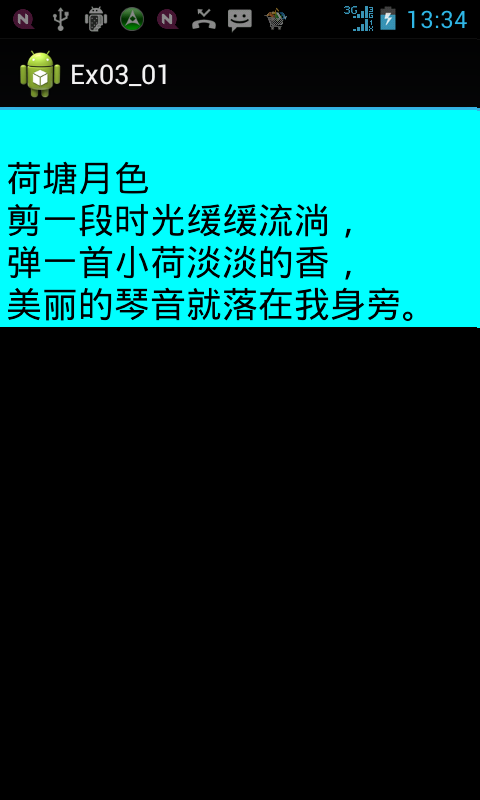
**Experiment 3**

1. 设计一个文本标签组件程序



创建名称为Ex03\_01的新项目，包名为com.ex03\_01。打开系统自动生成的项目框架，需要设计的文件为：

（1）设计布局文件activity\_main.xml；

（2）设计控制文件MainActivity.java；

（3）设计资源文件。

（1）设计界面布局文件activity\_main.xml

　　在界面布局文件activity\_main.xml中加入文本标签组件TextView，设置文本标签组件的id属性。

＜!--设置文本标签的属性值-->

<TextView

android:id="@+id/textView1"

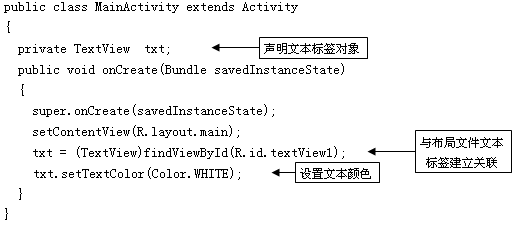
android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

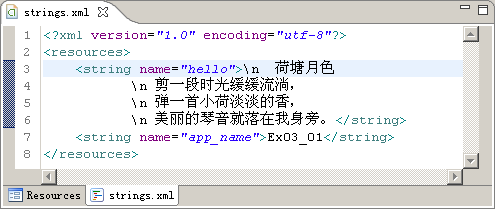
android:text="@string/hello" />

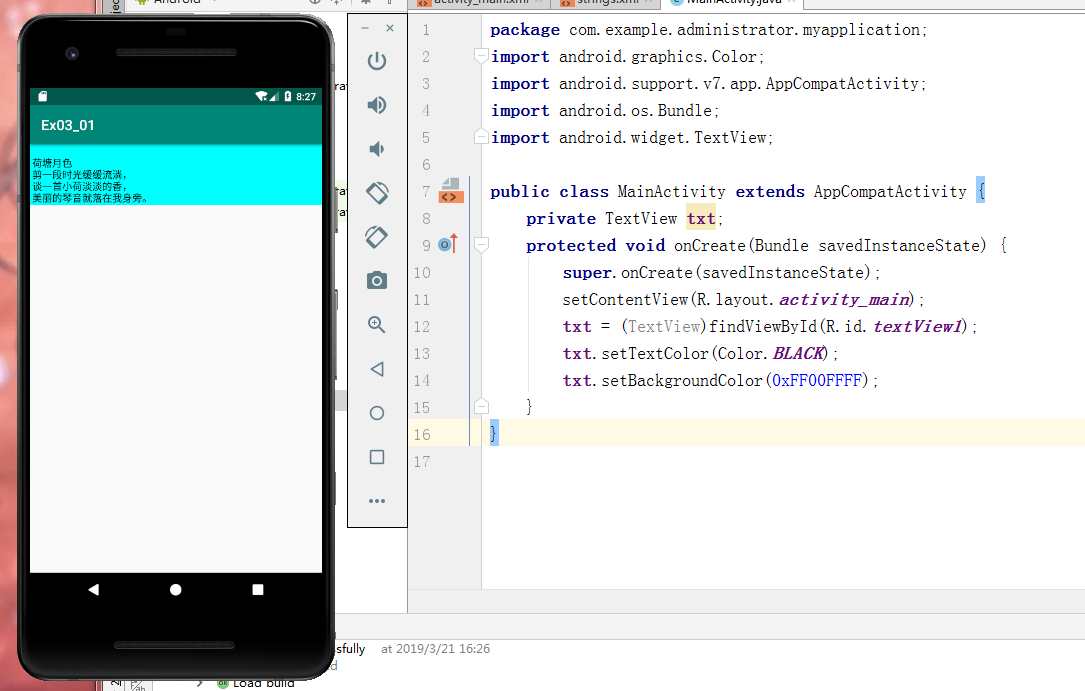
（2）设计控制文件MainActivity.java

在控制文件MainActivity.java源文件在添加文本标签组件，并将布局文件中所定义的文本标签元素属性值赋值给文本标签，与布局文件中文本标签建立关联。



（3）资源文件：strings.xml





1. 按钮Button及按钮处理事件

编写程序，当点击按钮命令后，页面标题及文本组件的文字内容发生变化。



创建名称为Ex03\_02的新项目，包名为com.ex03\_02。

1. 设计布局文件：activity\_main.xml  
   在布局文件中添加一个按钮，其id设置为button1。

1 <?xml version="1.0" encoding="utf-8"?>

2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

3 android:layout\_width="fill\_parent"

4 android:layout\_height="fill\_parent"

5 android:orientation="vertical">

6 <TextView

7 android:id="@+id/textView1"

8 android:layout\_width="fill\_parent"

9 android:layout\_height="wrap\_content"

10 android:text="@string/hello" />

11 <Button

12 android:id="@+id/button1"

13 android:layout\_width="fill\_parent"

14 android:layout\_height="wrap\_content"

15 android:text=" @string/button" />

16 </LinearLayout>

1. 控制文件：Ex03\_02Activity.java  
    在控制文件Ex03\_02Activity.java中，设计一个实现按钮监听接口的内部类mClick，当点击按钮时，触发onClick()事件。

1 package com.ex03\_02;

2 import android.app.Activity;

3 import android.os.Bundle;

4 import android.view.View;

5 import android.view.View.OnClickListener;

6 import android.widget.TextView;

7 import android.widget.Button;

9 public class Ex03\_02Activity extends Activity

10 {

11 private TextView txt;

12 private Button btn;

13 public void onCreate(Bundle savedInstanceState)

14 {

15 super.onCreate(savedInstanceState);

16 setContentView(R.layout.main);

17 txt = (TextView) findViewById(R.id.textView1);

18 btn = (Button)findViewById(R.id.button1);

19 btn.setOnClickListener(new mClick());

20 }

21 class mClick implements OnClickListener

22 {

23 public void onClick(View v)

24 {

25 Ex03\_02Activity.this.setTitle("改变标题");

26 txt.setText(R.string.newStr);

27 }

28 }

29 }

（3）资源文件：strings.xml

1 <?xml version="1.0" encoding="utf-8"?>

2 <resources>

3 <string name="hello">Hello World, 这是Ex03\_02的界面!</string>

4 <string name="app\_name">Ex03\_02</string>

5 <string name="button">点击我！</string>

6 <string name="newStr">改变了文本标签的内容</string>

7 </resources>



1. 设计一个密码验证程序。如图所示



（1）设计布局文件activity\_main.xml  
 在界面布局中，设置一个编辑框，用于输入密码，再设置一个按钮，判断密码是否正确，设置两个文本标签，其中一个显示提示信息“请输入密码”，另一个用于显示密码正确与否。

1 <?xml version="1.0" encoding="utf-8"?>

2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

3 android:layout\_width="fill\_parent"

4 android:layout\_height="fill\_parent"

5 android:orientation="vertical">

6 <TextView

7 android:id="@+id/myTextView01"

8 android:layout\_width="fill\_parent"

9 android:layout\_height="41px"

10 android:layout\_x="33px"

11 android:layout\_y="106px"

12 android:text="请输入密码:"

13 android:textSize="24sp"

14 />

<!--建立一个EditText -->

15 <EditText

16 android:id="@+id/myEditText"

17 android:layout\_width="180px"

18 android:layout\_height="wrap\_content"

19 android:layout\_x="29px"

20 android:layout\_y="33px"

21 android:inputType="text"

22 android:textSize="24sp" />

<!--建立一个Button -->

23 <Button

24 android:id="@+id/myButton"

25 android:layout\_width="100px"

26 android:layout\_height="wrap\_content"

27 android:text="确定"

28 android:textSize="24sp"

29 />

<!--建立一个TextView -->

30 <TextView

31 android:id="@+id/myTextView02"

32 android:layout\_width="180px"

33 android:layout\_height="41px"

34 android:layout\_x="33px"

35 android:layout\_y="106px"

36 android:textSize="24sp"

37 />

38 </LinearLayout>

（2）设计控制文件MainActivity.java  
 在控制文件MainActivity.java中，主要是设计按钮的监听事件，当单击按钮后，从文本编辑框中获取输入的文本内容，与密码“abc123”进行比较。

1 package com.ex03\_04;

2 import android.app.Activity;

3 import android.os.Bundle;

4 import android.view.View;

5 import android.view.View.OnClickListener;

6 import android.widget.EditText;

7 import android.widget.TextView;

8 import android.widget.Button;

9 public class MainActivity extends Activity

10 {

11 private EditText edit;

12 private TextView txt1,txt2;

13 private Button mButton01;

14 @Override

15 public void onCreate(Bundle savedInstanceState)

16 {

17 super.onCreate(savedInstanceState);

18 setContentView(R.layout.main);

19 txt1 = (TextView)findViewById(R.id.myTextView01);

20 txt2 = (TextView)findViewById(R.id.myTextView02);

21 edit = (EditText)findViewById(R.id.myEditText);

22 mButton01 = (Button)findViewById(R.id.myButton);

23 mButton01.setOnClickListener(new mClick());

24 }

定义实现监听接口的内部类

25 class mClick implements OnClickListener

26 {

27 public void onClick(View v)

28 {

29 String passwd;

30 passwd=edit.getText().toString();

31 if(passwd.equals("abc123"))

32 txt2.setText("欢迎进入快乐大本营!");

33 else

34 txt2.setText("非法用户,请立刻离开!");

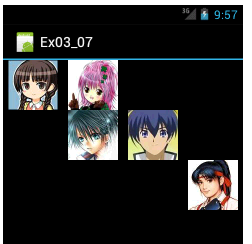
35 }

36 }

37 }

四、表格布局应用示例。设计一个3行4列的表格布局，组件安排如图所示。



创建名称为Ex03\_07的新项目。

生成项目框架后，将准备好的图像文件img1.png、img2.png、img3.png、img4.png、img5.png复制到res\drawable-hdpi目录下。

1. 表格布局的布局文件activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>

<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent“ >

<TableRow><!-- 第1行 -->

<ImageView android:id="@+id/mImageView1"

android:layout\_width="60px"

android:layout\_height="wrap\_content"

android:layout\_column="0" />

<ImageView android:id="@+id/mImageView2"

android:layout\_width="60px"

android:layout\_height="wrap\_content"

android:layout\_column="1" />

</TableRow>

<TableRow><!-- 第2行 -->

<ImageView android:id="@+id/mImageView3"

android:layout\_width="60px"

android:layout\_height="wrap\_content"

android:layout\_column="1" />

<ImageView android:id="@+id/mImageView4"

android:layout\_width="60px"

android:layout\_height="wrap\_content"

android:layout\_column="2" />

</TableRow>

<TableRow><!-- 第3行 -->

<ImageView android:id="@+id/mImageView5"

android:layout\_width="60px"

android:layout\_height="wrap\_content"

android:layout\_column="3" />

</TableRow>

</TableLayout>

1. 控制文件Ex03\_07Activity.java

package com.ex03\_07;

import android.app.Activity;

import android.os.Bundle;

import android.widget.ImageView;

public class Ex03\_07Activity extends Activity

{

ImageView img1, img2, img3, img4, img5;

@Override

public void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

img1 = (ImageView) this.findViewById(R.id.mImageView1);

img2 = (ImageView) this.findViewById(R.id.mImageView2);

img3 = (ImageView) this.findViewById(R.id.mImageView3);

img4 = (ImageView) this.findViewById(R.id.mImageView4);

img5 = (ImageView) this.findViewById(R.id.mImageView5);

img1.setImageResource(R.drawable.img1);

img2.setImageResource(R.drawable.img2);

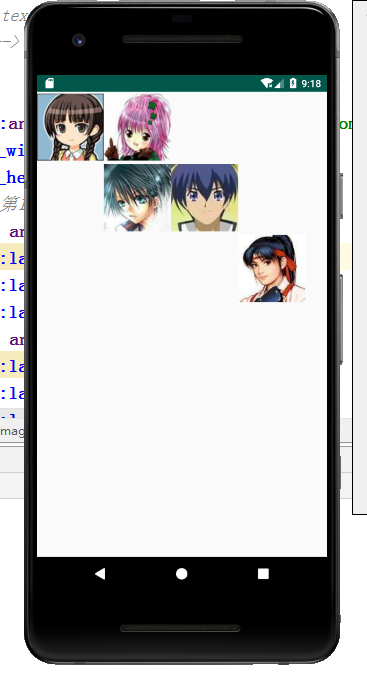
img3.setImageResource(R.drawable.img3);

img4.setImageResource(R.drawable.img4);

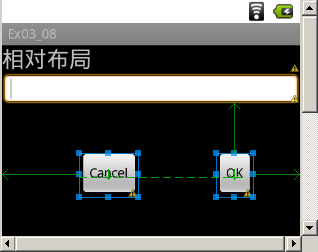
img5.setImageResource(R.drawable.img5);

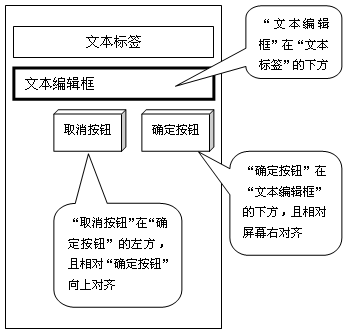
}

}



五、应用相对布局设计一个组件排列如图所示的应用程序。





<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent">

<TextView

android:id="@+id/label"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:textSize="24sp"

android:text="相对布局"/>

<EditText

android:id="@+id/edit"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:background="@android:drawable/editbox\_background"

android:layout\_below="@id/label"/><!-- 指定在文本标签的下方 -->

<Button

android:id="@+id/ok"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below=“@id/edit“ <!-- 指定在文本编辑框的下方 -->

android:layout\_alignParentRight="true"

android:layout\_marginLeft="10dip"

android:text="OK" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_toLeftOf=“@id/ok”　　<!-- 指定在OK按钮的左方 -->

android:layout\_alignTop="@id/ok"

android:text="Cancel" />

</RelativeLayout>

