

JFrame

A Frame is a container that can contain other components such as buttons, labels, text fields, etc. A Frame window can contain a title, a border, and also menus, text fields, buttons, and other components. An application should contain a frame so that we can add components inside it.

The Frame in Java Swing is defined in class javax.swing.JFrame. JFrame class inherits the java.awt.Frame class. JFrame is like the main window of the GUI application using swing.

Creating a Frame:

Commonly used Constructors and Methods:

JFrame class has many constructors used to create a JFrame:

1) JFrame() : This constructor is used to create a new frame that is initially invisible.

Syntax

```
JFrame objectname=new JFrame();
```

Example

```
JFrame f=new JFrame();
```

Example: Creates a frame without title and invisible.

```
import javax.swing.JFrame;
public class FrameEx
{
    public static void main(String[] args)
    {
        JFrame frame=new JFrame();           //creating instance of JFrame
    }
}
```

Note: In the above program the frame will not be visible on the screen during run time.

setVisible(Boolean value): This method belongs to the javax.swing.JFrame class. It is used to set the visibility of the component. It is by default false.

Syntax:

```
JFrameObject.setVisible(true or false);
```

Example:

```
Frame.setVisible(true);
```

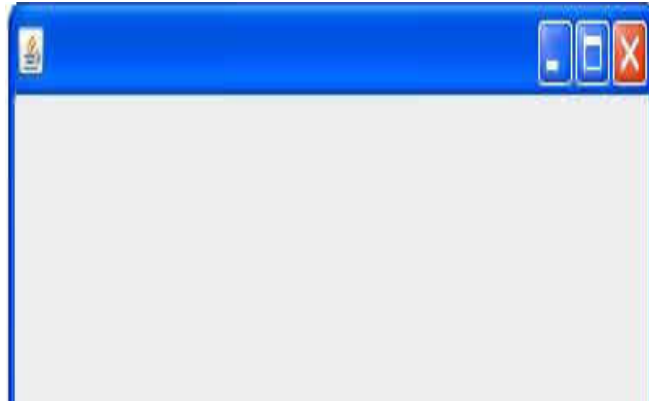
Example: Creates a frame without title and visible

```
import javax.swing.JFrame;
public class FrameEx
{
    public static void main(String[] args)
    {
        JFrame frame=new JFrame();           //creating instance of JFrame
    }
}
```

```

        frame.setVisible(true);    //making the frame visible
    }
}

```



2) JFrame(String title) : This constructor is used to create a JFrame with a title and invisible.

Syntax

```
JFrame objectname=new JFrame(title);
```

Example

```
JFrame frame=new JFrame("Welcome to Swing");
```

Example: Set title of JFrame

```

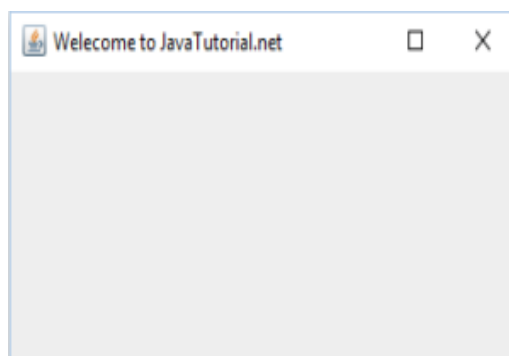
import javax.swing.JFrame;
public class FrameEx
{
    public static void main(String[] args)
    {
        JFrame frame= new JFrame("Welcome to Java");
        frame.setVisible(true);
    }
}

```

setTitle() : This method belongs to the java.swing.JFrame class. This method is used to set title in the frame's title bar.

Example

```
frame.setTitle("Welcome to Java");
```



Change window size of a JFrame

setSize() : This method used to resize the frame, it takes two parameters width and height. The frame becomes 300 pixels width and 100 pixels height.

Syntax:

```
frameobject.setSize(int width, int height);
```

Example

```
frame.setSize(500, 400 );
```

Resize a JFrame

After setting size of a JFrame you will notice you can still change it size by just simply putting the cursor at the corners and dragging it. Or if you press resize option next to close at the top right corner, it will maximize to the size of full screen. This happens because resize is set true by default. You can simply make false.

Example

```
frame.setResizable(false)
```

Now it will appear according to the dimensions you have given in code and will not resize by the graphical interface.

Change position on the screen

setLocation():

This method is used to change the position of Frame on screen. It takes two parameters x represents position along x-axis and y represents position along y-axis. The top left corner of your screen is (0, 0).

Example:

```
frame.setLocation(int x, int y);
```

Closing a JFrame

You can easily close your JFrame by clicking on the X(cross) at the top right corner of JFrame. However JFrame.setDefaultCloseOperation(int) is a method provided by JFrame class, you can set the operation that will happen when user clicks on cross. If "0" is given as a parameter, JFrame will not close even after clicking on cross.

The best practice is to use JFrame.EXIT_ON_CLOSE, it exits application (JFrame) and releases memory.

JFrame.HIDE_ON_CLOSE: It does not close JFrame, simply hides it.

JFrame.DISPOSE_ON_CLOSE: It dispose the frame off, but it keeps running and consumes memory.

JFrame.DO_NOTHING_ON_CLOSE: It does nothing when user clicks on close.

```
import javax.swing.JFrame;
public class FrameEx
{
    public static void main(String[] args)
    {
        JFrame frame= new JFrame();
        frame.setTitle("Welcome to Swing API");
        frame.setSize(600, 400);
        frame.setLocation(200, 200);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setResizable(false);
    }
}
```