

学习目标

本章主要介绍 Android UI 系统控件中文本控件 TextView、EditText, 按钮控件 Button、图片按钮控件 ImageButton、单选按钮 RadioButton、复选按钮 CheckBox、时间选择器 TimePicker、日期选择器 DatePicker、图片控件 ImageView、模拟时钟 AnalogClock 和数字时钟 DigitalClock。读者通过本章的学习, 能够深入了解 Android UI 系统控件, 掌握以下知识要点。

(1) 掌握文本控件 TextView、EditText 的类继承结构、属性及方法、应用。

(2) 掌握按钮控件 Button、图片按钮控件 ImageButton 的通常用法及属性。

(3) 掌握单选按钮 RadioButton、复选按钮 CheckBox 的常用方法、引用处理及属性方法设置。

(4) 掌握时间选择器 TimePicker、日期选择器 DatePicker 的使用方法及常用属性设置。

(5) 掌握图片控件 ImageView 的常用方法、引用处理及属性方法设置。

(6) 熟悉模拟时钟 AnalogClock 和数字时钟 DigitalClock 的常用方法。

Android 系统提供了许多控件给开发者使用, 开发者通过对这些控件编码及控件组合实现系统设计和功能开发。Android 系统的界面控件分为定制控件和系统控件, 系统控件是 Android 系统提供给用户已经封装的界面控件, 是在应用程序开发过程中的常见功能控件。系统控件更有利于帮助用户进行快速开发, 同时能够使开发者在使用 Android 系统进行开发的过程中, 保持应用程序的界面一致性。

在开发应用中,经常使用的系统控件有 TextView、EditText、Button、ImageButton、Checkbox、RadioButton、Spinner、ListView 和 TabHost 等。

Android UI 系统控件的使用,除了传统的程序代码中直接声明、创建之外,最能体现其设计思想的是,采用 XML 文件来描述控件,在 XML 中可以描述控件的宽度、长度、控件上的文本、控件的背景、控件的填充、设置源等。

5.1 文本控件简介

在 Android 系统中,文本控件包含 TextView 和 EditText 控件,它们都继承自 android.view.View,在 android.widget 包中。下面就文本控件的属性及使用方法进行详述。

5.1.1 文本框 TextView

android.widget 包中的 TextView 是文本表示控件,一般用来文本展示,是一种用于显示字符串的控件。主要功能是向用户展示文本的内容,可以作为应用程序的标签或者邮件正文的显示,默认情况下不允许用户直接编辑。

在程序设计和开发中,使用 TextView 可以采用的方式有两种,如下所示。

(1) 在程序中创建控件的对象来使用 TextView 控件。

如 TextView 控件,可以通过编写如下代码完成控件使用。

```
TextView tv=new TextView(this);
    tv.setText("大家好");
    setContentView(tv);
```

(2) 使用 XML 描述控件,并在程序中引用和使用。

① 在 res/layout 文件下的 XML 文件中描述控件。

```
<TextView
Android:id="@+id/text_view"
Android:layout_width="fill_parent"
Android:layout_height="wrap_content"
Android:textSize="16sp"
Android:padding="10dip"
Android:background="#00f0d0"
Android:text="大家好,这里是 TextView"/>
```

② 在程序中引用 XML 描述的 TextView。

```
TextView text_view = (TextView) findViewById(R.id.text_view);
```

上述两种方式的使用,各有优缺点,根据不同的需要,采用相应的方法。相比而言,采用第二种方法更好,其主要优势有:一是方便代码的维护;二是编码灵活;三是利于分工协作。

TextView 控件常用方法有 getText()和 setText()。TextView 控件有着与之相应的属性,通过选择不同的属性,给予其值,能够实现不同的效果。TextView 控件属性的设置

既可以在 XML 文件中通过属性名称设定赋值,也可以采用对应的方法,在程序代码中设定。其常用属性及对应方法如表 5-1 所示。

表 5-1 TextView 控件常用 XML 属性及对应方法

属性名称	对应方法	说明
android:text	setText(CharSequence)	设置 TextView 控件文字显示
android:autoLink	setAutoLinkMask(int)	设置是否当文本为 URL 链接/Email/电话号码/Map 时,文本显示为可点击的链接。可选值: none/web/email/phone/map/all
android:hint	setHint(int)	当 TextView 中显示的内容为空时,显示该文本
android:textColor	setTextColor(ColorStateList)	设置字体颜色
android:textSize	setTextSize(float)	设置字体大小
android:typeface	setTypeface(Typeface)	设置文本字体,必须是以下常量值之一: normal 0, sans 1, serif 2, monospace(等宽字体) 3
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
android:gravity	setGravity(int)	定义 TextView 在 x 轴和 y 轴方向上的显示方式
android:height	setHeight(int)	设置文本区域的高度,支持度量单位: px (像素)/dp/sp/in/mm(毫米)
android:minHeight	setMinHeight(int)	设置文本区域的最小高度
android:maxHeight	setMaxHeight(int)	设置文本区域的最大高度
android:width	setWidth(int)	设置文本区域的宽度,支持度量单位: px (像素)/dp/sp/in/mm(毫米)
android:minWidth	setMinWidth(int)	设置文本区域的最小宽度
android:maxWidth	setMaxWidth(int)	设置文本区域的最大宽度

5.1.2 TextView 应用案例

本节在上述 TextView 控件讲解的基础上,通过案例熟悉 TextView 控件的属性和用法,具体步骤如下。

(1) 创建一个新的 Android 工程,工程名为 TextViewDemo,目标 API 选择 28,应用程序名为 TextViewDemo,包名为 com.hisoft.activity,创建的 Activity 的名字为 TextViewActivity,最小 SDK 版本根据选择的目标 API 会自动添加为 17,创建后的项目工程如图 5-1 所示。

(2) 打开项目工程中 res→layout 目录下的 activity_main.xml 文件,设置线性布局,添加 4 个 TextView 控件,并设置属性,代码如下:

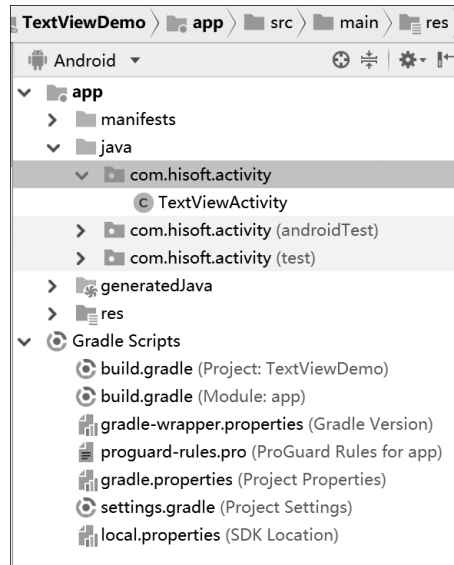


图 5-1 TextViewDemo 工程目录结构

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <?xml version="1.0" encoding="utf-8"?>
3. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
4.     android:orientation="vertical"
5.     android:layout_width="fill_parent"
6.     android:layout_height="fill_parent"
7.     >

8.     <TextView
9.         android:layout_width="fill_parent"
10.        android:layout_height="wrap_content"
11.        android:text="字体大小为 14 的文本"
12.        android:textSize="14pt"
13.    />

14.    <TextView
15.        android:layout_width="fill_parent"
16.        android:layout_height="wrap_content"
17.        android:singleLine="true"
18.        android:text="TextView 示例"
19.        android:ellipsize="middle"
20.    />

21.    <TextView
22.        android:layout_width="fill_parent"
23.        android:layout_height="wrap_content"

```

```

24.     android:singleLine="true"
25.     android:text="访问:http://www.zxxxx.cn"
26.     android:autoLink="web"
27.     />

28.     <TextView
29.         android:layout_width="fill_parent"
30.         android:layout_height="wrap_content"
31.         android:text="红色并带阴影的文本"
32.         android:shadowColor="#0000ff"
33.         android:shadowDx="15.0"
34.         android:shadowDy="20.0"
35.         android:shadowRadius="45.0"
36.         android:textColor="#ff0000"
37.         android:textSize="20pt"
38.     />

39. </LinearLayout>

```

(3) src 目录下 com.hisoft.activity 包下的 TextViewActivity.java 文件和 res→values 目录下 strings.xml 文件,都暂不做修改。部署运行项目工程,项目运行结果如图 5-2 所示。



图 5-2 TextViewDemo 运行结果

5.1.3 编辑框 EditText

EditText 控件继承自 android.widget.TextView,在 android.widget 包中。EditText 为输入框,是编辑文本控件,主要功能是让用户输入文本的内容。它是可以编辑的,是用来输入和编辑字符串的控件。

利用控件 EditText 不仅可以实现输入信息,还可以根据需要对输入信息进行限制约束。如:限制控件 EditText 输入信息。

```

<EditText
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"

```

```
android:inputType="numeber"/>
```

与上节所讲述的 TextView 一样,EditText 控件的使用也有两种:一种是在程序中创建控件的对象来使用 EditText 控件;另外一种是在 res/layout 文件下的 XML 文件中描述控件,程序中使用 EditText 控件。例如:

(1) 用 XML 描述一个 EditView。

```
<EditText Android:id="@+id/edit_text"
Android:layout_width="fill_parent"
Android:layout_height="wrap_content"
Android:text="这里可以输入文字" />
```

(2) 在程序中引用 XML 描述的 EditText。

```
EditText editText = (EditText) findViewById(R.id.editText);
```

EditText 常用方法: getText(),它也有与之相应的属性,通过选择不同的属性,给予其值,能够实现不同的效果。其常用属性及对应方法如表 5-2 所示。

表 5-2 EditText 控件常用 XML 属性及对应方法

属 性 名 称	对 应 方 法	说 明
android: hint		输入框的提示文字
android: password	setTransformationMethod (TransformationMethod)	设置文本框中的内容是否显示为密码,当为 true 时,以小点“.”显示文本
android: phoneNumber	setKeyListener(KeyListner)	设置文本框中的内容只能是电话号码,当为 true 时,表示电话框
android: digits	setKeyListener(KeyListener), 可使用此方法监听键盘来实现	设置允许输入哪些字符。如: 1234567890.+-*/%\n()
android: numeric	setKeyListener(KeyListener), 可使用此方法监听键盘来实现	设置只能输入数字,并且置顶可输入的数字格式,可选值有: Integer(正整数)、signed(整数,可带负号)、decimal(浮点数)
android: singleLine	setTransformationMethod (TransformationMethod)	设置文本框的单行模式
android: maxLenght	setFilters(InputFilter)	设置最大显示长度
android: cursorVisible	setCursorVisible(boolean)	设置光标是否可见,默认为可见
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: scrollHorizontally	setHorizontallyScrolling(boolean)	设置文本框是否可以水平滚动
android: selectAllOnFocus	setSelectAllOnFocus(boolean)	如果文本内容可选中,当文本框获得焦点时自动选中全部文本内容

续表

属 性 名 称	对 应 方 法	说 明
android: shadowColor	setShadowLayer(float,float, float,int)	为文本框设置指定颜色的阴影,需要与 shadowRadius 一起使用
android: shadowDx	setShadowLayer(float,float, float,int)	设置阴影横向坐标开始位置,为浮点数
android: shadowDy	setShadowLayer(float,float, float,int)	设置阴影纵向坐标开始位置,为浮点数
android: shadowRadius	setShadowLayer(float,float, float,int)	为文本框设置阴影的半径,为浮点数

5.1.4
 EditText 应用案例

在上述 EditText 控件讲解的基础上,通过案例熟悉 EditText 控件的属性和用法,具体步骤如下。

(1) 创建一个新的 Android 工程,工程名为 EditTextDemo,目标 API 选择 28,应用程序名为 EditTextDemo,包名为 com. hisoft. activity,创建的 Activity 的名字为 EditTextActivity,最小 SDK 版本根据选择的目标 API 会自动添加为 17,创建后的项目工程如图 5-3 所示。

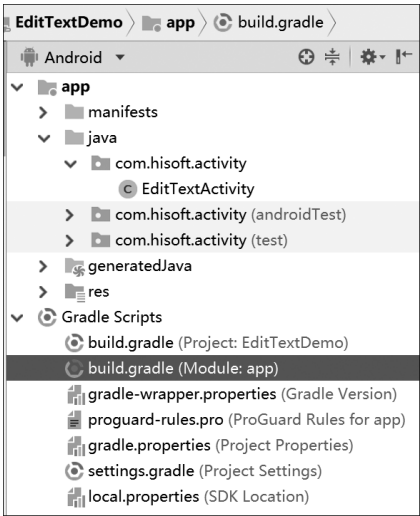


图 5-3 EditTextDemo 工程目录结构

(2) 打开项目工程中 res→layout 目录下 activity_main.xml 文件,设置线性布局,添加 1 个 TextView 控件和 1 个 EditText 控件,并设置相关属性,代码如下:

```

1.  <?xml version="1.0" encoding="utf-8"?>
2.  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.      android:orientation="vertical"
4.      android:layout_width="fill_parent"

```

```

5.      android:layout_height="fill_parent"
6.      >
7.      <TextView android:text="请输入:"
8.              android:id="@+id/textView1"
9.              android:layout_width="wrap_content"
10.             android:layout_height="wrap_content">
11.    </TextView>
12.    <EditText android:layout_height="wrap_content"
13.             android:layout_width="match_parent"
14.             android:id="@+id/editText1"
15.             android:hint="这里键入输入内容">
16.        <requestFocus></requestFocus>
17.    </EditText>

```

(3) src 目录下 com.hisoft.activity 包下的 EditTextActivity.java 文件和 res→values 目录下 strings.xml 文件,都暂不做修改。部署运行项目工程,项目运行结果如图 5-4 所示。



图 5-4 EditTextDemo 运行结果

5.2 按钮控件简介

5.2.1 按钮 Button

Button 是一种常用的按钮控件,继承自 android.widget.TextView,在 android.widget 包中。如图 5-5 所示,用户能够在该控件上点击,然后引发相应的事件处理函数。

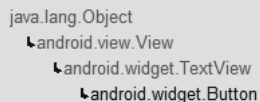


图 5-5 Button 类继承图

它的常用子类有 CheckBox、RadioButton、ToggleButton 等,在后续章节会讲到。

Button 按钮控件的通常用法如下。

在程序中通过 super.findViewById(id)得到在 layout 中 XML 文件中声明的 Button 的引用,然后使用 setOnClickListener (View.OnClickListener)添加监听,再在 View.OnClickListener 监听器中使用 v.equals (View)方法判断是哪一个按钮被按下,调用不同方法进行分别处理。具体如下。

(1) 用 XML 描述一个 Button。

```
<Button Android:id="@+id/button"
```

```

Android:layout_width="wrap_content"
Android:layout_height="wrap_content"
Android:text="这是一个 button" />

```

(2) 在程序代码中引用用 XML 描述的 Button。

```
Button button = (Button) findViewById(R.id.button);
```

(3) 给 Button 设置事件响应。

```
button.setOnClickListener(button_listener);
```

(4) 生成一个按钮事件监听器。

```

private Button.OnClickListener button_listener = new
Button.OnClickListener() {
public void onClick(View v) {
    switch(v.getId()) {
        case R.id.Button:
            textView.setText("Button 按钮 1");
            return;
        case R.id.Button01:
            textView.setText("Button 按钮 2");
            return;
    }
}
};

```

此外,也可以采用在 layout 中 XML 文件中声明分配一个方法给 Button,使用 android:onClick 属性,如:

```

1.      <Button
2.          android:layout_height="wrap_content"
3.          android:layout_width="wrap_content"
4.          android:text="@string/self_destruct"
5.          android:onClick="selfDestruct" />

```

当用户单击 Button 按钮时,Android 系统会自动调用 Activity 中的 selfDestruct (View)方法,但 selfDestruct(View)方法必须声明为 public,并只能接受 View 作为其唯一的参数。传递给这个方法的 View 是被点击的控件的一个引用,如下:

```

1.      public void selfDestruct(View view) {
2.          //Kabloey
3.      }

```

5.2.2 Button 应用案例

在开发应用中,Button 的使用较为常见,下面通过一个点击 Button 修改标题的案例来

介绍 Button 的应用。具体步骤如下。

(1) 创建一个新的 Android 工程,工程名为 ButtonDemo,目标 API 选择 28,应用程序名为 ButtonDemo,包名为 com.hisoft.activity,创建的 Activity 的名字为 ButtonActivity,最小 SDK 版本根据选择的目标 API 会自动添加为 17,创建后的项目工程如图 5-6 所示。

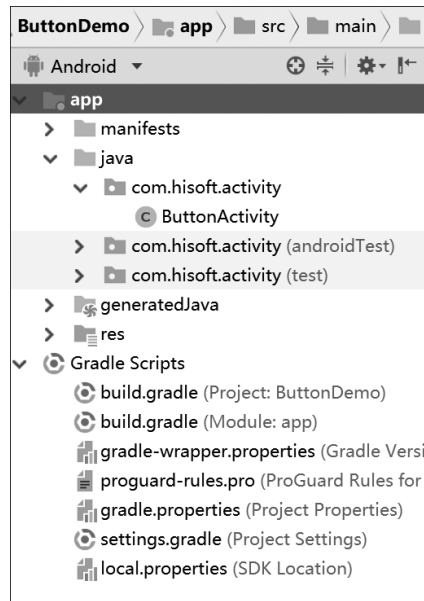


图 5-6 ButtonDemo 工程目录结构

(2) 打开项目工程中 res→layout 目录下 activity_main.xml 文件,设置线性布局,添加 1 个 Button 控件,并设置相关属性,代码如下所示:

```

1. <?xml version="1.0" encoding="utf-8"?>
2. <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.     android:orientation="vertical"
4.     android:layout_width="fill_parent"
5.     android:layout_height="fill_parent"
6.     >
7.     <Button android:text="按钮 1"
8.         android:id="@+id/button1"
9.         android:layout_width="wrap_content"
10.        android:layout_height="wrap_content">
11.     </Button>
12.</LinearLayout>
    
```

(3) 打开 src 目录下 com.hisoft.activity 包下的 ButtonActivity.java 文件,声明 Button,并获取引用,然后添加监听器,代码如下: