

# Part

12

Java Programming Language  
Mr.Rungrote Phonkam  
rungrote@it.kmitl.ac.th



# Contents

1. Java Program
2. Applet
3. HTML Applet Tag
4. JFC
5. Swing Components



# 1. Java Program

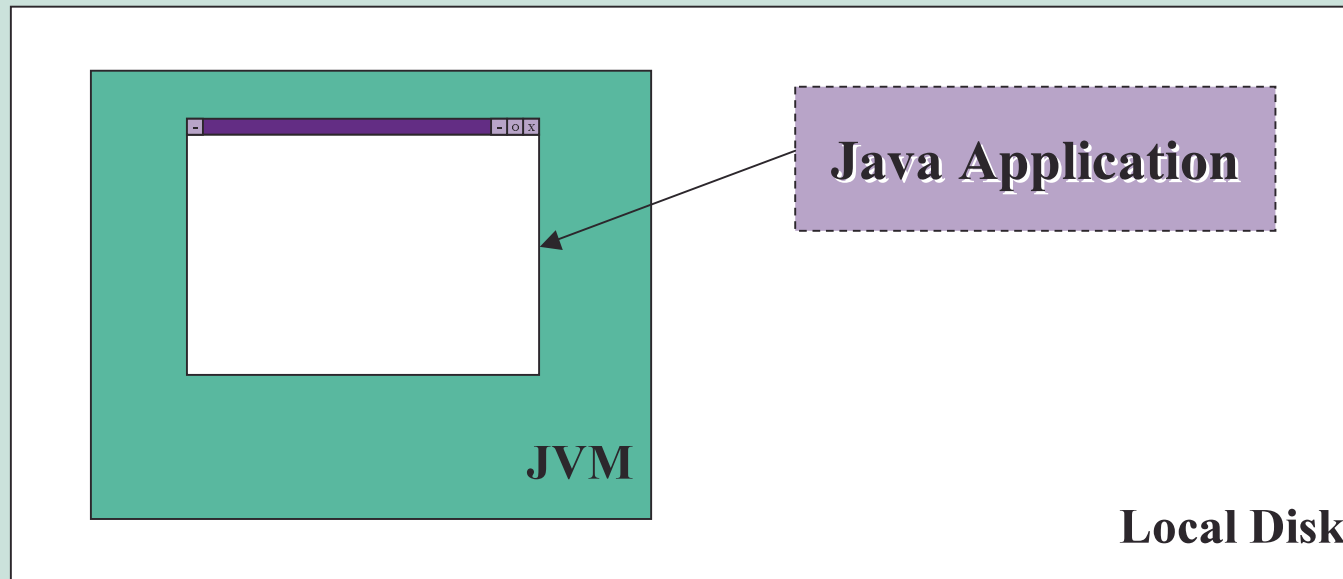
## Java Programs

- Java Application คือ โปรแกรมที่ทำงานในแบบ Stand-alone โดยการเรียกใช้ผ่านทางระบบปฏิบัติการ
- Java Applet คือ โปรแกรมที่ทำงานร่วมกับเว็บเพจ โดยการเรียกใช้ผ่านบราวเซอร์ (Web Client หรือ Web Browser)
- Java Servlet คือ โปรแกรมที่ทำงานอยู่บนเว็บเซิร์ฟเวอร์ เพื่อทำงานโดยส่งผลลัพธ์กลับไปยังผู้เรียกใช้ คือบราวเซอร์



# 1. Java Program

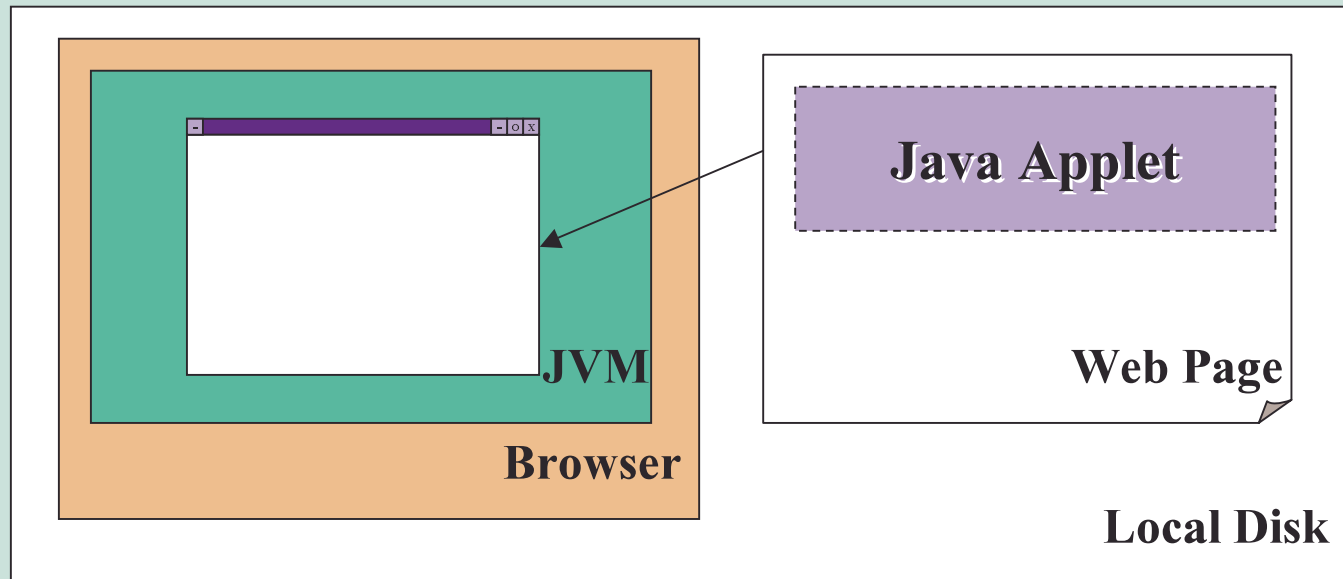
## Java Application





# 1. Java Program

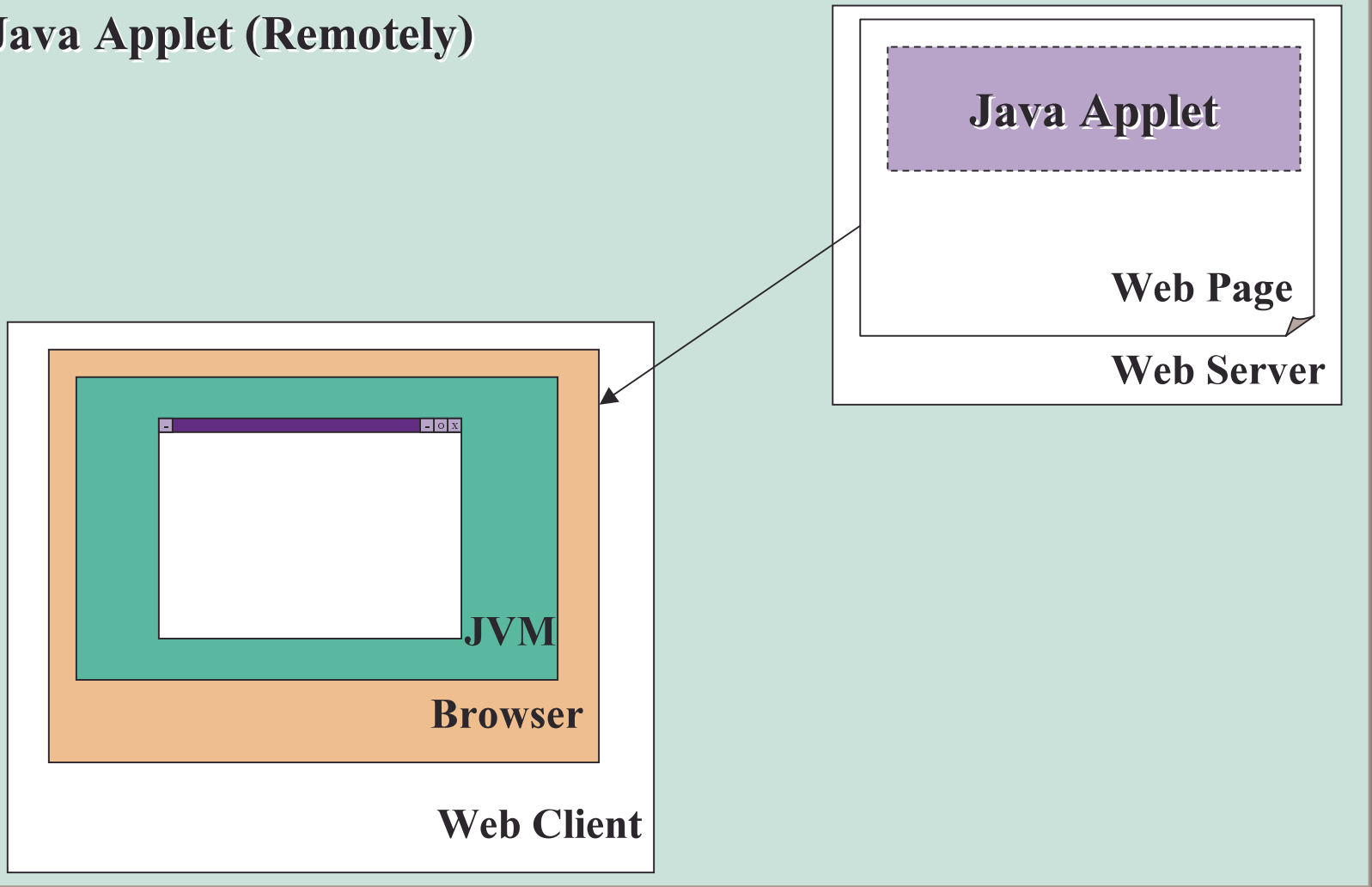
## Java Applet (Locally)





# 1. Java Program

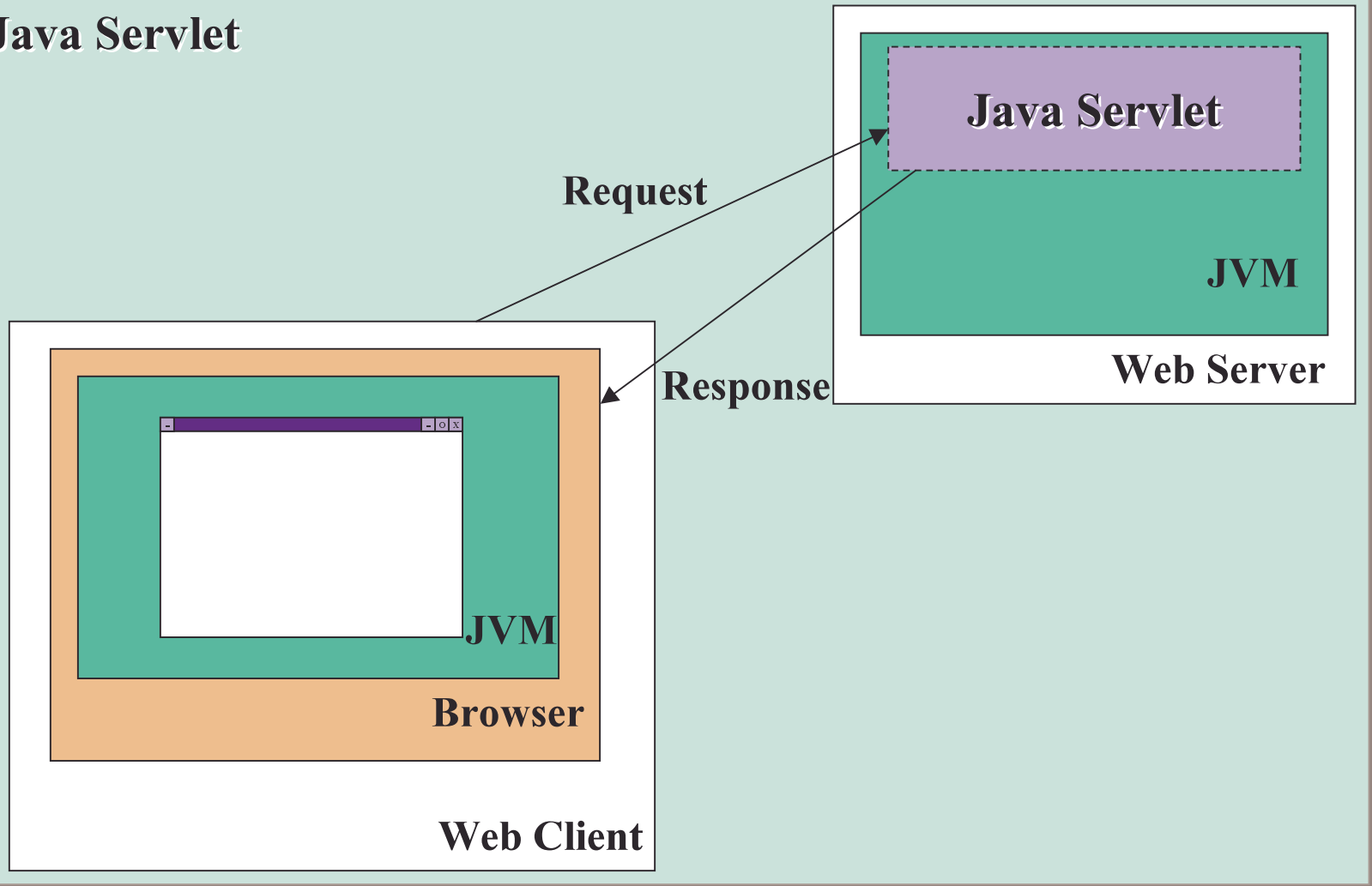
## Java Applet (Remotely)





# 1. Java Program

**Java Servlet**





## 2. Applet

### แอปเพล็ต (Applet)

- โปรแกรมที่สามารถใช้งานได้ในทุกแพลตฟอร์ม ทุกเวอร์ชันของ OS
- ไม่จำเป็นต้องมีการติดตั้งโปรแกรมก่อนการใช้งาน เพียงแค่การเรียกใช้ด้วยบราวเซอร์
- ทำให้รูปแบบที่ได้เปลี่ยนจาก Static ไปเป็น Dynamic





# 2. Applet

## รูปแบบการพัฒนา Applet

```
class myApplet extends Applet  
....  
.....
```

**Step 1:**  
สร้างซอร์สโค้ด  
รูปแบบ Applet

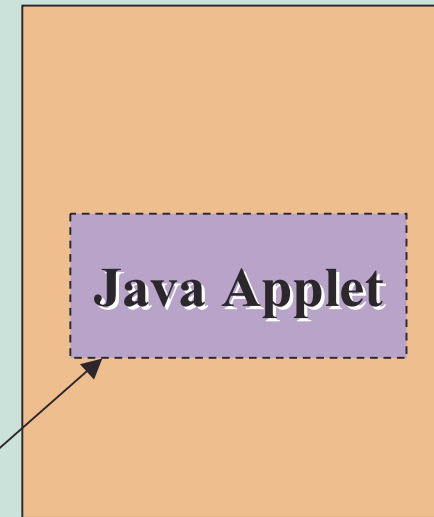
**Step 2:**  
คอมไพล์ให้ได้  
Bytecodes

myApplet.class

```
<APPLET code=myApplet.class>  
</APPLET>
```

**Step 3:**  
สร้างเว็บเพจ ที่มีคำสั่งใช้งาน Applet

**Step 4:**  
ใช้งาน Applet ด้วยบราวเซอร์



Browser



## 2. Applet

### โครงสร้างของซอร์สโค้ด Applet

```
Import java.applet.*;  
  
public class AppletName extends Applet {  
    public void init() { . . . }  
    public void start() { . . . }  
    public void stop() { . . . }  
    public void paint(Graphics G) { . . . }  
}
```



## 2. Applet

### โครงสร้างของซอร์สโค้ด Applet

Method	Description
init( )	ถูกเรียกเมื่อ Applet ถูกโหลดเข้าสู่ JVM และถูกเรียกเพียงครั้งเดียว
start( )	ถูกเรียกทุกครั้งเมื่อเพจที่มี Applet ถูกแสดงเป็นเพจหลัก
stop( )	ถูกเรียกทุกครั้งเมื่อเพจที่มี Applet ไม่ได้ถูกแสดงเป็นเพจหลัก
paint(Graphics)	ถูกเรียกใช้เมื่อพื้นที่ Applet ต้องการถูกวาดใหม่



## 2. Applet

```
import java.applet.*;
import java.awt.*;

public class Message extends Applet {
    public void paint(Graphics g) {
        g.drawString("Yes!", 25, 35);
    }
}
```

Message.java

```
<applet code=Message.class width=200 height=100>
</applet>
```

Message.html



## 2. Applet

### การทดสอบการเขียน Applet

Step 1: สร้างซอร์สโค้ด Message.java

Step 2: คอมไพล์ซอร์สโค้ด Message.java ได้ผลลัพธ์ Message.class

Step 3: สร้างเว็บเพจ Message.html ไว้ที่เดียวกับไฟล์ Message.class

Step 4: ทดลองใช้งาน โดยการเปิดไฟล์ Message.html ที่บราวเซอร์

- หรือ -

เรียกโปรแกรม AppletViewer Message.html



## 2. Applet

```
// <applet code=Message.class width=200 height=100>
// </applet>

import java.applet.*;
import java.awt.*;

public class Message extends Applet {
    public void paint(Graphics g) {
        g.drawString("Yes!", 25, 35);
    }
}
```

Message.java



## 2. Applet

### การทดสอบการเขียน Applet

Step 1: สร้างซอร์สโค้ด **Message.java**

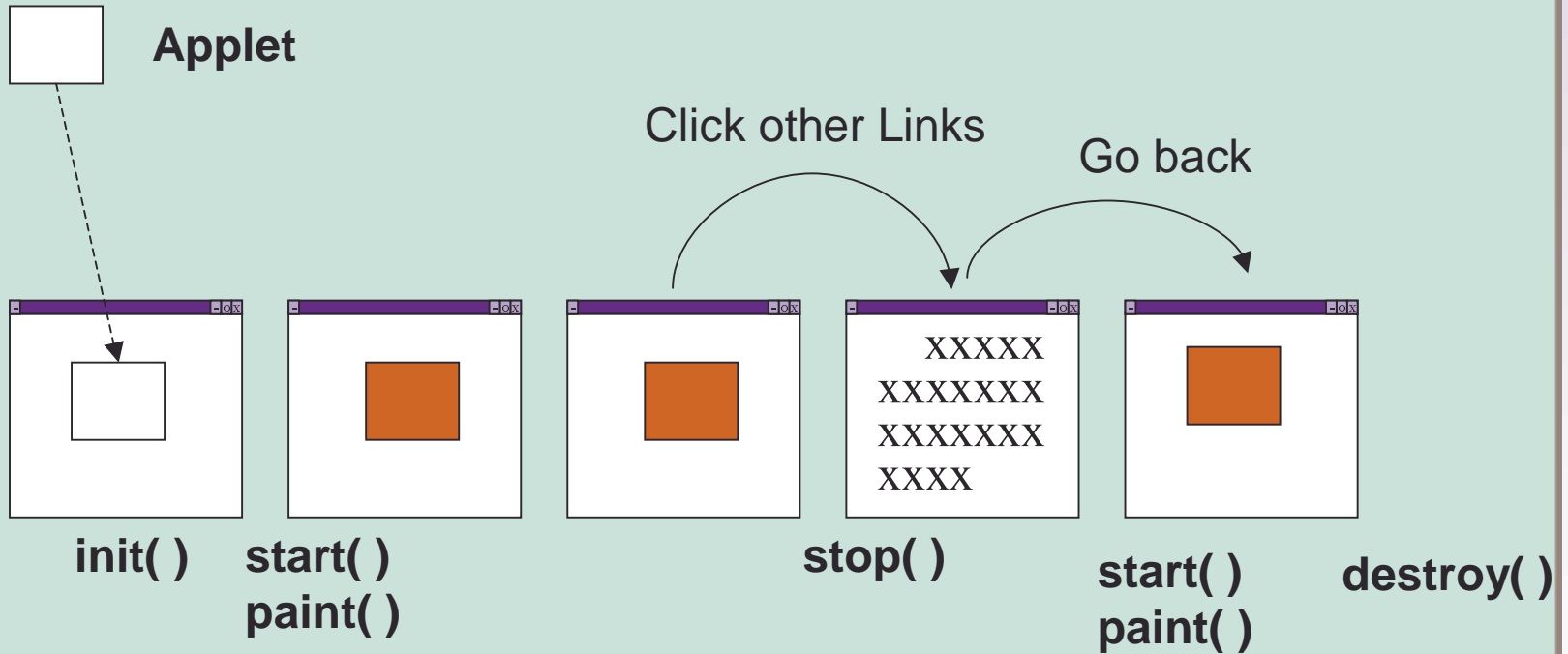
Step 2: คอมไพล์ซอร์สโค้ด Message.java ได้ผลลัพธ์ Message.class

Step4: เรียกโปรแกรม AppletViewer **Message.java**



# 2. Applet

ขั้นตอนการทำงานของเมธอดใน Applet







## 2. Applet

```
import java.applet.Applet;
import java.awt.Graphics;
import java.awt.Color;
public class SimpleApplet extends Applet
{   public void init( )
    {   setBackground(Color.blue);   }
    public void start( )
    {   System.out.println("Starting...");   }
    public void stop( )
    {   System.out.println("Stopping...");   }
    public void destrot( )
    {   System.out.println("Preparing to upload...");   }
    public void paint(Graphics g)
    {   System.out.println("Paint");
        g.setColor(Color.red);
        g.drawRect(0, 0, getSize().width -1, getSize().height -1);
        g.setColor(Color.green);
        g.drawString("Hello Java Applet", 12, 25);
    }
}
```



## 2. Applet

```
<HTML>  
<BODY>  
<APPLET CODE=SimpleApplet.class  
  WIDTH=200 HEIGHT=100>  
</APPLET>  
</BODY>  
</HTML>
```



# 2. Applet

```
C:\WINNT\System32\  
Font specified in  
Font specified in  
Font specified in  
Font specified in  
Font specified in  
Starting...  
Paint  
Stopping...  
Starting...  
Paint  
Stopping...  
Paint  
Starting...  
Paint  
Paint  
Paint  
Paint  
Paint  
Stopping...
```

The screenshot shows the 'Applet Viewer: Simpl...' window. The main area displays 'Hello Java Applet' in green text on a purple background. Below this, a status bar reads 'Applet started.'. To the left of the viewer is a control menu with the following options: Restart, Reload, Stop, Save..., Start, Clone..., Tag..., Info..., Edit, Character Encoding, Print..., Properties..., Close, and Quit. An orange arrow points from the 'Applet' label in the viewer's title bar to the 'Applet started.' status bar.



## 3. HTML Applet Tag

```
<APPLET      code=ClassName
              width=WidthPixel      height=HeithPixel
              [ archive=ArchiveFile ]
              [ codebase=URL ]
              [ vspace=vPixel ] [hspace=hPixel ]
              [ aligh=Alignment
              [ name=AppletName ]
              [ alt="Some Text" ]
              >
<param name=ParamName      value=ParamValue>
...
<param name=ParamName      value=ParamValue>
</APPLET>
```



## 3. HTML Applet Tag

ClassFileName	คือคลาสไฟล์ที่ถูกเขียนไว้ในแบบ Applet
WidthPixel, HeightPixel	คือความกว้างและความสูงของ Applet
ArchiveFile	คือไฟล์ที่ส่ง Applet มาพร้อมกับไฟล์อื่นๆรวมกัน
URL	คือ URL ของตำแหน่งที่มาของ Applet
vPixel, hPixel	คือระยะขอบของ Applet ในแนวตั้งและแนวนอน
Alignment	การจัดวาง Applet ซึ่งเป็นไปได้ทั้ง left, right, top, middle
AppletName	คือชื่อเรียก Applet
alt="Some Text"	เป็นการแสดงข้อความในกรณีที่เบราว์เซอร์ ไม่สนับสนุนการใช้งาน Applet
paramname	คือชื่อพารามิเตอร์
paramvalue	คือข้อมูลของพารามิเตอร์



# 3. HTML Applet Tag

## Parameters

```
<Applet ..... >  
<param Name=myParam Value="MyValue">  
</Applet>
```

วิธีการดึงค่า Parameter จากแท็ก Applet

```
class .....  
init( ) {  
    String s = getParameter("myParam");  
    System.out.println(s);  
}  
.....
```



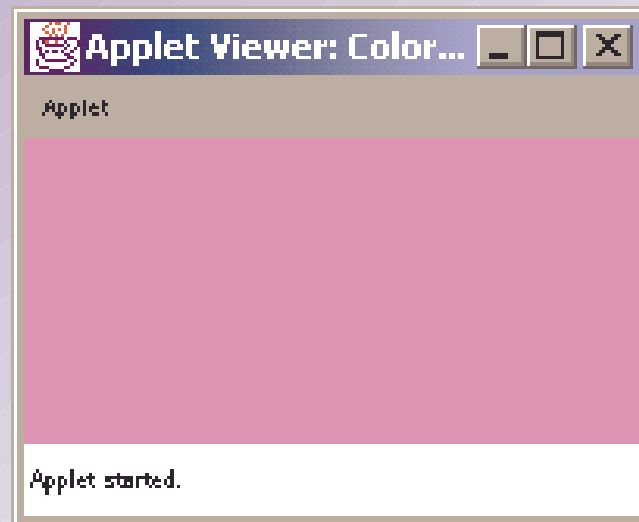
## 3. HTML Applet Tag

```
import java.applet.*;
import java.awt.*;
public class Colorize extends Applet {
    private int red=0, green=0, blue=0;
    public void init() {
        String s = getParameter("red");
        if (s != null)    red = Integer.parseInt(s);
        s = getParameter("green ");
        if (s != null)    green = Integer.parseInt(s);
        s = getParameter("blue ");
        if (s != null)    blue = Integer.parseInt(s);
        setBackground(new Color(red, green, blue));
    }
}
```



# 3. HTML Applet Tag

```
<applet code="Colorize.class" width=200 height=200>  
<param name=red value="255">  
<param name=green value="128">  
<param name=blue value="255">  
</applet>
```



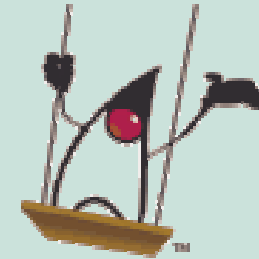




# 4. JFC

## Java Foundation Classes(JFC)

- ช่วยในการสร้างส่วน GUI ของโปรแกรม
- แบ่งส่วนประกอบย่อยได้ 5 ส่วนคือ
  - Swing Components ขึ้นส่วนทางด้านกราฟฟิก
  - Plugable Look and Feel ส่วนสนับสนุนการปรับแต่งรูปแบบ และมุมมอง
  - Accessibility API สนับสนุนการเขียนโปรแกรมสำหรับผู้ที่มีปัญหาทางด้านร่างกาย
  - Java 2D สนับสนุนการสร้างงานทั้งด้านภาพ และตัวอักษรในแบบ 2D
  - Drag & Drop สนับสนุนลักษณะการใช้งานโปรแกรมในแบบลากแล้วปล่อย





# 4. JFC

## Java Foundation Classes(JFC)

- JFC 1.1 เป็น Extension API ใน JDK 1.1
- แต่ใน JDK 1.2 และ 1.3 มี JFC เป็น Core API
- แพคเกจที่ใช้ใน JFC มี 15 ตัวคือ

`javax.swing.border,`

`javax.accessibility,`

`javax.swing.colorchooser,`

`javax.swing.filechooser,`

`javax.swing.plaf.basic,`

`javax.swing.plaf.multi,`

`javax.swing.text,`

`javax.swing.tree,`

`javax.swing,`

`javax.swing.event,`

`javax.swing.plaf,`

`javax.swing.plaf.metal,`

`javax.swing.table,`

`javax.swing.text.html,`

`javax.swing.undo.`

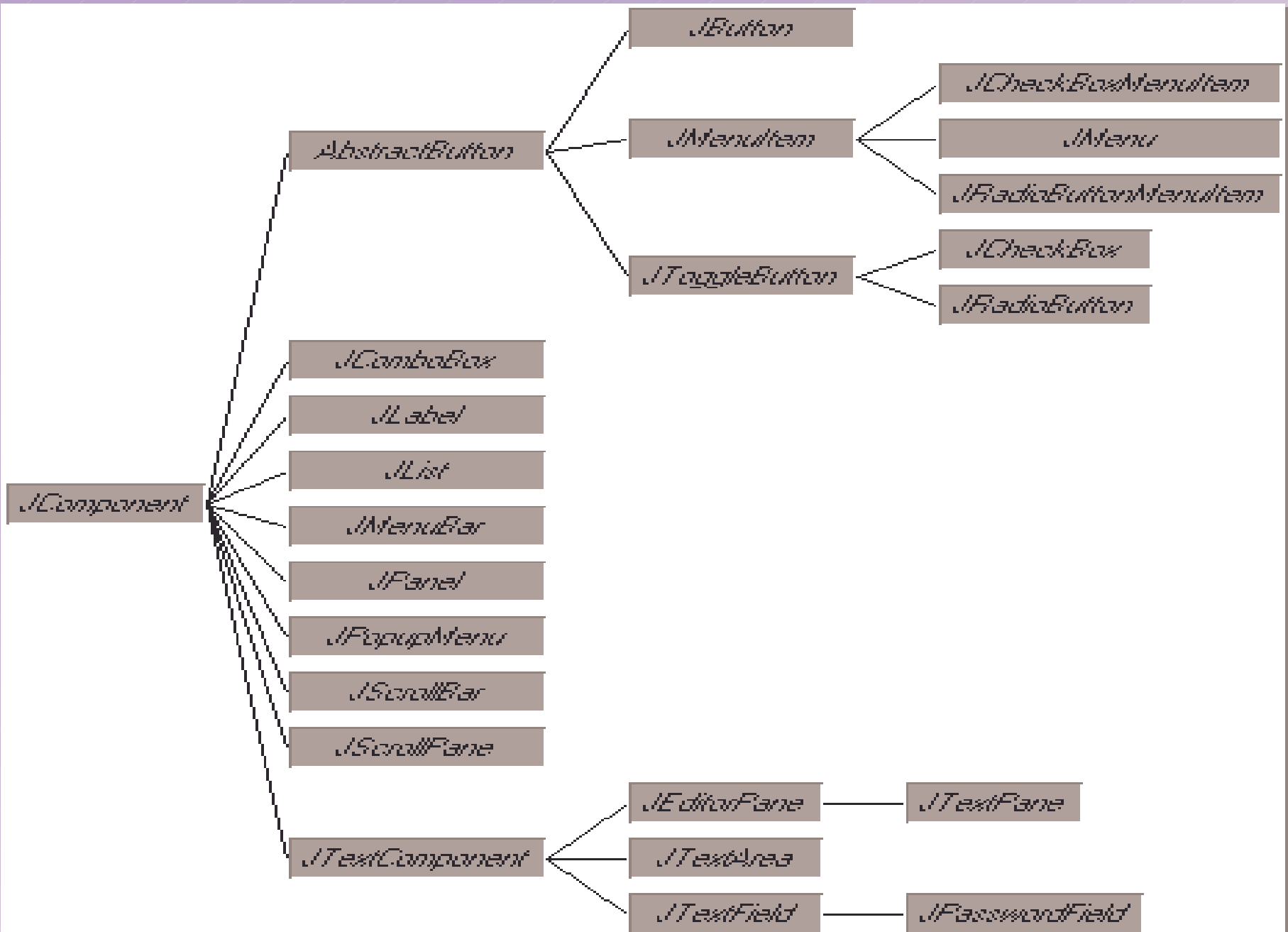
- Swing เป็นแบบ lightweight ขณะที่ AWT เป็น Heavyweight



# 5 Swing

## สวิงซ์ (Swing)

- การกำหนดส่วนติดต่อกับผู้ใช้ด้วยการแสดง Component เช่น TextBox, Menu, Icon, ScroolBar เป็นต้น
- การวาง Component สามารถวางบนพื้นที่ Container ซึ่งถูกจัดการโดย Layout Manger
- Component ถูกใช้งานโดยผู้ใช้โปรแกรมหมายถึงมี Event เกิดขึ้นกับ Component และการตอบสนองทำโดยส่วน Listener และ Adapter





# 5.1 Swing Component

```
import java.awt.*;
import java.awt.event.*;
public class TestSwing extends Frame {
    public static int WIDTH = 250;
    public static int HEIGHT = 130;
    public TestSwing(String lab) {
        super(lab); }
    public static void main(String s[]) {
        TestSwing frame = new TestSwing("Test Swing Application");
        frame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e)
                { System.exit(0); } });
        frame.setSize(WIDTH, HEIGHT);
        frame.add(new LabelPanel);
        frame.show();
    }
}
```



## 5.2 Swing Component(JLabel)

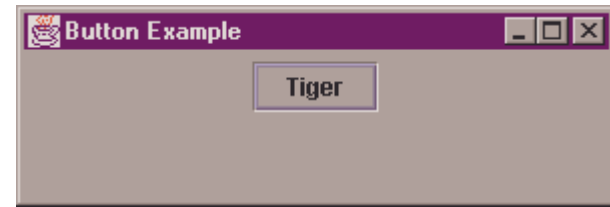
```
public class LabelPanel extends JPanel {
    public LabelPanel() {
        // Create and add a JLabel
        JLabel plainLabel = new JLabel("Plain Small Label"); add(plainLabel);
        // Create a 2nd JLabel
        JLabel fancyLabel = new JLabel("Fancy Big Label");
        // Instantiate a Font object to use for the label
        Font fancyFont = new Font("Serif", Font.BOLD | Font.ITALIC, 32);
        // Associate the font with the label
        fancyLabel.setFont(fancyFont);
        // Create an Icon
        Icon tigerIcon = new ImageIcon("SmallTiger.gif");
        // Place the Icon in the label
        fancyLabel.setIcon(tigerIcon);
        // Align the text to the right of the Icon
        fancyLabel.setHorizontalAlignment(JLabel.RIGHT);
        // Add to panel
        add(fancyLabel);
    }
}
```





## 5. 3 Swing Component(JButton)

```
public class ButtonPanel extends JPanel {  
    public ButtonPanel () {  
        JButton myButton = new JButton("Tiger");  
        add(myButton);  
    }  
}
```



```
public class ButtonPanel extends JPanel {  
    public ButtonPanel() {  
        Icon tigerIcon = new ImageIcon("SmallTiger.gif");  
        JButton myButton = new JButton("Tiger", tigerIcon);  
        add(myButton);  
    }  
}
```





## .4 Swing Component(JCheckBox)

```
class ToggleIcon implements Icon {
    boolean state;
    public ToggleIcon (boolean s) {    state = s;    }
    public void paintIcon (Component c, Graphics g, int x, int y) {
        int width = getIconWidth();
        int height = getIconHeight();
        g.setColor (Color.black);
        if (state)
            g.fillRect (x, y, width, height);
        else
            g.drawRect (x, y, width, height);
    }
    public int getIconWidth() {    return 10;    }
    public int getIconHeight() {    return 10;    }
}
```

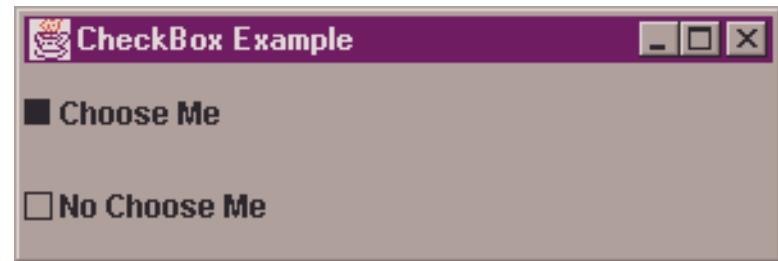
มีต่อ





## 5.4 Swing Component(JCheckBox)

```
public class CheckboxPanel extends JPanel {  
    Icon unchecked = new ToggleIcon (false);  
    Icon checked = new ToggleIcon (true);  
    public CheckboxPanel() {  
        // Set the layout for the JPanel  
        setLayout(new GridLayout(2, 1));  
        // Create checkbox with its state initialized to true  
        JCheckBox cb1 = new JCheckBox("Choose Me", true);  
        cb1.setIcon(unchecked);  
        cb1.setSelectedIcon(checked);  
        // Create checkbox with its state initialized to false  
        JCheckBox cb2 = new JCheckBox(  
            "No Choose Me", false);  
        cb2.setIcon(unchecked);  
        cb2.setSelectedIcon(checked);  
        add(cb1);    add(cb2);  
    }  
}
```





## 5.5 Swing Component(JRadioButton)

```
public class RadioButtonPanel extends JPanel {
    public RadioButtonPanel() {
        // Set the layout to a GridLayout
        setLayout(new GridLayout(4,1));
        // Declare a radio button
        JRadioButton radioButton;
        // Instantiate a ButtonGroup for functional
        // association among radio buttons
        ButtonGroup rbg = new ButtonGroup();

        // Create a label for the group
        JLabel label = new JLabel("Annual Salary: ");
        label.setFont(new Font("SansSerif", Font.BOLD, 14));
        add(label);

        // Add a new radio button to the pane
        radioButton = new JRadioButton("$45,000");
        add (radioButton);
        // set key accelerator
        radioButton.setMnemonic (KeyEvent.VK_4);
```

มีต่อ

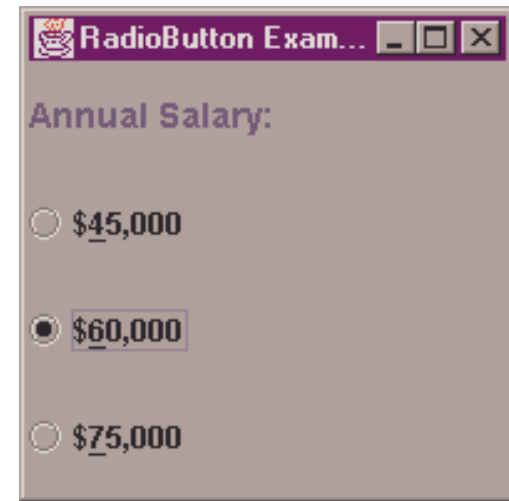


## 5.5 Swing Component(JRadioButton)

```
// Add the button to the ButtonGroup  
rbg.add (radioButton);
```

```
// Set this radio button to be the default  
radioButton.setSelected(true);
```

```
// Set up two more radio buttons  
radioButton = new JRadioButton("$60,000");  
radioButton.setMnemonic (KeyEvent.VK_6);  
add (radioButton);  
rbg.add (radioButton);  
radioButton = new JRadioButton("$75,000");  
radioButton.setMnemonic (KeyEvent.VK_7);  
add (radioButton);  
rbg.add (radioButton);  
}  
}
```





## 5.6 Swing Component(JToggleButton)

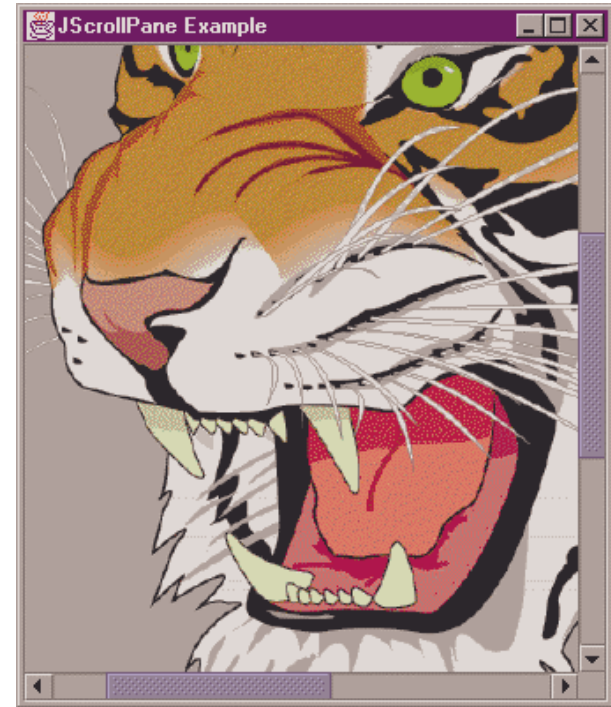
```
public class ToggleButtonPanel extends JPanel {  
    public ToggleButtonPanel() {  
        // Set the layout to a GridLayout  
        setLayout(new GridLayout(4,1, 10, 10));  
        add (new JToggleButton ("Fe"));  
        add (new JToggleButton ("Fi"));  
        add (new JToggleButton ("Fo"));  
        add (new JToggleButton ("Fum"));  
    }  
}
```





# 7 Swing Component(JScrollPane)

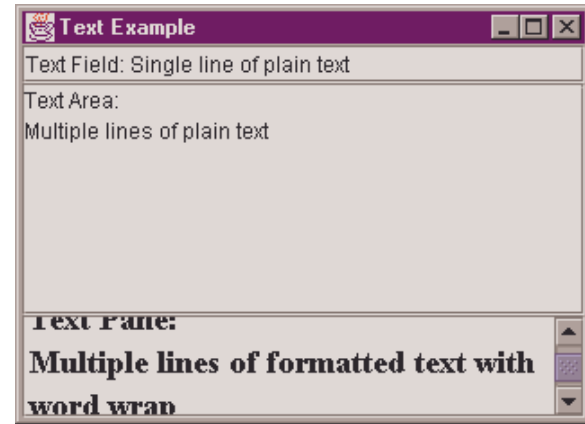
```
public class ScrollPanel extends JPanel {  
  
    public ScrollPanel() {  
        setLayout(new BorderLayout());  
        Icon bigTiger = new ImageIcon("BigTiger.gif");  
        JLabel tigerLabel = new JLabel(bigTiger);  
        JScrollPane scrollPane =  
            new JScrollPane(tigerLabel);  
        add(scrollPane, BorderLayout.CENTER);  
    }  
}
```





# 7 Swing Component(JScrollPane)

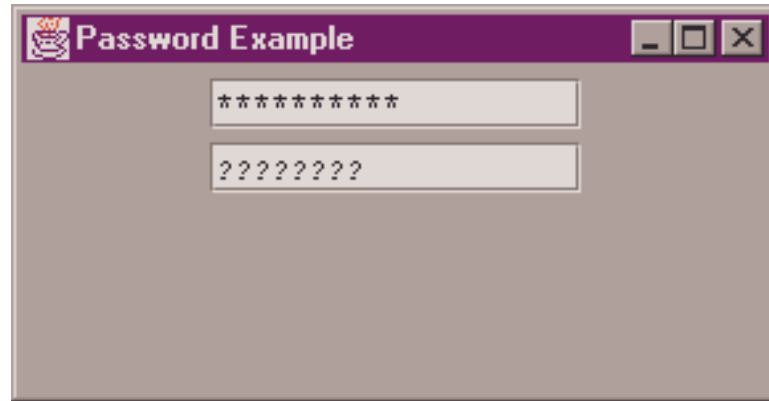
```
public class TextPanel extends JPanel {  
    public TextPanel() {  
        // Set the layout to a BorderLayout  
        setLayout(new BorderLayout());  
        // Create the three basic text components  
        JTextField textField = new JTextField();  
        JTextArea textArea = new JTextArea();  
        JTextPane textPane = new JTextPane();  
        add(textField, BorderLayout.NORTH);  
        add(new JScrollPane(textArea), BorderLayout.CENTER);  
        add(new JScrollPane(textPane), BorderLayout.SOUTH);  
    }  
}
```





# 7 Swing Component(JScrollPane)

```
class PasswordPanel extends JPanel {  
    PasswordPanel() {  
        JPasswordField pass1 = new JPasswordField(20);  
        JPasswordField pass2 = new JPasswordField(20);  
        pass2.setEchoChar ('?');  
        add(pass1);  
        add(pass2);  
    }  
}
```





# 8 Swing Component(JEditorPane)

```
import javax.swing.event.*;
import javax.swing.text.*;
import java.net.*;
import java.io.*;

public class Browser extends JPanel {
    Browser() {
        setLayout (new BorderLayout (5, 5));
        final JEditorPane jt = new JEditorPane();
        final JTextField input = new JTextField("http://java.sun.com");
        jt.setEditable(false); // make read-only
        // follow links
    }
}
```

มีต่อ





# 8 Swing Component(JEditorPane)

```
jt.addHyperlinkListener(new HyperlinkListener () {
    public void hyperlinkUpdate(final HyperlinkEvent e) {
        if (e.getEventType() == HyperlinkEvent.EventType.ACTIVATED) {
            SwingUtilities.invokeLater(new Runnable() {
                public void run() { Document doc = jt.getDocument();
                    try {URL url = e.getURL();
                        jt.setPage(url); input.setText (url.toString());
                    } catch (IOException io) {
                        JOptionPane.showMessageDialog (Browser.this,
                            "Can't follow link", "Invalid Input",JOptionPane.ERROR_MESSAGE);
                        jt.setDocument (doc);
                    }
                }
            });
        }
    }
});
```

มีต่อ



# 8 Swing Component(JEditorPane)

```
JScrollPane pane = new JScrollPane();
pane.setBorder (BorderFactory.createLoweredBevelBorder());
pane.getViewport().add(jt);
add(pane, BorderLayout.CENTER);
input.addActionListener (new ActionListener() {
    public void actionPerformed (ActionEvent e) {
        try {
            jt.setPage (input.getText());
        } catch (IOException ex) {
            JOptionPane.showMessageDialog (Browser.this, "Invalid URL",
                "Invalid Input", JOptionPane.ERROR_MESSAGE);
        }
    }
});
add (input, BorderLayout.SOUTH);
}
```





# 9 Swing Component(JScrollBar)

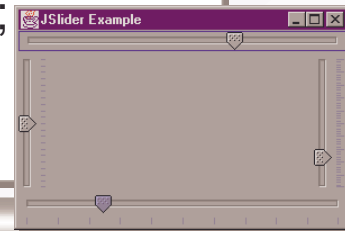
```
public class ScrollbarPanel extends JPanel {  
  
    public ScrollbarPanel() {  
        setLayout(new BorderLayout());  
        JScrollBar scrollBar1 = new JScrollBar (  
            JScrollBar.VERTICAL, 0, 5, 0, 100);  
        add(scrollBar1, BorderLayout.EAST);  
        JScrollBar scrollBar2 = new JScrollBar (  
            JScrollBar.HORIZONTAL, 0, 5, 0, 100);  
        add(scrollBar2, BorderLayout.SOUTH);  
    }  
}
```





## 5.10 Swing Component(JSlider)

```
public class SliderPanel extends JPanel {
    public SliderPanel() {
        setLayout(new BorderLayout());
        JSlider slider1 = new JSlider (JSlider.VERTICAL, 0, 100, 50);
        slider1.setPaintTicks(true);
        slider1.setMajorTickSpacing(10);
        slider1.setMinorTickSpacing(2);
        add(slider1, BorderLayout.EAST);
        JSlider slider2 = new JSlider (JSlider.VERTICAL, 0, 100, 50);
        slider2.setPaintTicks(true);
        slider2.setMinorTickSpacing(5);
        add(slider2, BorderLayout.WEST);
        JSlider slider3 = new JSlider (JSlider.HORIZONTAL, 0, 100, 50);
        slider3.setPaintTicks(true);
        slider3.setMajorTickSpacing(10);
        add(slider3, BorderLayout.SOUTH);
        JSlider slider4 = new JSlider (JSlider.HORIZONTAL, 0, 100, 50);
        slider4.setBorder(BorderFactory.createLineBorder(Color.blue));
        add(slider4, BorderLayout.NORTH);
    }
}
```





## 5.10 Swing Component(JSlider)

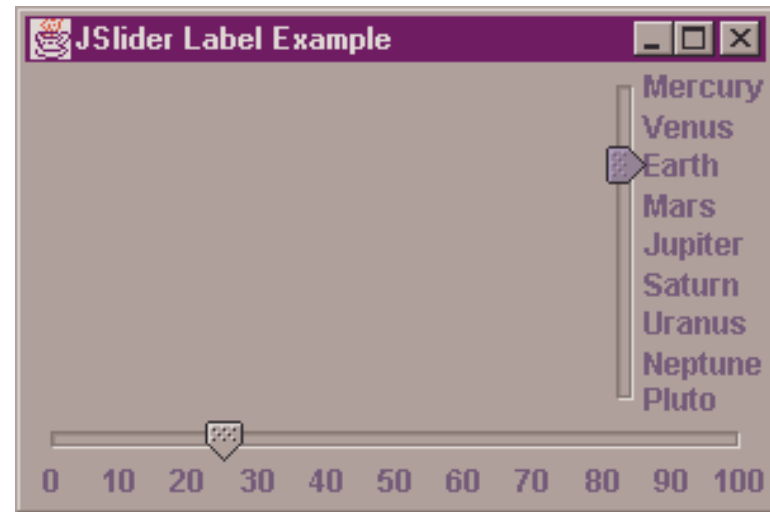
```
public class SliderPanel2 extends JPanel {
    public SliderPanel2() {
        setLayout(new BorderLayout());
        JSlider right, bottom;
        right = new JSlider(JSlider.VERTICAL, 1, 9, 3);
        Hashtable h = new Hashtable();
        h.put (new Integer (1), new JLabel("Mercury"));
        h.put (new Integer (2), new JLabel("Venus"));
        h.put (new Integer (3), new JLabel("Earth"));
        h.put (new Integer (4), new JLabel("Mars"));
        h.put (new Integer (5), new JLabel("Jupiter"));
        h.put (new Integer (6), new JLabel("Saturn"));
        h.put (new Integer (7), new JLabel("Uranus"));
        h.put (new Integer (8), new JLabel("Neptune"));
        h.put (new Integer (9), new JLabel("Pluto"));
    }
}
```

มีต่อ



## 5.10 Swing Component(JSlider)

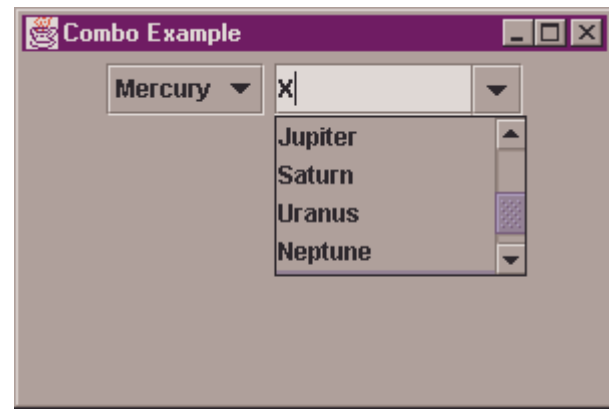
```
right.setLabelTable (h);
right.setPaintLabels (true);
right.setInverted (true);
bottom =
    new JSlider(JSlider.HORIZONTAL, 0, 100, 25);
bottom.setMajorTickSpacing (10);
bottom.setPaintLabels (true);
add(right, BorderLayout.EAST);
add(bottom, BorderLayout.SOUTH);
}
}
// import java.util.*; for HashTable
```





# 5.11 Swing Component(JComboBox)

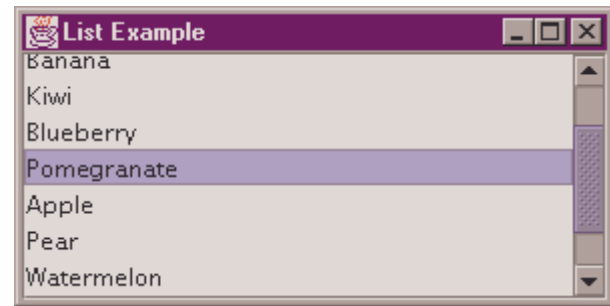
```
public class ComboPanel extends JPanel {  
    String choices[] = {  
        "Mercury", "Venus", "Earth",  
        "Mars", "Jupiter", "Saturn",  
        "Uranus", "Neptune", "Pluto"};  
    public ComboPanel() {  
        JComboBox combo1 = new JComboBox();  
        JComboBox combo2 = new JComboBox();  
        for (int i=0;i<choices.length;i++) {  
            combo1.addItem (choices[i]);  
            combo2.addItem (choices[i]);  
        }  
        combo2.setEditable(true);  
        combo2.setSelectedItem("X");  
        combo2.setMaximumRowCount(4);  
        add(combo1);  
        add(combo2);  
    }  
}
```





## 5.12 Swing Component(JList)

```
public class ListPanel extends JPanel {  
    String label [] = {"Cranberry", "Orange",  
        "Banana", "Kiwi", "Blueberry",  
        "Pomegranate", "Apple", "Pear",  
        "Watermelon", "Raspberry", "Snozberry"};  
};  
public ListPanel() {  
    setLayout (new BorderLayout());  
    JList list = new JList(label);  
    JScrollPane pane = new JScrollPane(list);  
    add(pane, BorderLayout.CENTER);  
}  
}
```







## 5.13 Swing Component(JBorder)

```
import javax.swing.border.*;

public class BorderPanel extends JPanel {
    class MyBorder implements Border {
        Color color;
        public MyBorder (Color c) { color = c; }
        public void paintBorder (Component c, Graphics g,
            int x, int y, int width, int height) {
            Insets insets = getBorderInsets(c);
            g.setColor (color);
            g.fillRect (x, y, 2, height);    g.fillRect (x, y, width, 2);
            g.setColor (color.darker());
            g.fillRect (x+width-insets.right, y, 2, height);
            g.fillRect (x, y+height-insets.bottom, width, 2);
        }
        public boolean isBorderOpaque() { return false; }
        public Insets getBorderInsets(Component c) {
            return new Insets (2, 2, 2, 2);
        }
    }
}
```

มีต่อ



## 5.13 Swing Component(JBorder)

```
public JPanel() {  
    setLayout (new GridLayout (4, 3, 5, 5));  
    JButton b = new JButton("Empty");  
    b.setBorder (new EmptyBorder (1,1,1,1));  
    add(b);  
    b = new JButton ("Etched");  
    b.setBorder (new EtchedBorder ());  
    add(b);  
    b = new JButton ("ColorizedEtched");  
    b.setBorder (new EtchedBorder (Color.red, Color.green));  
    add(b);  
    b = new JButton ("Titled/Line");  
    b.setBorder(new TitledBorder (new TitledBorder(  
        LineBorder.createGrayLineBorder(), "Hello", "World",  
        TitledBorder.RIGHT, TitledBorder.BOTTOM));  
    add(b);  
    b = new JButton ("Bevel Up");  
    b.setBorder(new BevelBorder(BevelBorder.RAISED));  
    add(b);  
}
```

มีต่อ



## 5.13 Swing Component(JBorder)

```
b = new JButton ("Soft Bevel Up");
b.setBorder(new SoftBevelBorder(SoftBevelBorder.RAISED));
add(b);
b = new JButton ("Soft Bevel Down");
b.setBorder(new SoftBevelBorder(SoftBevelBorder.LOWERED));
add(b);
b = new JButton ("Matte");
b.setBorder(new MatteBorder(5, 10, 5, 10, Color.red));
add(b);
b = new JButton ("Matte Icon");
Icon icon = new ImageIcon ("SmallTiger.gif");
b.setBorder(new MatteBorder(10, 10, 10, 10, icon));
add(b);
```

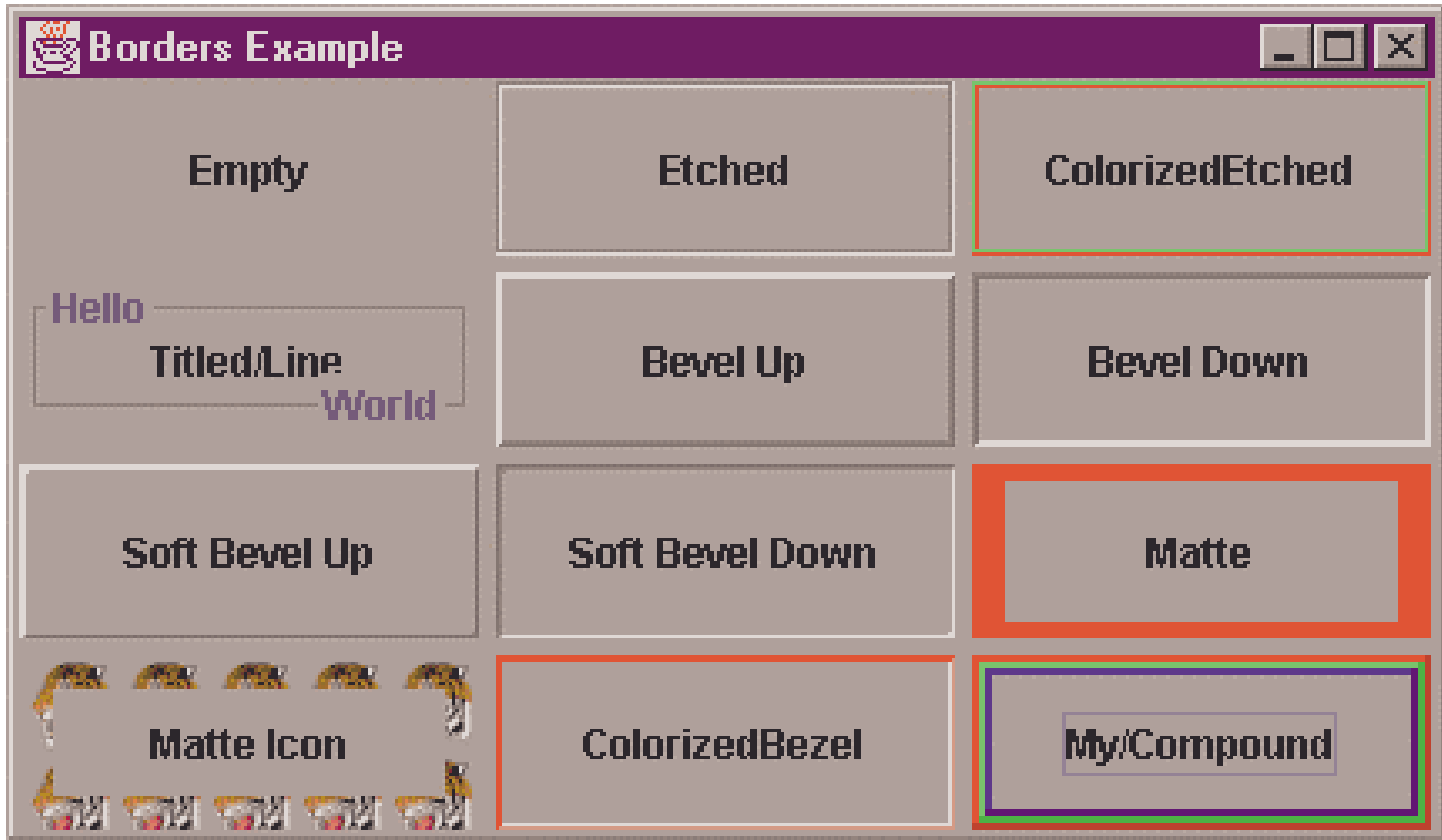
```
b = new JButton ("ColorizedBezel");
b.setBorder(new BevelBorder(BevelBorder.RAISED, Color.red, Color.pink));
add(b);
```

```
b = new JButton ("My/Compound");
b.setBorder(new CompoundBorder(new MyBorder(Color.red),
    new CompoundBorder (new MyBorder(Color.green),
        new MyBorder(Color.blue))));
```

```
add(b);
} }
```



# 5.13 Swing Component(JBorder)





# 14 Swing Component(JMenuBar)

```
public class MenuTester extends JFrame
    implements ActionListener {
    public void actionPerformed (ActionEvent e) {
        System.out.println (e.getActionCommand());
    }
    public MenuTester() {
        super ("Menu Example");
        JMenuBar jmb = new JMenuBar();
        JMenu file = new JMenu ("File");
        JMenuItem item;
        file.add (item = new JMenuItem ("New"));
        item.addActionListener (this);
        file.add (item = new JMenuItem ("Open"));
        item.addActionListener (this);
        file.addSeparator();
        file.add (item = new JMenuItem ("Close"));
        item.addActionListener (this);
        jmb.add (file);
    }
}
```

มีต่อ



# 14 Swing Component(JMenuBar)

```
JMenu edit = new JMenu ("Edit");
edit.add (item = new JMenuItem ("Copy"));
item.addActionListener (this);
Icon tigerIcon = new ImageIcon("SmallTiger.gif");
edit.add (item =
    new JMenuItem ("Woods", tigerIcon));
item.setHorizontalTextPosition (JMenuItem.LEFT);
item.addActionListener (this);
edit.add (item =
    new JMenuItem ("Woods", tigerIcon));
item.addActionListener (this);
jmb.add (edit);
```

```
JMenu choice = new JMenu ("Choices");
JCheckBoxMenuItem check =
    new JCheckBoxMenuItem ("Toggle");
check.addActionListener (this);
choice.add (check);
```

มีต่อ



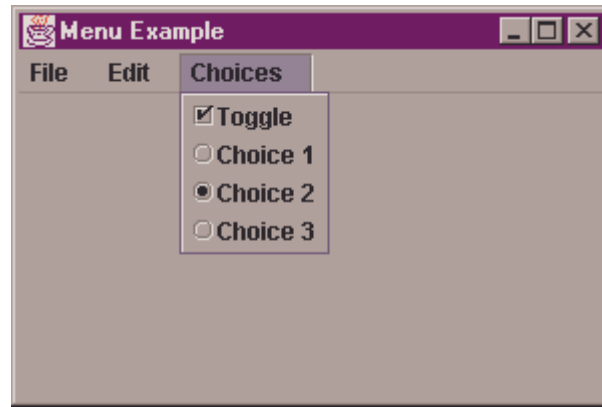
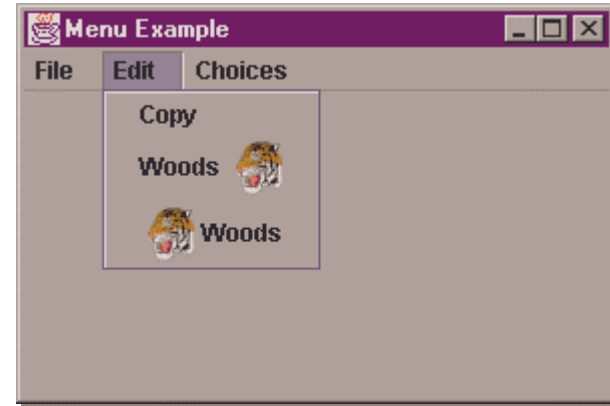
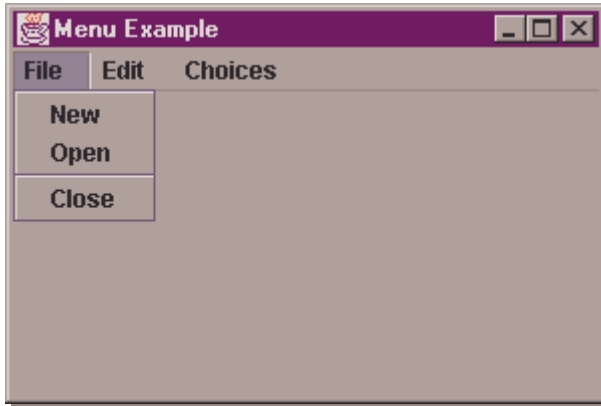
# 14 Swing Component(JMenuBar)

```
ButtonGroup rbg = new ButtonGroup();
    JRadioButtonMenuItem rad = new JRadioButtonMenuItem ("Choice 1");
    choice.add (rad);
    rbg.add (rad);
    rad.addActionListener (this);
    rad = new JRadioButtonMenuItem ("Choice 2");
    choice.add (rad);
    rbg.add (rad);
    rad.addActionListener (this);
    rad = new JRadioButtonMenuItem ("Choice 3");
    choice.add (rad);
    rbg.add (rad);
    rad.addActionListener (this);
    jmb.add (choice);

    setJMenuBar (jmb);
}
}
```



# 14 Swing Component(JMenuBar)

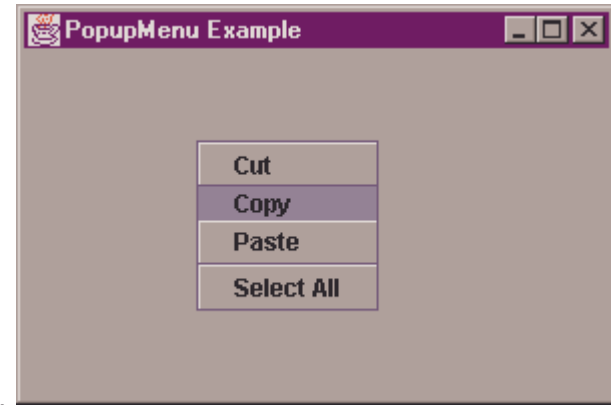






## 5.15 Swing Component(JPopupMenu)

```
public class PopupPanel extends JPanel {
    JPopupMenu popup = new JPopupMenu ();
    public PopupPanel() {
        JMenuItem item;
        popup.add (item = new JMenuItem ("Cut"));
        popup.add (item = new JMenuItem ("Copy"));
        popup.add (item = new JMenuItem ("Paste"));
        popup.addSeparator();
        popup.add (item = new JMenuItem ("Select All"));
        popup.setInvoker (this);
        addMouseListener (new MouseAdapter() {
            public void mousePressed (MouseEvent e) {
                if (e.isPopupTrigger()) { popup.show (e.getComponent(),
                    e.getX(), e.getY()); }
            }
            public void mouseReleased (MouseEvent e) {
                if (e.isPopupTrigger()) { popup.show (e.getComponent(),
                    e.getX(), e.getY()); }
            }
        });
    }
}
```





## 5.16 Swing Component(ToolTip)

```
public class TooltipPanel extends JPanel {  
    public TooltipPanel() {  
        JButton myButton = new JButton("Hello");  
        myButton.setToolTipText ("World");  
        add(myButton);  
    }  
}
```

