

# Java Listener for applet

Topics : [JAVA](#)

Written on [April 11, 2023](#)

In Java, listeners are used to detect and respond to events such as button clicks, mouse movements, or keyboard input. In the context of applets, listeners are used to detect and respond to events that occur within the applet.

To create a listener for an applet, you need to follow these steps:

1. Create a listener class that implements the appropriate listener interface. For example, if you want to detect button clicks, you would create a class that implements the ActionListener interface.
2. Add the listener to the appropriate component in the applet. For example, if you want to detect button clicks on a button, you would add the listener to the button using the `addActionListener` method.
3. Implement the appropriate method in the listener class to respond to the event. For example, if you want to respond to a button click, you would implement the `actionPerformed` method in the ActionListener class.

Here is an example code snippet that demonstrates how to create a MouseListener in Java:

```
import java.awt.*;
import java.awt.event.*;

public class MyMouseListener implements MouseListener {
    public void mouseClicked(MouseEvent e) {
        System.out.println("Mouse clicked at (" + e.getX() + ", " + e.getY() +
        ")");
    }
    public void mouseEntered(MouseEvent e) {
        System.out.println("Mouse entered at (" + e.getX() + ", " + e.getY() +
```

```

    "));
    }
    public void mouseExited(MouseEvent e) {
        System.out.println("Mouse exited at (" + e.getX() + ", " + e.getY() +
    "));
    }
    public void mousePressed(MouseEvent e) {
        System.out.println("Mouse pressed at (" + e.getX() + ", " + e.getY() +
    "));
    }
    public void mouseReleased(MouseEvent e) {
        System.out.println("Mouse released at (" + e.getX() + ", " + e.getY() +
    "));
    }
}

```

In this example, we define a class `MyMouseListener` that implements the `MouseListener` interface. This interface defines five methods: `mouseClicked`, `mouseEntered`, `mouseExited`, `mousePressed`, and `mouseReleased`.

Each of these methods takes a `MouseEvent` object as a parameter, which contains information about the mouse event (such as the location of the mouse when the event occurred).

In the example, each of these methods simply prints a message to the console indicating the type of mouse event that occurred and the location of the mouse when the event occurred.

To use this listener in your program, you would create an instance of the `MyMouseListener` class and add it to the component that you want to listen for mouse events on, using the `addMouseListener` method.

For example:

```

Button myButton = new Button("Click me");
MyMouseListener listener = new MyMouseListener();
myButton.addMouseListener(listener);

```

This would add the `MyMouseListener` object to the `myButton` component, allowing it to respond to mouse events.

© Copyright **Aryatechno**. All Rights Reserved. Written tutorials and materials by [Aryatechno](#)