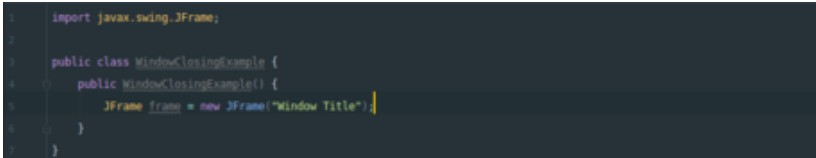


# How to Close a Window in Java

This article will show you how to close a window in Java. Closing a window is much easier using Swing's JFrame, but it's also doable using AWT's Frame. Obtain an instance of a JFrame, or create a new one. Set default close operation....

Method 1 of 2:

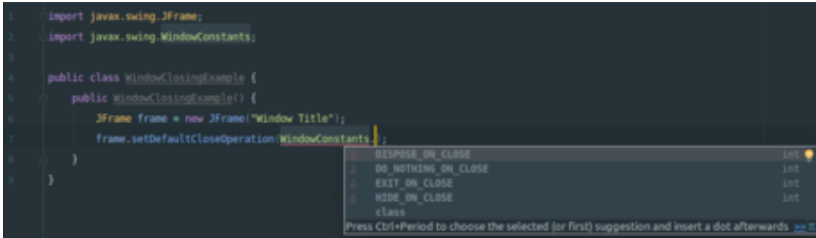
## Using javax.swing.JFrame

1. 

```
import javax.swing.JFrame;

public class WindowClosingExample {
    public WindowClosingExample() {
        JFrame frame = new JFrame("Window Title");
    }
}
```

Obtain an instance of a JFrame, or create a new one.

2. 

```
import javax.swing.JFrame;
import javax.swing.WindowConstants;

public class WindowClosingExample {
    public WindowClosingExample() {
        JFrame frame = new JFrame("Window Title");
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
    }
}
```

**Set default close operation.** Default close operation is set using the setter method inside the JFrame class setDefaultCloseOperation that determines what happens when the close button is clicked and takes the following parameters:

1. **WindowConstants.EXIT\_ON\_CLOSE** - Closes the frame and terminates the execution of the program.
2. **WindowConstants.DISPOSE\_ON\_CLOSE** - Closes the frame and does not necessarily terminate the execution of the program.
3. **WindowConstants.HIDE\_ON\_CLOSE** - Makes the frame appear like it closed by setting its visibility property to false. The difference between HIDE\_ON\_CLOSE and DISPOSE\_ON\_CLOSE is that the latter releases all of the resources used by the frame and its components.
4. **WindowConstants.DO\_NOTHING\_ON\_CLOSE** - Does nothing when the close button is pressed. Useful if you wish to, for example, display a confirmation dialog before the window is closed. You can do that by adding a WindowListener to the frame and overriding windowClosing method. Example of the custom close operation:

```

frame.setDefaultCloseOperation(WindowConstants.DO_NOTHING_ON_CLOSE
); frame.addWindowListener(new WindowAdapter() { @Override public
void windowClosing(WindowEvent e) {
// Ask for confirmation before terminating the program. int option
= JOptionPane.showConfirmDialog( frame,
"Are you sure you want to close the application?",
"Close Confirmation", JOptionPane.YES_NO_OPTION, JOptionPane.
QUESTION_MESSAGE); if (option == JOptionPane.YES_OPTION) { System.
exit(0); } } });

```

Method 2 of 2:

## Using java.awt.Frame

1.

```

1 import java.awt.Frame;
2
3 public class WindowClosingExample {
4     public WindowClosingExample() {
5         Frame frame = new Frame("Window Title");
6     }
7 }

```

**Obtain an instance of a Frame, or create a new one.**

2.

```

1 import java.awt.Frame;
2 import java.awt.event.WindowAdapter;
3
4 public class WindowClosingExample {
5     public WindowClosingExample() {
6         Frame frame = new Frame("Window Title");
7         frame.addWindowListener(new WindowAdapter() {
8             }
9     });
10 }
11 }

```

**Add window listener.** Call `addWindowListener` method on the instance. The required argument is `WindowListener`. You can either implement every method of the `WindowListener` interface or override only the methods you need from `WindowAdapter` class.

3.

```

1 import java.awt.Frame;
2 import java.awt.event.WindowAdapter;
3 import java.awt.event.WindowEvent;
4
5 public class WindowClosingExample {
6     public WindowClosingExample() {
7         Frame frame = new Frame("Window Title");
8         frame.addWindowListener(new WindowAdapter() {
9             @Override
10            public void windowClosing(WindowEvent e) {
11                // TODO: Handle click on window close button
12            }
13        });
14    }
15 }

```

**Handle window closing event.** Implement `windowClosing` method from `WindowListener` interface or override it from `WindowAdapter` class. There are two ways of closing a window:

1. Dispose the window after the close button is clicked:
  1. Call `dispose` method inside `windowClosing` method.

```
frame.addWindowListener(new WindowAdapter() { @Override public void
    windowClosing(WindowEvent e) {
// Dispose the window after the close button is clicked. dispose();
    } });
```

2. Terminate the program after the close button is clicked:

1. Call `System.exit` method inside `windowClosing` method.

```
frame.addWindowListener(new WindowAdapter() { @Override public void
    windowClosing(WindowEvent e) {
// Terminate the program after the close button is clicked. System.
    exit(0); } });
```

You finished reading the article "**How to Close a Window in Java**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.