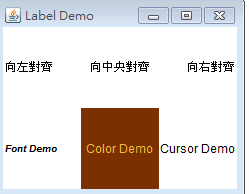
Label & Button

1. 請參考投影片內容，建立以下視窗應用程式

請將1.程式**執行結果**截圖置入作業中、2.程式原始檔置入作業中

1. 建立6個各種樣式的Label
   1. 向左對齊的Label
   2. 向水平中央對齊的Label
   3. 向右對齊的Label
   4. 具有字型的Label
   5. 具有前景顏色與背景顏色的Label
   6. 具有不同滑鼠指標的Label



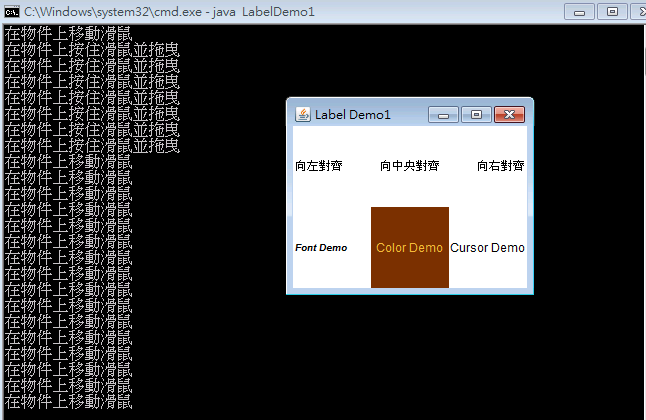
|  |
| --- |
| import java.awt.\*;  import java.awt.event.\*;  public class LabelDemo extends java.awt.Frame {  public static void main(String args[]){  new LabelDemo();  }    // 建構函式  public LabelDemo() {  super("Label Demo");  final int row = 2; // 列  final int column = 3; // 行  setLayout(new GridLayout(row, column));  java.awt.Label label;  label = new Label();  label.setText("向左對齊");  label.setAlignment(Label.LEFT);  add(label);  label = new Label("向中央對齊");  label.setAlignment(Label.CENTER);  add(label);  label = new Label("向右對齊", Label.RIGHT);  add(label);  label = new Label("Font Demo", Label.LEFT);  label.setFont(new Font("微軟正黑體", Font.BOLD | Font.ITALIC, 10));  add(label);  label = new Label("Color Demo", Label.CENTER);  label.setBackground(new Color(120,50,0));  label.setForeground(new Color(245,185,60));  add(label);  label = new Label("Cursor Demo", Label.RIGHT);  label.setCursor(new Cursor(Cursor.HAND\_CURSOR));  add(label);  this.setSize(250, 200);  Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();  Dimension frameSize = this.getSize();  if (frameSize.height > screenSize.height)  frameSize.height = screenSize.height;  if (frameSize.width > screenSize.width)  frameSize.width = screenSize.width;  this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);  this.setVisible(true);    this.addWindowListener(new WindowAdapter() {  public void windowClosing(WindowEvent e) {  System.exit(0);  }  });  }  } |

1. 建立1個Label，並使其具有以下監控事件(只有這個Label有,其他區域都沒有), **執行結果可參考moodle**
   1. Mouse



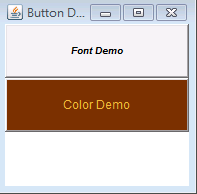
|  |
| --- |
| import java.awt.\*;  import java.awt.event.\*;  import java.awt.event.MouseEvent;  import java.awt.event.MouseListener;  public class LabelDemo extends java.awt.Frame implements MouseListener {  public static void main(String args[]){  new LabelDemo();  }    // 建構函式  public LabelDemo() {  super("Label Demo");  final int row = 2; // 列  final int column = 3; // 行  setLayout(new GridLayout(row, column));  java.awt.Label label;  label = new Label();  label.setText("向左對齊");  label.setAlignment(Label.LEFT);  label.addMouseListener(this);  add(label);  add(label);  label = new Label("向中央對齊");  label.setAlignment(Label.CENTER);  add(label);  label = new Label("向右對齊", Label.RIGHT);  add(label);  label = new Label("Font Demo", Label.LEFT);  label.setFont(new Font("dialog", Font.BOLD | Font.ITALIC, 10));  add(label);  label = new Label("Color Demo", Label.CENTER);  label.setBackground(new Color(120,50,0));  label.setForeground(new Color(245,185,60));  add(label);  label = new Label("Cursor Demo", Label.RIGHT);  label.setCursor(new Cursor(Cursor.HAND\_CURSOR));  add(label);  this.setSize(250, 200);  Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();  Dimension frameSize = this.getSize();  if (frameSize.height > screenSize.height)  frameSize.height = screenSize.height;  if (frameSize.width > screenSize.width)  frameSize.width = screenSize.width;  this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);  // 顯示視窗  this.setVisible(true);    this.addWindowListener(new WindowAdapter() {  public void windowClosing(WindowEvent e) {  System.exit(0);  }  });  }  @Override  public void mouseClicked(MouseEvent arg0) {  // TODO Auto-generated method stub  System.out.println("按下並釋放滑鼠");  }  @Override  public void mouseEntered(MouseEvent arg0) {  // TODO Auto-generated method stub  System.out.println("滑鼠移置物件上");  }  @Override  public void mouseExited(MouseEvent arg0) {  // TODO Auto-generated method stub  System.out.println("滑鼠離開物件");  }  @Override  public void mousePressed(MouseEvent arg0) {  // TODO Auto-generated method stub  System.out.println("按下滑鼠按鍵");  }  @Override  public void mouseReleased(MouseEvent arg0) {  // TODO Auto-generated method stub  System.out.println("釋放滑鼠按鍵");    }      } |

* 1. Mouse Motion Listener



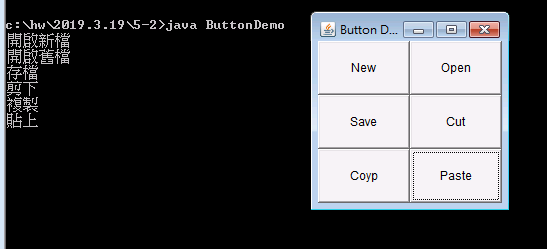
|  |
| --- |
| import java.awt.\*;  import java.awt.event.\*;  import java.awt.event.MouseEvent;  import java.awt.event.MouseListener;  public class LabelDemo1 extends java.awt.Frame implements MouseMotionListener {  public static void main(String args[]){  new LabelDemo1();  }    // 建構函式  public LabelDemo1() {  super("Label Demo1");  final int row = 2; // 列  final int column = 3; // 行  // 定義 Layout Manager 為 GridLayout  setLayout(new GridLayout(row, column));  java.awt.Label label;  label = new Label();  label.setText("向左對齊");  label.setAlignment(Label.LEFT);  label.addMouseMotionListener(this);  add(label);  add(label);  label = new Label("向中央對齊");  label.setAlignment(Label.CENTER);  add(label);  label = new Label("向右對齊", Label.RIGHT);  add(label);  label = new Label("Font Demo", Label.LEFT);  label.setFont(new Font("dialog", Font.BOLD | Font.ITALIC, 10));  add(label);  label = new Label("Color Demo", Label.CENTER);  label.setBackground(new Color(120,50,0));  label.setForeground(new Color(245,185,60));  add(label);  label = new Label("Cursor Demo", Label.RIGHT);  label.setCursor(new Cursor(Cursor.HAND\_CURSOR));  add(label);  this.setSize(250, 200);  Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();  Dimension frameSize = this.getSize();  if (frameSize.height > screenSize.height)  frameSize.height = screenSize.height;  if (frameSize.width > screenSize.width)  frameSize.width = screenSize.width;  this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);  // 顯示視窗  this.setVisible(true);    this.addWindowListener(new WindowAdapter() {  public void windowClosing(WindowEvent e) {  System.exit(0);  }  });  }  @Override  public void mouseDragged(MouseEvent arg0) {  // TODO Auto-generated method stub  System.out.println("在物件上按住滑鼠並拖曳");  }  @Override  public void mouseMoved(MouseEvent arg0) {  // TODO Auto-generated method stub  System.out.println("在物件上移動滑鼠");    }    } |

1. 建立2個各種樣式的Button
   1. 具有字型的Button
   2. 具有前景顏色與背景顏色的Button



|  |
| --- |
| import java.awt.\*;  import java.awt.event.\*;  public class ButtonDemo extends java.awt.Frame {  public static void main(String args[]){  new ButtonDemo();  }    // 建構函式  public ButtonDemo() {  super("Button Demo");  final int row = 3; // 列  final int column = 2; // 行  // 定義 Layout Manager 為 GridLayout  setLayout(new GridLayout(row, column));  java.awt.Button button;  // 設定字型 - 粗斜體字型  button = new Button("Font Demo");  button.setFont(new Font("微軟正黑體", Font.BOLD | Font.ITALIC, 10));  add(button);  // 設定顏色  button = new Button("Color Demo");  // 設定背景顏色  button.setBackground(new Color(120,50,0));  // 設定前景顏色  button.setForeground(new Color(245,185,60));  add(button);  // 設定視窗的大小  this.setSize(200, 200);  // Center the frame  Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();  Dimension frameSize = this.getSize();  if (frameSize.height > screenSize.height)  frameSize.height = screenSize.height;  if (frameSize.width > screenSize.width)  frameSize.width = screenSize.width;  this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);  // 顯示視窗  this.setVisible(true);  this.addWindowListener(new WindowAdapter() {  public void windowClosing(WindowEvent e) {  System.exit(0);  }  });  }  } |

1. 建立6個Button，並使其具有動作事件並有相關動作ㄝ**執行結果可參考moodle**
   1. New，被按下時印出 開啟新檔
   2. Open，被按下時印出 開啟舊檔
   3. Save，被按下時印出 存檔
   4. Cut，被按下時印出 剪下
   5. Copy，被按下時印出 複製
   6. Paste，被按下時印出 貼上



|  |
| --- |
| import java.awt.\*;  import java.awt.event.\*;  public class ButtonDemo extends java.awt.Frame implements ActionListener {  Button button1 = new Button("New");  Button button2 = new Button("Open");  Button button3 = new Button("Save");  Button button4 = new Button("Cut");  Button button5 = new Button("Coyp");  Button button6 = new Button("Paste");      public static void main(String args[]){  new ButtonDemo();  }    // 建構函式  public ButtonDemo() {  super("Button Demo");  final int row = 3; // 列  final int column = 2; // 行  // 定義 Layout Manager 為 GridLayout  setLayout(new GridLayout(row, column));  java.awt.Button button;      button1.addActionListener(this);  add(button1);  button2.addActionListener(this);  add(button2);  button3.addActionListener(this);  add(button3);  button4.addActionListener(this);  add(button4);  button5.addActionListener(this);  add(button5);  button6.addActionListener(this);  add(button6);    // 設定視窗的大小  this.setSize(200, 200);  // Center the frame  Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();  Dimension frameSize = this.getSize();  if (frameSize.height > screenSize.height)  frameSize.height = screenSize.height;  if (frameSize.width > screenSize.width)  frameSize.width = screenSize.width;  this.setLocation((screenSize.width - frameSize.width) / 2, (screenSize.height - frameSize.height) / 2);  // 顯示視窗  this.setVisible(true);  this.addWindowListener(new WindowAdapter() {  public void windowClosing(WindowEvent e) {  System.exit(0);  }  });  }  @Override  public void actionPerformed(ActionEvent arg0) {  // TODO Auto-generated method stub    if(arg0.getSource().equals (button1))  {  System.out.println("開啟新檔");  }    if(arg0.getSource().equals (button2))  {  System.out.println("開啟舊檔");  }    if(arg0.getSource().equals (button3))  {  System.out.println("存檔");  }    if(arg0.getSource().equals (button4))  {  System.out.println("剪下");  }    if(arg0.getSource().equals (button5))  {  System.out.println("複製");  }    if(arg0.getSource().equals (button6))  {  System.out.println("貼上");  }    }  } |