Java Swing Tutorial

**Java Swing tutorial** is a part of Java Foundation Classes (JFC) that is *used to create window-based applications*. It is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java.

Unlike AWT, Java Swing provides platform-independent and lightweight components.

The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox, JMenu, JColorChooser etc.

Difference between AWT and Swing

There are many differences between java awt and swing that are given below.

|  |  |  |
| --- | --- | --- |
| **No.** | **Java AWT** | **Java Swing** |
| 1) | AWT components are **platform-dependent**. | Java swing components are **platform-independent**. |
| 2) | AWT components are **heavyweight**. | Swing components are **lightweight**. |
| 3) | AWT **doesn't support pluggable look and feel**. | Swing **supports pluggable look and feel**. |
| 4) | AWT provides **less components** than Swing. | Swing provides **more powerful components** such as tables, lists, scrollpanes, colorchooser, tabbedpane etc. |
| 5) | AWT **doesn't follows MVC**(Model View Controller) where model represents data, view represents presentation and controller acts as an interface between model and view. | Swing **follows MVC**. |

What is JFC

The Java Foundation Classes (JFC) are a set of GUI components which simplify the development of desktop applications.

Do You Know

* How to create runnable jar file in java?
* How to display image on a button in swing?
* How to change the component color by choosing a color from ColorChooser ?
* How to display the digital watch in swing tutorial ?
* How to create a notepad in swing?
* How to create puzzle game and pic puzzle game in swing ?
* How to create tic tac toe game in swing ?

### Hierarchy of Java Swing classes

The hierarchy of java swing API is given below.



### Commonly used Methods of Component class

The methods of Component class are widely used in java swing that are given below.

|  |  |
| --- | --- |
| **Method** | **Description** |
| public void add(Component c) | add a component on another component. |
| public void setSize(int width,int height) | sets size of the component. |
| public void setLayout(LayoutManager m) | sets the layout manager for the component. |
| public void setVisible(boolean b) | sets the visibility of the component. It is by default false. |

## Java Swing Examples

There are two ways to create a frame:

* By creating the object of Frame class (association)
* By extending Frame class (inheritance)

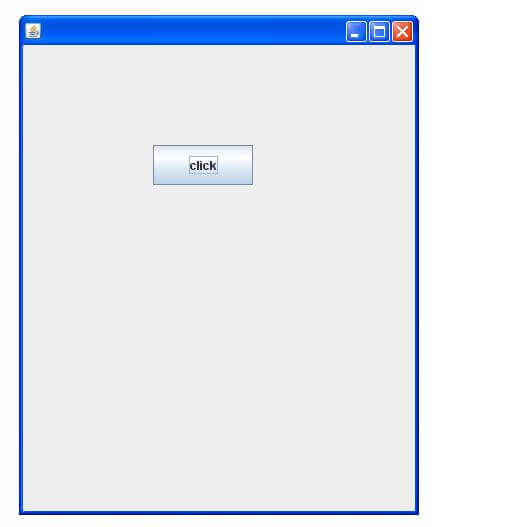
We can write the code of swing inside the main(), constructor or any other method.

### Simple Java Swing Example

Let's see a simple swing example where we are creating one button and adding it on the JFrame object inside the main() method.

*File: FirstSwingExample.java*

1. **import** javax.swing.\*;
2. **public** **class** FirstSwingExample {
3. **public** **static** **void** main(String[] args) {
4. JFrame f=**new** JFrame();//creating instance of JFrame
6. JButton b=**new** JButton("click");//creating instance of JButton
7. b.setBounds(130,100,100, 40);//x axis, y axis, width, height
9. f.add(b);//adding button in JFrame
11. f.setSize(400,500);//400 width and 500 height
12. f.setLayout(**null**);//using no layout managers
13. f.setVisible(**true**);//making the frame visible
14. }
15. }



### Example of Swing by Association inside constructor

We can also write all the codes of creating JFrame, JButton and method call inside the java constructor.

*File: Simple.java*

1. **import** javax.swing.\*;
2. **public** **class** Simple {
3. JFrame f;
4. Simple(){
5. f=**new** JFrame();//creating instance of JFrame
7. JButton b=**new** JButton("click");//creating instance of JButton
8. b.setBounds(130,100,100, 40);
10. f.add(b);//adding button in JFrame
12. f.setSize(400,500);//400 width and 500 height
13. f.setLayout(**null**);//using no layout managers
14. f.setVisible(**true**);//making the frame visible
15. }
17. **public** **static** **void** main(String[] args) {
18. **new** Simple();
19. }
20. }

The setBounds(int xaxis, int yaxis, int width, int height)is used in the above example that sets the position of the button.

### Simple example of Swing by inheritance

We can also inherit the JFrame class, so there is no need to create the instance of JFrame class explicitly.

*File: Simple2.java*

1. **import** javax.swing.\*;
2. **public** **class** Simple2 **extends** JFrame{//inheriting JFrame
3. JFrame f;
4. Simple2(){
5. JButton b=**new** JButton("click");//create button
6. b.setBounds(130,100,100, 40);
8. add(b);//adding button on frame
9. setSize(400,500);
10. setLayout(**null**);
11. setVisible(**true**);
12. }
13. **public** **static** **void** main(String[] args) {
14. **new** Simple2();
15. }}

[download this example](https://static.javatpoint.com/src/swing/first2.zip)

*What we will learn in Swing Tutorial*

* JButton class
* JRadioButton class
* JTextArea class
* JComboBox class
* JTable class
* JColorChooser class
* JProgressBar class
* JSlider class
* Digital Watch
* Graphics in swing
* Displaying image
* Edit menu code for Notepad
* OpenDialog Box
* Notepad
* Puzzle Game
* Pic Puzzle Game
* Tic Tac Toe Game
* BorderLayout
* GridLayout
* FlowLayout
* CardLayout

Next Topic[Java JButton Class](https://www.javatpoint.com/java-jbutton)

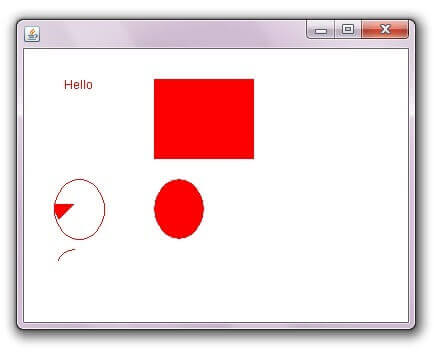
# Displaying graphics in swing:

|  |
| --- |
| java.awt.Graphics class provides many methods for graphics programming. |

## Commonly used methods of Graphics class:

|  |
| --- |
| 1. **public abstract void drawString(String str, int x, int y):** is used to draw the specified string. 2. **public void drawRect(int x, int y, int width, int height):** draws a rectangle with the specified width and height. 3. **public abstract void fillRect(int x, int y, int width, int height):** is used to fill rectangle with the default color and specified width and height. 4. **public abstract void drawOval(int x, int y, int width, int height):** is used to draw oval with the specified width and height. 5. **public abstract void fillOval(int x, int y, int width, int height):** is used to fill oval with the default color and specified width and height. 6. **public abstract void drawLine(int x1, int y1, int x2, int y2):** is used to draw line between the points(x1, y1) and (x2, y2). 7. **public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):** is used draw the specified image. 8. **public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used draw a circular or elliptical arc. 9. **public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used to fill a circular or elliptical arc. 10. **public abstract void setColor(Color c):** is used to set the graphics current color to the specified color. 11. **public abstract void setFont(Font font):** is used to set the graphics current font to the specified font. |

### Example of displaying graphics in swing:



1. **import** java.awt.\*;
2. **import** javax.swing.JFrame;
4. **public** **class** DisplayGraphics **extends** Canvas{
6. **public** **void** paint(Graphics g) {
7. g.drawString("Hello",40,40);
8. setBackground(Color.WHITE);
9. g.fillRect(130, 30,100, 80);
10. g.drawOval(30,130,50, 60);
11. setForeground(Color.RED);
12. g.fillOval(130,130,50, 60);
13. g.drawArc(30, 200, 40,50,90,60);
14. g.fillArc(30, 130, 40,50,180,40);
16. }
17. **public** **static** **void** main(String[] args) {
18. DisplayGraphics m=**new** DisplayGraphics();
19. JFrame f=**new** JFrame();
20. f.add(m);
21. f.setSize(400,400);
22. //f.setLayout(null);
23. f.setVisible(**true**);
24. }
26. }

[download this example](https://static.javatpoint.com/src/swing/swinggraphics.zip)

Next Topic[Displaying Image In Swing](https://www.javatpoint.com/Displaying-image-in-swing)