第 3 章　Android 常用基本控件

我们在进行界面布局时，添加的按钮、文本框、编辑框和图片等，都是 Android 的基本控件。这些控件实现了程序的一些基本功能。本章将针对这类控件进行详细的介绍，使读者掌握基本控件的使用，开发出简单的 Android 程序。

3.1　文本控件概述

Android 系统提供给用户已经封装好的界面控件称为系统控件。系统控件更有利于帮助用户进行快速开发，同时能够使 Android 系统中应用程序的界面保持一致性。

3.1.1　控件属性

Android 支持的基本控件有以下几种，如图 3.1 所示。

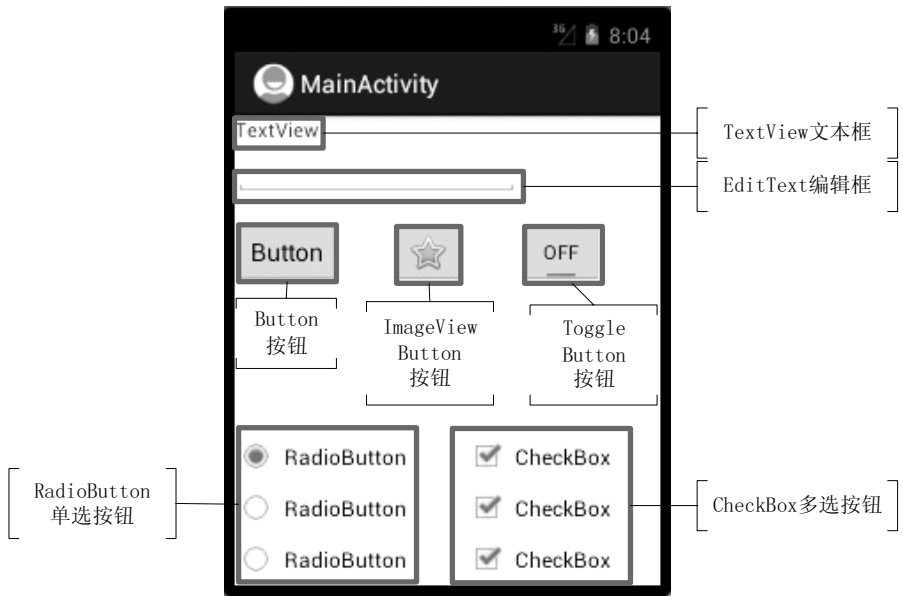

图 3.1　基本控件

注意：由于篇幅有限，图中所列并非 Android 支持的所有基本控件。

Android 的控件，一般是在 res/layout 下的布局文件中声明使用。声明的同时，还要设置控件的属性，控制其在界面中的显示效果。设置控件的属性有两种方法，一种是在布局文件中设置参数，另一种是在代码中调用对应方法实现。控件常用属性及其对应方法如表 3-1 所示。

表 3-1　控件常用属性及其对应方法

属性名称	对应方法	说　　明
id	setId(int id)	设置该控件的id
layout_width	setWidth(int pixels)	设置该控件的宽度
layout_height	setHeight(int pixels)	设置该控件的高度

3.1.2　控件使用

在布局文件的 Graphical Layout 视图中有一个 Palette 面板。该面板中包含了 Android 中的所有控件。我们在使用控件时，可以直接拖动所需控件到右侧手机界面，如图 3.2 所示，添加了一个 Button 控件。也可以手动编辑代码添加控件。

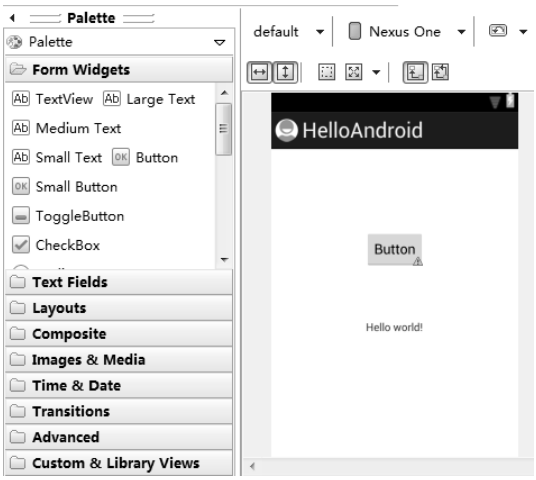

图 3.2　添加控件

在布局文件中声明的控件，只负责界面显示。如果要想使用控件实现某些具体功能，就需要在 Activity 中编辑代码实现。实现过程如下：

（1）使用 super.setContentView(R.layout.某布局 layout 文件名)来加载布局文件；

（2）使用 super.findViewById(R.id.控件的 ID)获取控件引用；

（3）使用这个引用对控件进行操作，例如添加监听，设置内容等。

3.2　文本类控件

文本类控件主要用于在界面中显示文本，包含 TextView 和 EditText 两种。下面我们将详细介绍。

3.2.1　TextView

TextView 是 Android 程序开发中最常用的控件之一，它一般使用在需要显示一些信息的时候，它不能输入，只能通过初始化设置或在程序中修改。TextView 常用属性及其对应方法如表 3-2 所示。

表 3-2　TextView 常用属性及对应方法说明

属性名称	对 应 方 法	说　　明
android:autoLink	setAutoLinkMask(int)	设置是否将指定格式的文本转化为可点击的超链接显示。传入的参数值可取ALL、EMAIL_ADDRESSES、MAP_ADDRESSES、PHONE_NUMBERS和WEB_URLS
android:height	setHeight(int)	定义TextView的准确高度，以像素为单位
android:width	setWidth(int)	定义TextView的准确宽度，以像素为单位
android:singleLine	setTransformationMethod(TransformationMethod)	设置文本内容只在一行内显示
android:text	setText(CharSequence)	为TextView设置显示的文本内容
android:textColor	setTextColor(ColorStateList)	设置TextView的文本颜色
android:textSize	setTextSize(float)	设置TextView的文本大小
android: textStyle	setTypeface(Typeface)	设置TextView的文本字体
android:ellipsize	setEllipsize(TextUtils.TruncateAt)	如果设置了该属性，当TextView中要显示的内容超过了TextView的长度时，会对内容进行省略，可取的值有start、middle、end和marquee

TextView 文本字体属性示意图如图 3.3 所示。

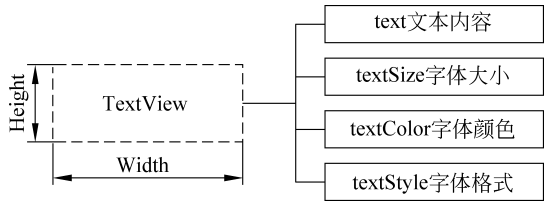

图 3.3　TextView 文本字体属性示意图

TextView 语法格式如下：

```
<TextView
    <!-- TextView 边框包围内容-->
    android:layout_width=" "
    android:layout_height=" "
    <!-- TextView 准确高度宽度-->
    android:width=" "
    android:height=" "
    android:text=" "
    <!--字体大小-->
    android:textSize=" "
    android:textColor=" "
    <!--字体格式-->
    android:textStyle=" "
    <!--文本显示位置-->
    android:gravity=" "
    <!--是否转为可点击的超链接形式-->
    android:autoLink=" "
    <!--是否只在一行内显示全部内容-->
    android:singleLine=" "

    android:ellipsize=" "/>
```

【示例 3-1】 TextView 的使用。新建项目 TextView，在布局中添加三个 TextView。第

一个 TextView 的文本以 web 形式显示“http://www.google.com”，第二个 TextView 的文本只进行一些字体设置，第三个 TextView 的文本以省略尾部内容显示 26 个英文字母。运行程序，效果如图 3.4 所示。

布局代码如下：

```
<TextView
       android:id="@+id/textView1"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:text="@string/tv1"
       android:textSize="20sp"
       android:autoLink="web"
       android:singleLine="true"/>

    <TextView
       android:id="@+id/textView2"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_alignParentLeft="true"
       android:layout_below="@+id/textView1"
       android:layout_marginTop="20dp"
       android:textSize="30sp"
       android:textColor="#0000FF"
       android:textStyle="italic"
       android:text="@string/tv2" />

    <TextView
       android:id="@+id/textView3"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:layout_alignParentLeft="true"
       android:layout_below="@+id/textView2"
       android:layout_marginTop="20dp"
       android:textSize="30sp"
       android:singleLine="true"
       android:ellipsize="end"
       android:text="@string/tv3" />
```

图 3.4　TextView

控件	属　性	值
TextView	id	@+id/textview1
	textSize	20sp
	auto_Link	web
	singleLine	true
	text	http://www.google.com
TextView	id	@+id/textview2
	textSize	30sp
	textColor	#0000ff
	textStyle	italic
	text	TextView
TextView	id	@+id/textview3
	textSize	30sp
	singleLine	true
	ellipsize	end
	text	abcdefghijklmnopqrstuvwxyz

3.2.2　EditText

我们在第一次使用一些应用软件时，常常需要输入用户名和密码进行注册和登录。实现此功能，就需要使用 Android 系统中的编辑框 EditText。EditText 也是一种文本控件，除了 TextView 的一些属性外，EditText 还有一些特有的属性，如表 3-3 所示。

表 3-3　EditText 常用属性及对应方法说明

属 性 名 称	对 应 方 法	说　　明
android:lines	setLines(int)	通过设置固定的行数来决定EditText的高度
android:maxLines	setMaxLines(int)	设置最大的行数
android:minLines	setMinLines(int)	设置最小的行数
android:inputType	setTransformationMethod(TransformationMethod)	设置文本框中的内容类型，可以是密码、数字、电话号码等类型
android:scrollHorizontally	setHorizontallyScrolling(boolean)	设置文本框是否可以水平滚动
android: capitalize	setKeyListener(KeyListener)	如果设置，自动转换用户输入的内容为大写字母
android: hint	setHint(int)	文本为空时，显示提示信息
android:maxLength	setFilters(InputFilter)	设置最大显示长度

Edittext 属性示意图如图 3.5 所示。

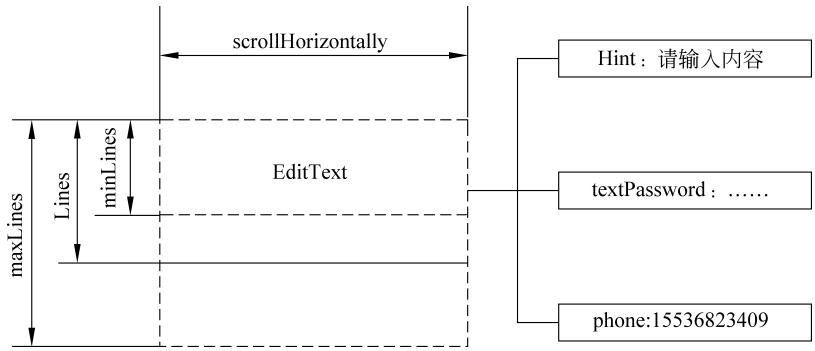

图 3.5　Edittext 属性示意图

EditText 语法格式如下：

```
<EditText
    <!--文本提示内容-->
    android:hint=""
    <!--文本内容显示在固定行中-->
    android:lines=""
    <!--文本最大显示长度-->
    android:maxLength=" "
    <!--文本显示类型-->
    android:inputType=" "

    android:scrollHorizontally=""/>
```

【示例 3-2】 EditText 的使用。新建项目 EditText，在布局文件中添加三个 EditText。

第一个提示输入密码；第二个输入电话号码；第三个输入内容全部转为大写，并限制文本长度。运行程序，效果如图 3.6 所示。

布局代码如下：

```
<EditText
    android:id="@+id/EditText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:password="true"
    android:hint="请输入密码">
</EditText>

<EditText
    android:id="@+id/EditText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/EditText1"
    android:layout_marginTop="26dp"
    android:phoneNumber="true"
    android:lines="1" />

<EditText
    android:id="@+id/EditText3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/EditText2"
    android:layout_marginTop="26dp"
    android:maxLength="10"
    android:scrollHorizontally="true"
    android:capitalize="characters" />
```

图 3.6　EditText

控件	属　性	值
EditText	hint	请输入密码
	password	true
EditText	phoneNumber	true
	lines	1
EditText	maxLength	10
	scrollHorizontally	true
	capitalize	characters

3.3　Button 类控件

Button 类控件主要包括 Button、ImageButton、ToggleButton、RadioButton 和 CheckBox。下面我们将详细介绍。

3.3.1　Button

Button 是 Android 程序开发过程中，较为常用的一类控件。用户可以通过单击 Button 来触发一系列事件，然后为 Button 注册监听器，来实现 Button 的监听事件。

为 Button 注册监听有两种方法，一种是在布局文件中，为 Button 控件设置 OnCilck 属性，然后在代码中添加一个 public void OnCilck 属性值{}方法；另一种是在代码中绑定匿名监听器，并且重写 onClick 方法。下面我们通过例子来演示为 Button 注册监听。

【示例 3-3】 新建项目 Button，在布局中添加 Button1 和 Button2。在 Activity 中编辑代码为 Button1 注册监听，单击 Button1，修改界面标题“Button1 注册成功”；在布局文件中为 Button2 设置 OnClick 属性值注册监听，单击 Button2，修改界面标题“Button2 注册成功”。

布局文件代码：

```
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/button1"
        />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_marginTop="60dp"
        <!--设置 OnClick 属性-->
        android:onClick="click"
        android:text="@string/button2" />
```

逻辑代码：

```
public class MainActivity extends Activity {
    //声明 Button1、Button2
    Button button1,button2;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        //加载布局文件
        setContentView(R.layout.activity_main);
        //获取 Button1、Button2 引用
        button1 = (Button)findViewById(R.id.button1);
        button2 = (Button)findViewById(R.id.button2);
        //为 Button1 注册监听
        button1 .setOnClickListener(new OnClickListener() {
             public void onClick(View v) {
                 // TODO Auto-generated method stub
                 setTitle("Button1 注册成功");
             }
         });
    }

        //为 Button2 注册监听,方法名为 OnClick 属性值
        public void click(View v) {
```

```
        setTitle("Button2 注册成功");
        }
}
```

注意：Button 控件的 OnClick 属性，其参数值为在代码中添加的对应方法名，因此在设置该参数值时，需注意命名规范。

程序执行过程如图 3.7 所示。

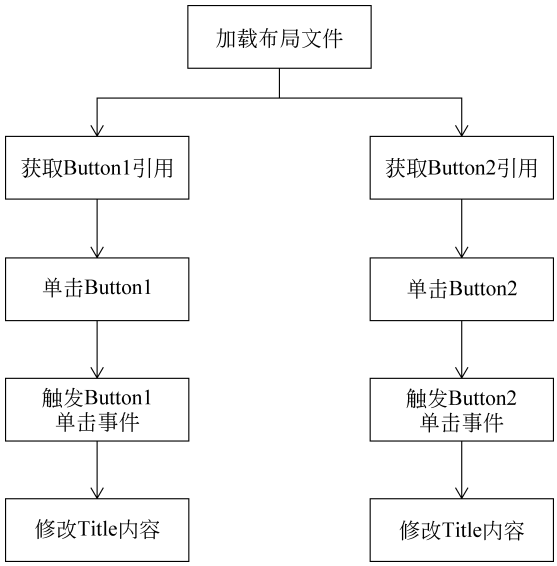

图 3.7　程序执行流程图

运行程序，效果如图 3.8 所示。

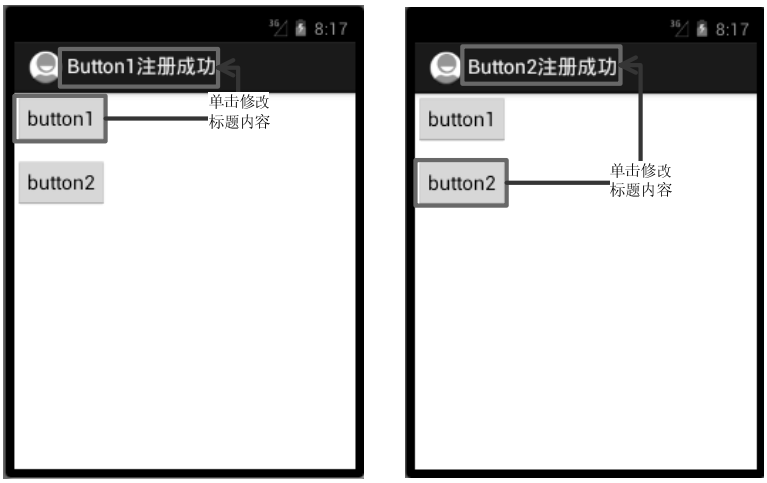

图 3.8　Button 监听

3.3.2　ImageButton

ImageButton（图片按钮）也是一种 Button。它与 Button 控件类似，只是在设置图片时有些区别。ImageButton 控件中，设置按钮显示的图片可以通过 android:src 属性，也可以通

过 setImageResource(int)方法来设置。

ImageButton 语法格式如下：

```
<ImageButton
       <!-- ImageButton 按钮的 ID -->
       android:id=" "
       <!-- ImageButton 宽度和高度-->
       android:layout_width=" "
       android:layout_height=" "
       <!-- ImageButton 背景图片-->
       android:src=" " />
```

【示例 3-4】 ImageButton 的使用。新建项目 ImageButton，添加两个 ImageButton 控件。第一个使用 drawable 中的图片资源作为按钮背景，第二个使用系统提供的图片作为按钮背景。运行程序，效果如图 3.8 所示。

布局代码如下：

```
    <ImageButton
        android:id="@+id/imageButton1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:src="@drawable/paint" />

    <ImageButton
        android:id="@+id/imageButton2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_below="@+id/imageButton1"
        android:layout_marginTop="42dp"
        android:src="@android:drawable/btn_minus" />
```

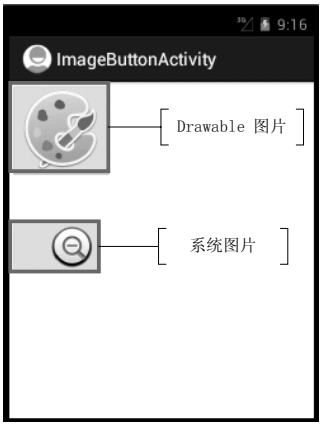

图 3.8　ImageButton

控件	属　性	值
ImageButton	src	@drawable/paint
ImageButton	src	@android:drawable/btn_minus

注意：在设置 src 属性时，加载 Drawable 对象，参数值则为@drawable/对象名；加载系统提供的资源图片，参数值则为@android:drawable/图片名。

【示例 3-5】 下面演示一个单击 ImageButton，改变其背景图片的案例。

首先，在 res/drawable-mdpi 目录下新建一个 myselector.xml，在其中输入如下代码：

```
<?xml version="1.0" encoding="utf-8"?>
 <selector xmlns:android="http://schemas.android.com/apk/res/android">
    <!--未点击时显示背景-->
    <item android:state_pressed="false"
         android:drawable="@drawable/ic_action_search" />
    <!--点击时显示背景-->
    <item android:state_pressed="true"
         android:drawable="@drawable/ic_launcher" />
</selector>
```

然后，设置布局文件中，ImageButton 控件的 src 属性参数为 myselector.xml 的引用：

```
   android:src="@drawable/myselector"
```

selector 是 Android 控件的背景选择器，采用 XML 文件格式。我们可以通过设置 item 项中的以下属性，然后引用图片改变 ImageButton 显示背景。

- android:state_selected：选中；
- android:state_focused：获得焦点；
- android:state_pressed：点击；
- android:state_enabled：设置是否响应事件。

运行程序，效果如图 3.9 所示。

图 3.9　单击改变 ImageButton 背景图片

3.3.3　ToggleButton

ToggleButton（开关按钮）是 Android 系统中比较简单的一个组件，它带有亮度指示，具有选中和未选中两种状态（默认为未选中状态），并且需要为不同的状态设置不同的显示文本。ToggleButton 常用属性及对应方法如表 3-4 所示。

表 3-4　ToggleButton 常用属性及对应方法说明

属性名称	对应方法	说　　明
android:disabledAlpha		设置按钮在禁用时的透明度，属性值必须为浮点型
android:textoff	setTextOff(CharSequence textOff)	未选中时按钮的文本
android:texton	setTextOn(CharSequence textOn)	选中时按钮的文本

ToggleButton 语法格式如下：

```
<ToggleButton
    <!-- ToggleButton 按钮的 ID -->
    android:id=" "
    <!-- ToggleButton 被选中时显示的文本内容-->
    android:textOn=" "
    <!-- ToggleButton 未被选中时显示的文本内容-->
    android:textOff=" "/>
```

【示例 3-6】 ToggleButton 的使用。新建项目 ToggleButton，在布局文件中添加一个 ToggleButton 控件。设置其被选中时显示“开”，未被选中时显示“关”。运行程序，效果如图 3.10 所示。

布局代码如下：

```
<ToggleButton
    android:id="@+id/toggleButton1"
    android:layout_width="1500dp"
    android:layout_height="80dp"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:textOn="开"
    android:textOff="关" />
```

控件	属性	值
ToggleButton	textOn	开
	textOff	关

图 3.10　ToggleButton

3.3.4　RadioButton

RadioButton（单选按钮）在 Android 平台上也比较常用，比如一些选择项会用到单选按钮。它是一种单个圆形单选框双状态的按钮，可以选择或不选择。在 RadioButton 没有被选中时，用户通过单击来选中它。但是，在选中后，无法通过单击取消选中。

单选按钮由 RadioButton 和 RadioGroup 两部分组成。RadioGroup 是单选组合框，用于将 RadioButton 框起来。在多个 RadioButton 被 RadioGroup 包含的情况下，同一时刻只可以选择一个 RadioButton，并用 setOnCheckedChangeListener 来对 RadioGroup 进行监听。RadioButton 语法格式如下：

```
<RadioGroup
      <!-- RadioGroup 单选组合框的 ID -->
```

```
    android:id=" "
    <!-- RadioButton 排列方式-->
    android:orientation=" " >
    <RadioButton
    <!-- RadioButton 单选按钮的 ID -->
       android:id=" "
    <!-- RadioButton 文本内容-->
       android:text=" " />
 ……
</RadioGroup>
```

【示例 3-7】 RadioButton 的使用。新建项目 RadioButton，在布局文件中添加一个 TextView 显示“请选择：”；添加一个 RadioGroup 控件，设置 RadioButton 以垂直方式排列；在 RadioGroup 控件中添加两个 RadioButton 控件，分别显示“火车”和“飞机”；再添加一个 TextView 显示“您选择的是：”。在逻辑代码部分编辑代码，当选中不同选项时，在第二个 TextView 后追加显示选项内容。运行程序，效果如图 3.11 所示。

图 3.11　RadioButton

控件	属　性	值
TextView	id	@+id/tv1
	text	请选择：
RadioGroup	id	@+id/rg1
	orientation	vertical
Radiobutton	id	@+id/rb1
	text	火车
Radiobutton	id	@+id/rb2
	text	飞机
TextView	id	@+id/tv2
	text	您选择的是：

布局代码如下：

```
 <TextView
     android:id="@+id/tv1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="30sp"
    android:text="请选择:" />
<RadioGroup
    android:id="@+id/rg1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="vertical" >
    <RadioButton
       android:id="@+id/rb1"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:textSize="30sp"
       android:text="火车" />
     <RadioButton
       android:id="@+id/rb2"
       android:layout_width="wrap_content"
       android:layout_height="wrap_content"
       android:textSize="30sp"
       android:text="飞机" />
```

```
    </RadioGroup>
     <TextView
         android:id="@+id/tv2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="30sp"
        android:text="您选择的是：" />
```

关键逻辑代码：

```
  //为 RadioGroup 注册监听
  radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener(){
                public void onCheckedChanged(RadioGroup group, int checkedId){
                // TODO Auto-generated method stub
                //通过 id 判断第一个 RadioButton 被选中
                if (checkedId == R.id.rb1) {
                    //显示第一个 RadioButton 内容
                    textView.setText("您选择的是：" + radioButton1.getText());
                //第二个 RadioButton 被选中
                }else {
                    //显示第二个 RadioButton 内容
                    textView.setText("您选择的是：" + radioButton2.getText());
                }
            }
        });
```

3.3.5　CheckBox

CheckBox（复选按钮），顾名思义是一种可以进行多选的按钮，默认以矩形表示。与 RadioButton 相同，它也有选中或者不选中双状态。我们可以先在布局文件中定义多选按钮，然后对每一个多选按钮进行事件监听 setOnCheckedChangeListener，通过 isChecked 来判断选项是否被选中，做出相应的事件响应。CheckBox 语法格式如下：

```
    <CheckBox
        <!-- CheckBox 复选按钮 ID-->
        android:id=" "
        <!-- CheckBox 文本内容-->
        android:text=" " />
```

【示例 3-8】 CheckBox 的使用。新建项目 CheckBox，在布局文件中添加一个 TextView 显示“请选择”；添加三个 CheckBox 控件，分别显示“火车”、“飞机”和“轮船”；再添加一个 TextView 显示“您选择的是：”。在逻辑代码部分编辑代码，当选中不同选项时，在第二个 TextView 后追加显示选项内容。运行程序，效果如图 3.12 所示。

布局代码如下：

```
  <TextView
        android:id="@+id/textView1"
        android:layout_width="121dp"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="请选择" />

    <CheckBox
        android:id="@+id/checkBox1"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="火车" />

    <CheckBox
        android:id="@+id/checkBox2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="飞机" />

    <CheckBox
        android:id="@+id/checkBox3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="轮船" />

     <TextView
        android:id="@+id/textview2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25sp"
        android:text="您选择的是：" />
```

图 3.12　CheckBox

控件	属　性	值
TextView	id	@+id/textview1
	text	请选择
CheckBox	id	@+id/checkbox1
	text	火车
CheckBox	id	@+id/checkbox2
	text	飞机
CheckBox	id	@+id/checkbox3
	text	轮船
TextView	id	@+id/textview2
	text	您选择的是：

关键逻辑代码：

```
  //为第一个 CheckBox 注册监听
  checkBox1.setOnCheckedChangeListener(new OnCheckedChangeListener() {
            public void onCheckedChanged(CompoundButton buttonView,
            boolean isChecked){
                //如果第一个 CheckBox 被选中
                if (isChecked == true) {
                    //显示第一个 CheckBox 内容
                    textView.append(checkBox1 .getText() + ",");
                }
            }
        });
       //为第二个 CheckBox 注册监听
       checkBox2.setOnCheckedChangeListener(new OnCheckedChangeListener()
{
            //如果第二个 CheckBox 被选中
```

```
            public   void   onCheckedChanged(CompoundButton   buttonView,
boolean isChecked) {
                if (isChecked == true) {
                    //显示第二个 CheckBox 内容
                    textView.append(checkBox2 .getText() + ",");
                }
            }
        });
        //为第三个 CheckBox 注册监听
        checkBox3.setOnCheckedChangeListener(new OnCheckedChangeListener()
{
            public void onCheckedChanged(CompoundButton buttonView,
            boolean isChecked) {
            //如果第三个 CheckBox 被选中
                if (isChecked == true) {
                    //显示第三个 CheckBox 内容
                    textView.append(checkBox3 .getText() + ",");
                }

            }
        });
```

3.4　图片控件 ImageView

ImageView 是一个图片控件，负责显示图片。图片的来源可以是系统提供的资源文件，也可以是 Drawable 对象。ImageView 常用的属性及其对应方法如表 3-5 所示。

表 3-5　ImageView 常用属性及对应方法说明

属 性 名 称	对 应 方 法	说　　明
android:adjustViewBounds	setAdjustViewBounds(boolean)	设置是否需要 ImageView 调整自己的边界来保证所显示的图片的长宽比例
android:maxHeight	setMaxHeight(int)	ImageView 的最大高度，可选
android:maxWidth	setMaxWidth(int)	ImageView 的最大宽度，可选
android:scaleType	setScaleType(ImageView.ScaleType)	控制图片应如何调整或移动来适合 ImageView 的尺寸
android:src	setImageResource(int)	设置 ImageView 要显示的图片

ImageView 语法格式如下：

```
   <ImageView
      <!-- ImageView 图片控件 ID-->
      android:id=" "
      <!--是否保持长宽比-->
      android:adjustViewBounds=" "
      <!-- ImageView 最大高度和最大宽度-->
      android:maxHeight=" "
      android:maxWidth=" "
      <!--是否调整图片适应 ImageView-->
      android:scaleType=" "
      android:src=" " />
```

【示例 3-9】 ImageView 的使用。新建项目 ImageView，在布局文件中添加两个

ImageView，第一个显示系统图片，第二个显示 drawable 图片。运行程序，效果如图 3.13 所示。

布局代码如下：

```
<ImageView
    android:id="@+id/imageView1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:src="@android:drawable/btn_star" />

<ImageView
    android:id="@+id/imageView2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/imageView1"
    android:layout_marginTop="74dp"
    android:adjustViewBounds="true"
    android:maxHeight="300dp"
    android:maxWidth="300dp"
    android:scaleType="fitXY"
    android:src="@drawable/paint" />
```

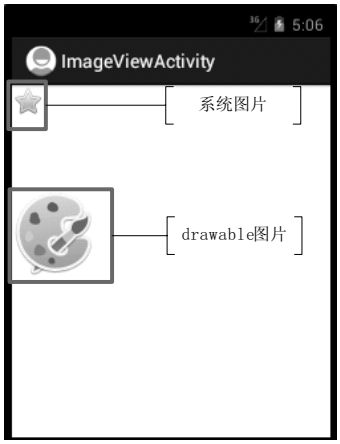

图 3.13　ImageView

控件	属　性	值
ImageView	id	@+id/imageView1
	src	@android:drawable/btn_star
ImageView	id	@+id/imageView2
	src	@drawable/paint
	adjustViewBounds	true
	scaleType	fitXY
	maxHeight	300dp
	maxWidth	300dp

3.5　时 钟 控 件

时钟控件包括 AnalogClock 和 DigtialClock，这两种控件都负责显示时间。不同的是，AnalogClock 是模拟时钟，只显示时针和分针；而 DigtialClock 显示数字时钟，可精确到秒。两者可以结合使用，能更准确的表达时间。

【示例 3-10】 结合使用 AnalogClock 和 DigtialClock。新建项目 Clock，在布局文件中添加一个 AnalogClock 控件和一个 DigtialClock 控件，显示系统时间。运行程序，效果如图 3.14 所示。

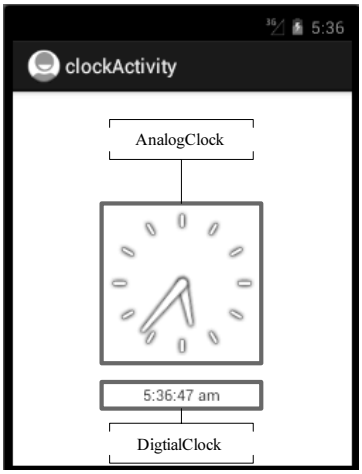

图 3.14　AnalogClock 和 DigtialClock 结合使用

3.6　日期与时间控件

Android 为用户提供了显示日期与时间的控件 DatePicker 和 TimePicker，下面我们将详细介绍。

3.6.1　DatePicker

日期选择控件（DatePicker）主要的功能向用户提供包含了年、月、日的日期数据，并允许用户对其进行选择。DatePicker 相关属性如表 3-6 所示。

表 3-6　DatePicker 相关属性

属 性 名 称	属 性 说 明
calendarViewShown	是否显示日历视图
maxDate	日历视图显示的最大日期，格式为 mm/dd/yyyy
minDate	日历视图显示的最小日期，格式为 mm/dd/yyyy
spinnersShown	是否显示微调控件

DatePicker 语法格式如下：

```
<DatePicker
       <!-- DatePicker ID-- >
       android:id=" "
       <!--是否显示日历视图-->
       android:calendarViewShown=" "
       <!--日历视图显示的最小日期和最大日期,格式为 mm/dd/yyyy-- >
       android:minDate=" "
       android:maxDate=" "
       <!--是否调整图片适应 ImageView-->
       android:spinnersShown=" "/>
```

【示例 3-11】 DatePicker 的使用。新建项目 DatePicker，在布局文件中添加一个 DatePicker 显示系统日期。设置其显示日历视图和微调控件，并设定日历视图显示的最大日期和最小日期。运行程序，效果如图 3.15 所示。使用微调控件，可以修改日期。

布局代码如下：

```
<DatePicker
    android:id="@+id/datePicker1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:minDate="1-1-1970"
    android:maxDate="12-31-2040"/>
```

控件	属　性	值
DatePicker	id	@+id/datePicker1
	calendarViewShown	true
	maxDate	12-31-2040
	minDate	1-1-1970
	spinnersShown	true

图 3.15　DatePicker

如果将上述布局文件中 DatePicker 的 android:spinnersShown 属性设置为 false，就只显示日历视图，如图 3.16 所示。

图 3.16　只显示日历视图

3.6.2　TimePicker

时间选择控件（TimePicker）向用户显示一天中的时间（可以为 24 小时制，也可以为 AM/PM 制），并允许用户进行修改。

【示例 3-12】 TimePicker 的使用。新建项目 TimePicker，在布局文件中添加一个 TimePicker，以 AM/PM 制显示系统时间。运行程序，效果如图 3.17 所示。

【示例 3-13】 在 TimePickerActivity 中添加代码，使用 TimePicker 以 24 小时制显示

系统时间。运行程序，效果如图 3.18 所示。

图 3.17　AM/PM 制 TimePicker

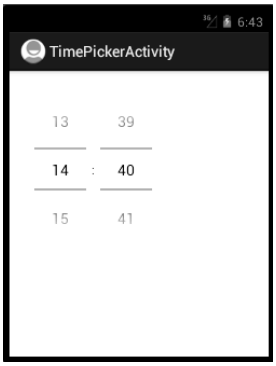

图 3.18　24 小时制 TimePicker

关键代码如下：

```
<!--设置 TimePicker 为 24 小时制-- >
timePicker.setIs24HourView(true);
<!--设置为 14 时-- >
timePicker.setCurrentHour(14);
<!--设置为 40 分-->
timePicker.setCurrentMinute(40);
```

3.7　小　　结

本章主要介绍了 Android 中一些常用的、比较简单的控件。其中，Button 类控件需要注册监听，实现具体功能，需要读者认真学习掌握。掌握这些控件的用法，并结合上一章的布局知识，就能够开发出简单的用户界面。

3.8　习　　题

1. 新建项目 EditText，在布局中添加两个 EditText。第一个显示为密码格式，在未输入密码时，显示文本“请输入密码”；第二个显示电话号码，输入电话号码时，界面弹出拨号盘。程序运行效果如图 3.19 所示。

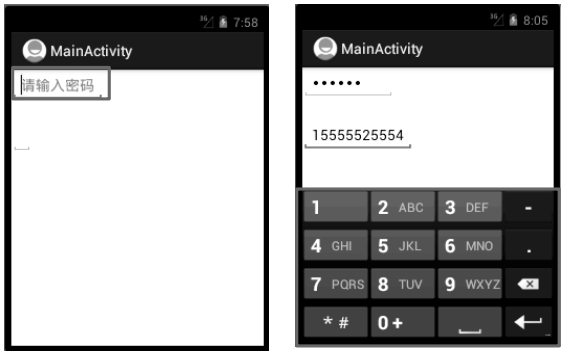

图 3.19　EditText

【分析】本题目主要考查读者对 EditText 的掌握。可以参考 3.2.2 节的开发程序。

【核心代码】本题的核心代码如下所示。

```
<EditText
    android:id="@+id/EditText1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:inputType="textPassword"
    android:hint="请输入密码">
</EditText>
<EditText
    android:id="@+id/EditText2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_below="@+id/EditText1"
    android:layout_marginTop="26dp"
    android:maxLength="11"
    android:inputType="phone"
    android:lines="1" />
```

2. 新建项目 Button，在布局中添加两个按钮。Button1 在代码中绑定监听，修改标题内容为“Button1 注册成功”；Button2 在布局中通过 onclick 属性绑定监听，修改标题内容为“Button2 注册成功”。程序运行效果如图 3.20 所示。

图 3.20　Button 注册监听

【分析】本题目主要考查读者对 Button 两种注册监听方式的掌握。可以参考 3.3.1 节的内容。

【核心代码】本题的核心代码如下所示。

布局文件代码：

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="button1"
    />
<Button
    android:id="@+id/button2"
```