,20);

    Button b=**new** Button("Click Here");

    b.setBounds(50,100,60,30);

    //2nd step

    b.addActionListener(**this**);

    f.add(b);f.add(tf);

    f.setSize(400,400);

    f.setLayout(**null**);

    f.setVisible(**true**);

}

//3rd step

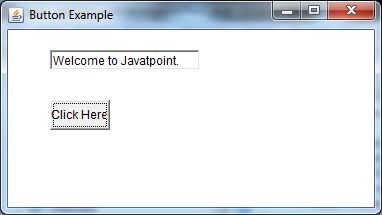
**public** **void** actionPerformed(ActionEvent e){

            tf.setText("Welcome to Javatpoint.");

}

1. }

Output:



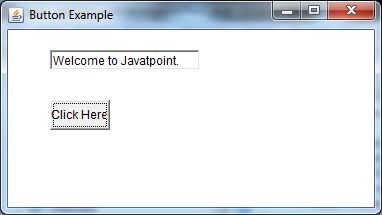
Java ActionListener Example: Using Anonymous class

We can also use the anonymous class to implement the ActionListener. It is the shorthand way, so you do not need to follow the 3 steps:

1. b.addActionListener(**new** ActionListener(){
2. **public** **void** actionPerformed(ActionEvent e){
3. tf.setText("Welcome to Javatpoint.");
4. }
5. });

Let us see the full code of ActionListener using anonymous class.

1. **import** java.awt.\*;
2. **import** java.awt.event.\*;
3. **public** **class** ActionListenerExample {
4. **public** **static** **void** main(String[] args) {
5. Frame f=**new** Frame("ActionListener Example");
6. **final** TextField tf=**new** TextField();
7. tf.setBounds(50,50, 150,20);
8. Button b=**new** Button("Click Here");
9. b.setBounds(50,100,60,30);
11. b.addActionListener(**new** ActionListener(){
12. **public** **void** actionPerformed(ActionEvent e){
13. tf.setText("Welcome to Javatpoint.");
14. }
15. });
16. f.add(b);f.add(tf);
17. f.setSize(400,400);
18. f.setLayout(**null**);
19. f.setVisible(**true**);
20. }
21. }



Java MouseListener Interface

The Java MouseListener is notified whenever you change the state of mouse. It is notified against MouseEvent. The MouseListener interface is found in java.awt.event package. It has five methods.

Methods of MouseListener interface

The signature of 5 methods found in MouseListener interface are given below:

1. **public** **abstract** **void** mouseClicked(MouseEvent e);
2. **public** **abstract** **void** mouseEntered(MouseEvent e);
3. **public** **abstract** **void** mouseExited(MouseEvent e);
4. **public** **abstract** **void** mousePressed(MouseEvent e);
5. **public** **abstract** **void** mouseReleased(MouseEvent e);

Java MouseListener Example

**import** java.awt.\*;

**import** java.awt.event.\*;

**public** **class** MouseListenerExample **extends** Frame **implements** MouseListener{

    Label l;

    MouseListenerExample(){

        addMouseListener(**this**);

        l=**new** Label();

        l.setBounds(20,50,100,20);

        add(l);

        setSize(300,300);

        setLayout(**null**);

        setVisible(**true**);

    }

**public** **void** mouseClicked(MouseEvent e) {

        l.setText("Mouse Clicked");

    }

**public** **void** mouseEntered(MouseEvent e) {

        l.setText("Mouse Entered");

    }

**public** **void** mouseExited(MouseEvent e) {

        l.setText("Mouse Exited");

    }

**public** **void** mousePressed(MouseEvent e) {

        l.setText("Mouse Pressed");

    }

**public** **void** mouseReleased(MouseEvent e) {

        l.setText("Mouse Released");

    }

**public** **static** **void** main(String[] args) {

**new** MouseListenerExample();

}

}

Output:

