

Qual diferença entre AWT e SWING? Quando devo usar cada uma?

a biblioteca AWT usa api's dos sistema operacional onde vc compila o teu código e a SWING usa api's da sua JVM (Java Virtual Machine).

Por usar api's da JVM , sistemas desenvolvidos com SWING são multi-plataforma ou seja , vc compila em uma máquina só e pode rodar em qualquer sistema operacional , uma vez q com AWT o sistema perde essa portabilidade.

AWT é um pacote com classes que são componentes visuais (botões, frames, labels). o problema desse pacote é que ele trabalha com uma "aparência nativa". se vc faz um programa com interface gráfica de componentes AWT, no windows terá uma aparência, no linux terá outro, no mac terá outro....

ja o SWING, que tb é um pacote de classes que são componentes visuais, usa uma aparência "do java"... a aparência é igual em todos os OS... as classes do SWING herdam das classes AWT - se vc não conhece o conceito de herança em OO,

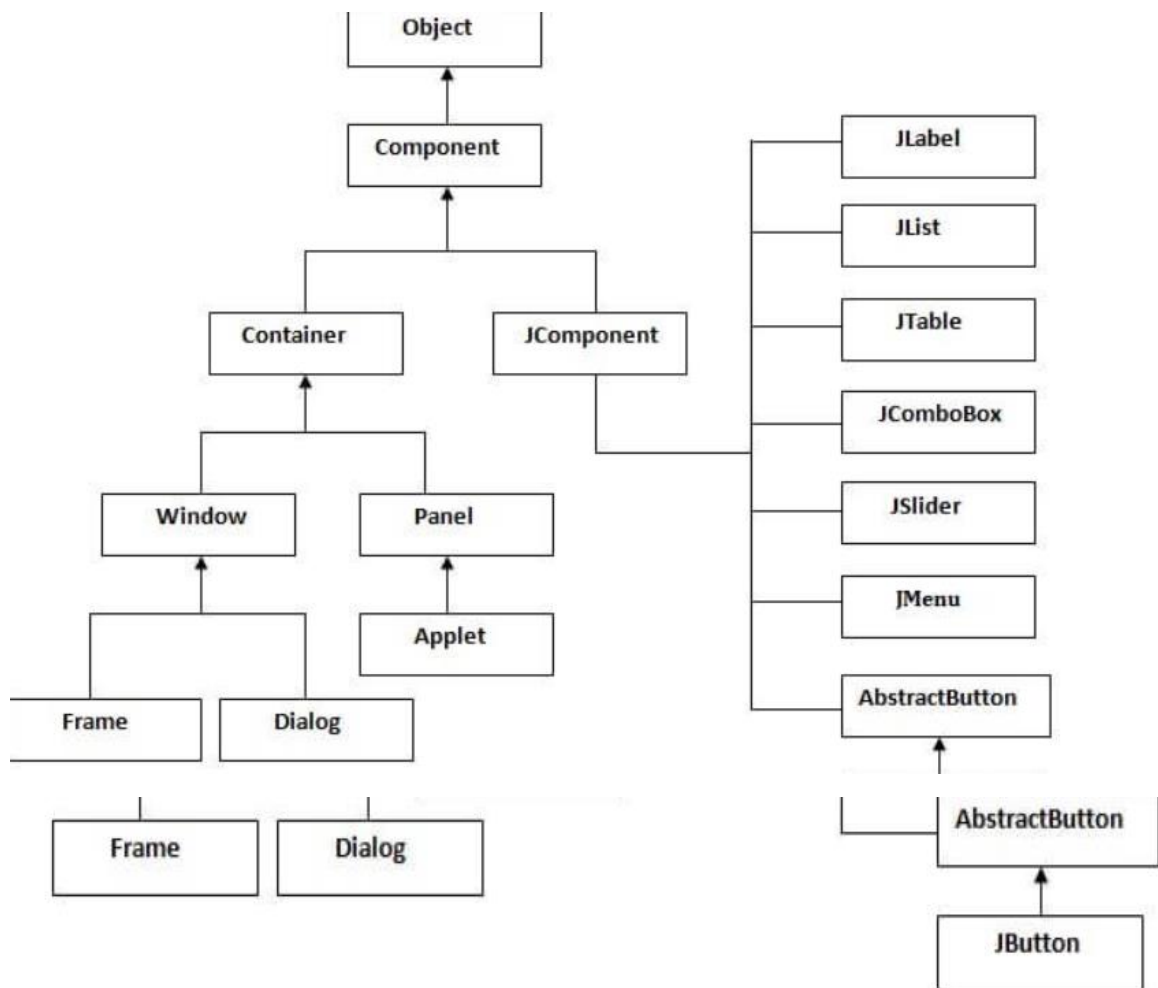
Difference between AWT and Swing

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 .

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

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
5.
6. JButton b=new JButton("click");//creating instance of JButton
7. b.setBounds(130,100,100, 40);//x axis, y axis, width, height
8.
9. f.add(b);//adding button in JFrame
10.
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. }
```



File: Simple.java

```
1. import javax.swing.*;
2. public class Simple {
3. JFrame f;
4. Simple(){
5. f=new JFrame();//creating instance of JFrame
6.
7. JButton b=new JButton("click");//creating instance of JButton
8. b.setBounds(130,100,100, 40);
9.
10. f.add(b);//adding button in JFrame
11.
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. }
16.
17. public static void main(String[] args) {
    18. new Simple();
    19. }
    20. }
```

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);
7.
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. }}
```

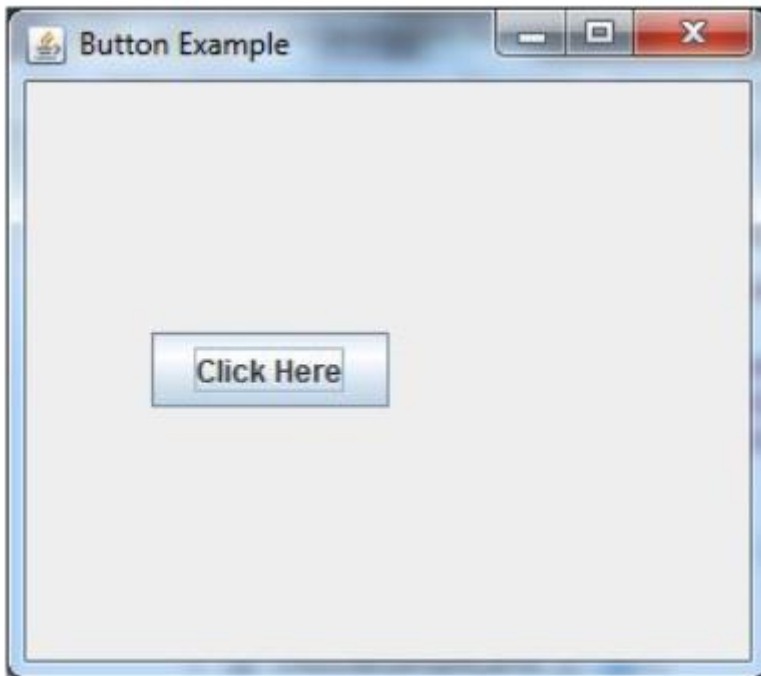
Commonly used Constructors:

Constructor	Description
JButton()	It creates a button with no text and icon.
JButton(String s)	It creates a button with the specified text.
JButton(Icon i)	It creates a button with the specified icon object.

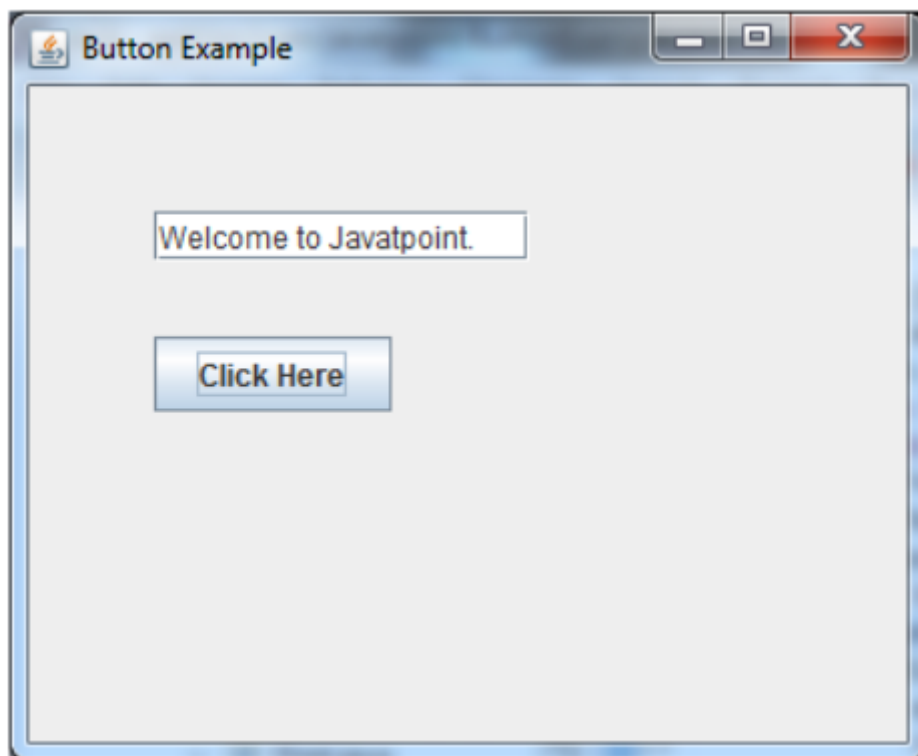
Commonly used Methods of AbstractButton class:

Methods	Description
void setText(String s)	It is used to set specified text on button
String getText()	It is used to return the text of the button.
void setEnabled(boolean b)	It is used to enable or disable the button.
void setIcon(Icon b)	It is used to set the specified Icon on the button.
Icon getIcon()	It is used to get the Icon of the button.
void setMnemonic(int a)	It is used to set the mnemonic on the button.
void addActionListener(ActionListener a)	It is used to add the action listener to this object.

```
1. import javax.swing.*;
2. public class ButtonExample {
3.     public static void main(String[] args) {
4.         JFrame f=new JFrame("Button Example");
5.         JButton b=new JButton("Click Here");
6.         b.setBounds(50,100,95,30);
7.         f.add(b);
8.         f.setSize(400,400);
9.         f.setLayout(null);
10.        f.setVisible(true);
11.    }
12. }
```




```
1. import java.awt.event.*;
2. import javax.swing.*;
3. public class ButtonExample {
4.     public static void main(String[] args) {
5.         JFrame f=new JFrame("Button Example");
6.         final JTextField tf=new JTextField();
7.         tf.setBounds(50,50, 150,20);
8.         JButton b=new JButton("Click Here");
9.         b.setBounds(50,100,95,30);
10.        b.addActionListener(new ActionListener(){
11.            public void actionPerformed(ActionEvent e){
12.                tf.setText("Welcome to Javatpoint.");
13.            }
14.        });
15.        f.add(b);f.add(tf);
16.        f.setSize(400,400);
17.        f.setLayout(null);
18.        f.setVisible(true);
19.    }
20. }
```



```
1. import javax.swing.*;
2. class LabelExample
3. {
4.     public static void main(String args[])
5.     {
6.         JFrame f= new JFrame("Label Example");
7.         JLabel l1,l2;
8.         l1=new JLabel("First Label.");
9.         l1.setBounds(50,50, 100,30);
10.        l2=new JLabel("Second Label.");
11.        l2.setBounds(50,100, 100,30);
12.        f.add(l1); f.add(l2);
13.        f.setSize(300,300);
14.        f.setLayout(null);
15.        f.setVisible(true);
16.    }
17. }
```



```
1. import java.awt.event.*;
2. public class LabelExample extends Frame implements ActionListener{
3.     JTextField tf; JLabel l; JButton b;
4.     LabelExample(){
5.         tf=new JTextField();
6.         tf.setBounds(50,50, 150,20);
7.         l=new JLabel();
8.         l.setBounds(50,100, 250,20);
9.         b=new JButton("Find IP");
10.        b.setBounds(50,150,95,30);
11.        b.addActionListener(this);
12.        add(b);add(tf);add(l);
13.        setSize(400,400);
14.        setLayout(null);
15.        setVisible(true);
16.    }
17.    public void actionPerformed(ActionEvent e) {
18.        try{
19.            String host=tf.getText();
20.            String ip=java.net.InetAddress.getByName(host).getHostAddress();
21.            l.setText("IP of "+host+" is: "+ip);
22.        }catch(Exception ex){System.out.println(ex);}
23.    }
24.    public static void main(String[] args) {
25.        new LabelExample();
26.    } }
```

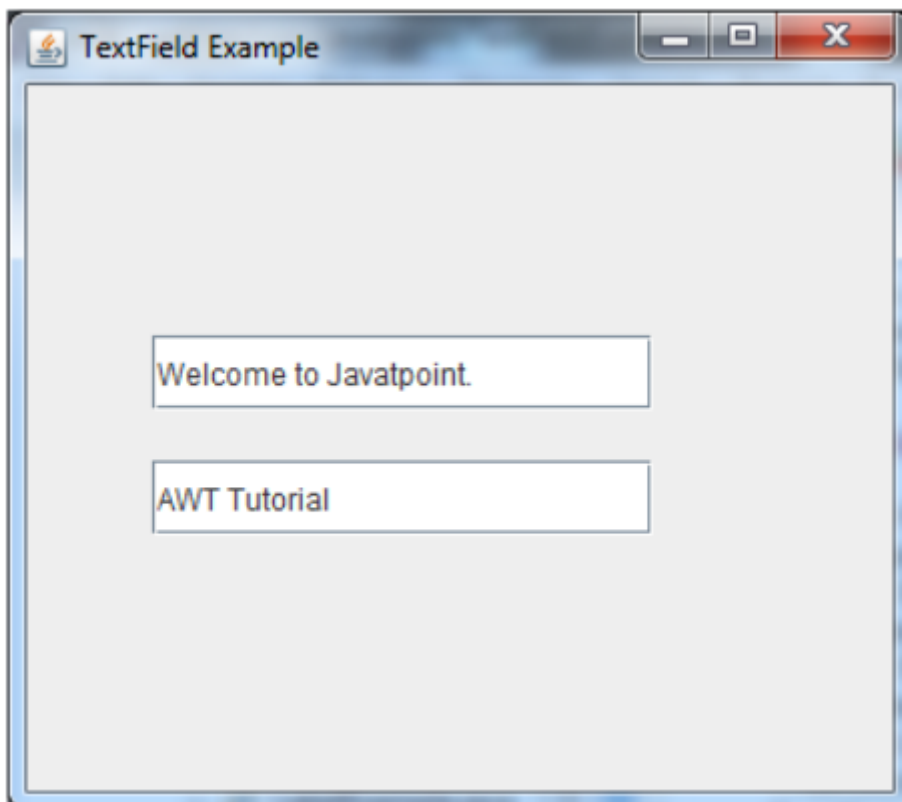


www.javatpoint.com

IP of www.javatpoint.com is: 144.76.11.18

Find IP

```
1. import javax.swing.*;
2. class TextFieldExample
3. {
4.     public static void main(String args[])
5.     {
6.         JFrame f= new JFrame("TextField Example");
7.         JTextField t1,t2;
8.         t1=new JTextField("Welcome to Javatpoint.");
9.         t1.setBounds(50,100, 200,30);
10.        t2=new JTextField("AWT Tutorial");
11.        t2.setBounds(50,150, 200,30);
12.        f.add(t1); f.add(t2);
13.        f.setSize(400,400);
14.        f.setLayout(null);
15.        f.setVisible(true);
16.    }
17. }
```



```
1. import javax.swing.*;
2. public class TextAreaExample
3. {
4.     TextAreaExample(){
5.         JFrame f= new JFrame();
6.         JTextArea area=new JTextArea("Welcome to javatpoint");
7.         area.setBounds(10,30, 200,200);
8.         f.add(area);
9.         f.setSize(300,300);
10.        f.setLayout(null);
11.        f.setVisible(true);
12.    }
13. public static void main(String args[])
14. {
15.     new TextAreaExample();
16. }}
```

