

1、课程名称：标签组件：JLabel



2、本课程预计讲解的知识点



- 了解JLabel类的作用
- 可以设置标签的文字显示风格
- 可以设置标签的内容及图片

3、具体内容

JLabel 表示一个标签，标签就是只用于显示信息的一个组件。

JLabel类的常用方法和常量

No.	方法及常量	类型	描述
1	public static final int LEFT	常量	标签文本左对齐
2	public static final int CENTER	常量	标签文本居中对齐
3	public static final int RIGHT	常量	标签文本右对齐
4	public JLabel() throws HeadlessException	构造	创建一个JLabel对象
5	public JLabel(String text) throws HeadlessException	构造	创建一个标签并指定文本内容，默认为左对齐
6	public JLabel(String text,int alignment) throws HeadlessException	构造	创建一个标签并指定文本内容以及对齐方式，可以使用JLabel.LEFT、JLabel.RIGHT、JLabel.CENTER三个值
7	public JLabel(String text,Icon icon,int horizontalAlignment)	构造	创建具有指定文本、图像和水平对齐方式的JLabel对象
8	public JLabel(Icon image,int horizontalAlignment)	构造	创建具有指定图像和水平对齐方式的JLabel实例
9	public void setText(String text)	普通	设置标签的文本
10	public String getText()	普通	取得标签的文本
11	public void setAlignment(int alignment)	普通	设置标签的对齐方式
12	public void setIcon(Icon icon)	普通	设置指定的图像

```
import java.awt.Dimension ;  
import java.awt.Color ;  
import java.awt.Point ;  
import javax.swing.JLabel ;  
import javax.swing.JFrame ;  
public class JLabelDemo01 {  
    public static void main(String args[]) {  
        JFrame frame = new JFrame("Welcome To MLDN") ;  
        JLabel lab = new JLabel("MLDN",JLabel.CENTER) ; // 实例化标签对象  
        frame.add(lab) ; // 将组件件入到面板之中  
        Dimension dim = new Dimension() ;  
        frame.setBackground(Color.WHITE) ;//设置窗体的背景颜色  
        dim.setSize(200,70) ;  
        frame.setSize(dim) ;  
        Point point = new Point(300,200) ;// 设置坐标  
        frame.setLocation(point) ;  
        frame.setVisible(true) ;  
    }  
}
```

在标签中也可以设置标签的字体。

在标签中也可以设置标签的字体。

更改JLabel的文字样式

- 如果现在要想更改使用的字体，则可以直接使用Component类中定义的以下方法：
 - public void setFont(Font f)
- 在设置的时候使用了java.awt.Font类来表示字体，Font类的常用操作方法及常量如下

No.	方法及常量	类型	描述
1	public static final int BOLD	常量	文字显示为粗体
2	public static final int ITALIC	常量	文字显示风格为斜体
3	public static final int PLAIN	常量	文字显示风格为普通样式
4	public Font(String name,int style,int size)	构造	实例化对象，指定显示风格及大小
5	public String getFontName()	普通	得到字体的名称

```
import java.awt.Dimension ;  
import java.awt.Color ;  
import java.awt.Font ;  
import java.awt.Point ;  
import javax.swing.JLabel ;  
import javax.swing.JFrame ;  
public class JLabelDemo02 {  
    public static void main(String args[]) {  
        JFrame frame = new JFrame("Welcome To MLDN") ;  
        JLabel lab = new JLabel("MLDN",JLabel.CENTER) ; // 实例化标签对象  
        frame.add(lab) ; // 将组件件入到面板之中  
        Dimension dim = new Dimension() ;  
        frame.setBackground(Color.WHITE) ;//设置窗体的背景颜色  
        dim.setSize(200,70) ;  
        frame.setSize(dim) ;  
        Point point = new Point(300,200) ;// 设置坐标  
        frame.setLocation(point) ;  
        frame.setVisible(true) ;  
    }  
}
```

我如何可以得到本机的全部可用字体呢？实际上此类问题可以通过以下固定的一段代码解决：

```
import java.awt.GraphicsEnvironment ;  
public class GetAllFonts {  
    public static void main(String args[]) {  
        GraphicsEnvironment eq = GraphicsEnvironment.getLocalGraphicsEnvironment() ;  
        String fontNames[] = eq.getAvailableFontFamilyNames() ;  
        for(int x=0;x<fontNames.length;x++) {  
            System.out.println(fontNames[x]) ;  
        }  
    }  
}
```

在JLabel中也可以设置显示的图片

如果想要设置显示的图片，则可以使用IO流的方式进行图片的读取。

```
import java.awt.Dimension ;  
import java.awt.Color ;  
import java.awt.Font ;  
import java.awt.Point ;  
import java.io.File ;  
import java.io.InputStream ;  
import java.io.FileInputStream ;  
import javax.swing.JLabel ;  
import javax.swing.JFrame ;  
import javax.swing.ImageIcon ;  
public class JLabelDemo03 {  
    public static void main(String args[]) {  
        JFrame frame = new JFrame("Welcome To MLDN") ;  
        String picPath = "d:" + File.separator + "mldn.gif" ;  
        File file = new File(picPath) ; // 实例化File类的对象  
        InputStream input = null ;  
        byte b[] = new byte[(int)file.length()] ;  
        try {  
            input = new FileInputStream(file) ;  
            input.read(b) ;  
            input.close() ;  
        } catch (Exception e) {  
            e.printStackTrace() ;  
        }  
        ImageIcon icon = new ImageIcon(b) ;  
        JLabel lab = new JLabel("MLDN",icon,JLabel.CENTER) ; // 实例化标签对象  
        Font fit = new Font("Serif",Font.ITALIC + Font.BOLD,28) ;  
        lab.setFont(fit) ;  
        lab.setForeground(Color.RED) ;  
        lab.setBackground(Color.YELLOW) ;//设置窗体的背景颜色  
        frame.add(lab) ; // 将组件件入到面板之中  
        Dimension dim = new Dimension() ;  
        frame.setBackground(Color.WHITE) ;//设置窗体的背景颜色  
        dim.setSize(300,160) ;  
        frame.setSize(dim) ;  
        Point point = new Point(300,200) ;// 设置坐标  
        frame.setLocation(point) ;  
        frame.setVisible(true) ;  
    }  
}
```

但是，以上设置图片的方法实在是太麻烦了，需要通过IO流的方式进行读取。

可以在ImageIcon中设置图片的路径：

```
public ImageIcon(String filename)
```

```
import java.awt.Dimension ;  
import java.awt.Color ;  
import java.awt.Font ;  
import java.awt.Point ;  
import java.io.File ;
```

```
import javax.swing.JLabel ;  
import javax.swing.JFrame ;  
import javax.swing.ImageIcon ;  
public class JLabelDemo04 {  
    public static void main(String args[]) {  
        JFrame frame = new JFrame("Welcome To MLDN") ;  
        String picPath = "d:" + File.separator + "mldn.gif" ;  
        ImageIcon icon = new ImageIcon(picPath) ;  
        JLabel lab = new JLabel("MLDN",icon,JLabel.CENTER) ; // 实例化标签对象  
        Font fit = new Font("Serif",Font.ITALIC + Font.BOLD,28) ;  
        lab.setFont(fit) ;  
        lab.setForeground(Color.RED) ;  
        lab.setBackground(Color.YELLOW) ;//设置窗体的背景颜色  
        frame.add(lab) ; // 将组件件入到面板之中  
        Dimension dim = new Dimension() ;  
        frame.setBackground(Color.WHITE) ;//设置窗体的背景颜色  
        dim.setSize(300,160) ;  
        frame.setSize(dim) ;  
        Point point = new Point(300,200) ;// 设置坐标  
        frame.setLocation(point) ;  
        frame.setVisible(true) ;  
    }  
}
```

4、总结

- 可以通过JLabel设置标签的显示组件
- 可以在标签上设置显示的文字风格或图片。

5、预习任务

按钮组件：JButton

