



Android Widget

Widget 也称微件,其最初的概念是 1998 年一个叫 Rose 的苹果计算机工程师提出的,直到 2003 年 Widget 在苹果操作系统上以小工具形式出现的时候才正式为人们所知,随后无数大公司都开始接受并应用这一思路。Widget 实际上是一个小型的应用程式,它可以是一个时钟,一个日记簿,一段视频,天气预报,一个 Flash 游戏等小程序。简单地说,Widget 可以理解为“应用小插件”,一种可供用户制作和自由下载的小工具,它包含娱乐、工作和学习等多种实用功能。目前的 Widget 应用大体可分为三种: Desktop Widget、Web Widget 以及 Mobile Widget。Desktop Widget 可以在计算机桌面上独立执行,用户无须通过浏览器便可连接到网络。时至今日,很多人已对苹果、雅虎、Google 及微软开发的 Desktop Widget 比较熟悉。通过这些小型应用软件,用户可把各类网上信息(如天气、新闻头条、图片等)放到桌面上。Mobile Widget 实际上是运行在移动设备上的 Desktop Widget。

为了提高开发效率,Android 系统提供有标准的开发类 Widget,这类 Widget 就是开发中常用的控件。虽然控件是一个拥有一定功能的 Widget,但控件不能独立运行,它需要嵌到应用程序中才可以运行。控件是对数据和方法的封装,控件可以有自己的属性和方法,属性是控件数据的简单访问者,方法是控件一些可见的功能。控件是 Android 用户界面中重要的组成元素之一,主要包括文本框、按钮、单选按钮、多选按钮等,能够实现人机交互,如显示文本、输入编辑文本、选择选项等。

5.1 TextView

TextView 控件用来向用户显示一行或多行的文本或者标签,它是不可编辑的。TextView 控件中包含很多属性,如文本的颜色和大小等属性,在 XML 文件中可以对属性设置相应的属性值,以达到相应的效果。TextView 控件常用属性如表 5-1 所示。

下面通过例 5-1 说明 TextView 控件。本例首先描述了一个线性布局,然后在布局中添加了一个 TextView 控件,用来显示文本。本例由两个文件组成,一个是用户界面 XML 文件 mylayout.xml,另一个是 JavaScript 文件 test.js。

表 5-1 TextView 常用属性和说明

属 性	说 明
android:gravity	设置控件中文字对齐方向,其值为 left、center 和 right
android:height	设置控件高度
android:width	设置控件宽度
android:hint	当控件显示内容为空时显示的文本(即提示信息)
android:textColorHint	设置提示信息文字的颜色,默认为灰色,与 hint 一起使用
android:text	设置控件上的文本内容
android:textColor	设置文本颜色
android:textSize	设置文本字体大小
android:typeface	设置文本字体,Android 系统默认支持三种字体,分别为:sans, serif, monospace,除此之外还可以使用其他字体文件(*.ttf)
android:background	设置控件背景颜色
android:textStyle	设置字体为粗体和斜体,取值为 bold(粗体)和 italic(斜体),值可以组合,组合时值之间要加 符号,例如,android:textStyle="bold italic"表示文本为粗斜体
android:lines	设置文本的行数,设置两行就显示两行,即使第二行没有数据
android:autoLink	当文本为 URL 链接/email/电话号码/map 时,设置是否文本显示为可单击的链接。可选值有 none/web/email/phone/map/all

【例 5-1】 (代码位置: \5\textview)

文件 mylayout.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    ><!--描述一个线性布局-->
    <TextView
        android:text="这是 TextView 控件,用来显示文本,文字居中对齐。"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="25px"<!--设置文本字体大小-->
        android:textColor="#fff00000"<!--设置文本颜色-->
        android:background="#00ff00"<!--设置控件背景颜色-->
        android:textStyle="bold"<!--设置字体为粗体-->
        android:gravity="center"
        android:height="200px"
```

```
</><!--在线性布局中添加一个 TextView 控件-->
</LinearLayout>
```

文件 test.js:

```
load("/sdcard/com.googlecode.rhinoforandroid/extras/rhino/android.js");
var droid=new Android();
var layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
droid.fullShow(layout);
droid.eventWait(10000);

function file_get_contents(fileName) {
    var file=new java.io.File(fileName);
    var reader=new java.io.BufferedReader(new java.io.FileReader(file));
    var tempString=null;
    var fileString="";
    //一次读入一行,直到读入 null 时文件结束
    while ((tempString=reader.readLine()) !=null) {
        fileString=fileString+tempString ;
    }
    reader.close();
    return fileString;
}
```

在文件 mylayout.xml 中,首先定义了一个垂直线性布局,然后在该布局中通过 TextView 标签添加了一个文本显示控件,通过属性 android:text 设置文本显示内容为“这是 TextView 控件,用来显示文本,文字居中对齐。”,通过属性 android:textSize 设置文本字体大小为“25px”,通过属性 android:textColor 设置文本字体颜色,通过属性 android:textStyle 设置文本字体为粗体。

程序运行之前,先把文件 mylayout.xml 和文件 test.js 复制到手机或模拟器中的 /sdcard/sl4a/scripts/ 目录,然后再运行 test.js,这时屏幕上会出现一个文本控件,显示文本内容为“这是 TextView 控件,用来显示文本,文字居中对齐。”。

5.2 EditText

用户使用手机编辑发送信息时,屏幕上会出现一个文本编辑框,用来输入、编辑信息。文本编辑框控件也就是 EditText 控件,它是可编辑的文本控件,用户可以在该控件中编辑要输入的内容,然后该控件接收用户的输入,并把用户的输入传输给 Android 应用处理,实现人机交互。EditText 控件是 Android 应用中非常重要的控件,应用非常广泛。例如,用户在登录 QQ 时,需要通过密码编辑框输入密码;用户在拨打电话时,需要通过文本编辑框输入电话号码等。

用户使用 EditText 控件时,输入的文本内容可以是单行文本,也可以是多行文本,可以是不同的类型,如数字、文字、时间日期、邮箱地址、密码等,这时控件的属性 android:inputType 显得尤其重要,在 XML 文件中可以通过设置属性 android:inputType 的属性值,指定 EditText 控件接收相应类型的文本,同时,EditText 控件还会让虚拟键盘来适应输入框中内容的类型。属性 android:inputType 的值及含义说明如表 5-2 所示。

表 5-2 android:inputType 值及含义

属 性 值	含义(文本类型)	属 性 值	含义(文本类型)
text	任何文本	textPassword	密码输入
number	数字	datetime	时间日期
numberDecimal	带小数点的数字	time	时间
phone	电话号码	date	日期
textEmailAddress	邮箱地址格式	textMultiLine	多行输入
textUri	网址	singleLine	单行输入

下面通过实例 5-2 描述了 EditText 控件。该例中通过属性 android:inputType="text"设置该控件可接受任何文本,没有格式类型约束。本例包含两个文件,一个是 XML 布局文件 mylayout.xml;另一个是 PHP 文件 test.php。

【例 5-2】 (代码位置: \5\edittext)

文件 mylayout.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff000000"
    android:gravity="center"
    ><!--描述一个线性布局-->
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20px"
    android:inputType="text"
    /><!--在线性布局中添加一个文本编辑控件-->
</LinearLayout>
```

文件 test.js:

```
load("/sdcard/com.googlecode.rhinoforandroid/extras/rhino/android.js");
var droid=new Android();
```

```

var layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
droid.fullShow(layout);
droid.eventWait(10000);

function file_get_contents(fileName) {
    var file=new java.io.File(fileName);
    var reader=new java.io.BufferedReader(new java.io.FileReader(file));
    var tempString=null;
    var fileString="";
    //一次读入一行,直到读入 null 为文件结束
    while ((tempString=reader.readLine()) !=null) {
        fileString=fileString+tempString ;
    }
    reader.close();
    return fileString;
}

```

程序运行之前,先把文件 mylayout.xml 和文件 test.php 复制到手机或模拟器中的 /sdcard/sl4a/scripts/ 目录,然后再运行 test.js,这时屏幕上会出现一个文本编辑框,用户可以在该编辑框中输入任何文本。

用户登录时,经常被要求在密码框中输入文本,这时,密码框将不显示文本内容而是用点表示文本内容。实例 5-3 描述了一个密码框控件,该例中通过属性 android:inputType="textPassword" 设置该控件输入文本的格式为密码输入。程序运行后,屏幕上会出现一个密码输入框,用户可以在该编辑框中输入密码。

【例 5-3】 (代码位置: \5\pwdtext)

文件 mylayout.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff000000"
    android:gravity="center"
><!-- 描述一个线性布局-->
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="20px"
    android:inputType="textPassword"
    /><!-- 在线性布局中添加一个密码框控件-->
</LinearLayout>

```

文件 test.php:

```
load("/sdcard/com.googlecode.rhinoforandroid/extras/rhino/android.js");
var droid=new Android();
var layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
droid.fullShow(layout);
droid.eventWait(10000);

function file_get_contents(fileName) {
    var file=new java.io.File(fileName);
    var reader=new java.io.BufferedReader(new java.io.FileReader(file));
    var tempString=null;
    var fileString="";
    // 一次读入一行,直到读入 null 为文件结束
    while ((tempString=reader.readLine()) !=null) {
        fileString=fileString+tempString ;
    }
    reader.close();
    return fileString;
}
```

用户在输入文本时,如果文本内容较多,文本框一行输满后,用户希望换行接着输入文本,这时需要多行文本框。例 5-4 描述了一个多行文本框控件,该例中通过属性 `android:inputType="textMultiLine"` 设置该控件可以多行输入文本,当文本框一行文本输满时会自动换行,文本框也接受回车换行符。程序运行后,屏幕上会出现一个多行文本框,用户可以在该文本框中进行多行输入文本。

【例 5-4】 (代码位置: \5\multilinetext)

文件 mylayout.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff000000"
    android:gravity="center"
    ><!-- 描述一个线性布局-->
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20px"
        android:inputType="textMultiLine"
        /><!-- 在线性布局中添加一个多行文本框-->
    </LinearLayout>
```

文件 test.js:

```
load("/sdcard/com.googlecode.rhinoforandroid/extras/rhino/android.js");
var droid=new Android();
var layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
droid.fullShow(layout);
droid.eventWait(10000);

function file_get_contents(fileName) {
    var file=new java.io.File(fileName);
    var reader=new java.io.BufferedReader(new java.io.FileReader(file));
    var tempString=null;
    var fileString="";
    // 一次读入一行,直到读入 null 为文件结束
    while ((tempString=reader.readLine()) !=null) {
        fileString=fileString+tempString ;
    }
    reader.close();
    return fileString;
}
```

用户拨打电话,在文本框中输入信息时,为了避免误输入,希望系统只提供拨打电话时所涉及的符号,这时需要电话格式的文本框。在文本框中输入信息时,系统会弹出软键盘供用户输入,软键盘只提供拨打电话时所涉及的符号。需要指出的是,还有其他格式的文本框也会提供软键盘,但系统会根据格式提供不同的软键盘,例如, textUri 和 Date 格式的文本框会提供网址和日期格式的软键盘。

实例 5-5 描述了一个电话格式文本框控件,用来输入电话号码,该例中通过属性 android:inputType="phone"设置该控件为电话格式文本框。程序运行后,屏幕会出现一个软键盘,这个软键盘只提供拨打电话时所涉及的符号,用户可以在编辑框中输入电话号码。

【例 5-5】 (代码位置: \5\phonetext)

文件 mylayout.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#ff000000"
    android:gravity="center"
><!--描述一个线性布局-->
<EditText
```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20px"
        android:inputType="phone"
        android:text="987654321"
    /><!--在线性布局中添加一个电话格式文本框-->
</LinearLayout>

```

文件 test.js:

```

load("/sdcard/com.googlecode.rhinoforandroid/extras/rhino/android.js");
var droid=new Android();
var layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
droid.fullShow(layout);
droid.eventWait(10000);

function file_get_contents(fileName) {
    var file=new java.io.File(fileName);
    var reader=new java.io.BufferedReader(new java.io.FileReader(file));
    var tempString=null;
    var fileString="";
    // 一次读入一行,直到读入 null 为文件结束
    while ((tempString=reader.readLine()) !=null) {
        fileString=fileString+tempString ;
    }
    reader.close();
    return fileString;
}

```

5.3 Button

用户在进行登录时,界面上会有一个含有文本“登录”的 Button 控件,单击该控件后可以实现登录。Button 是按钮控件,是 Android 界面中最常用的控件之一,用户可以单击它产生事件,实现各种操作,例如,可以点击按钮实现信息的确认等操作。添加 Button 控件时,需要给按钮控件定义一个 ID 号,并且可以通过控件相应的属性值来设置控件的大小、颜色、文本的内容、字体等。

下面通过例 5-6 说明 Button 控件。本例首先描述了一个线性布局,然后在布局中添加了一个 Button 控件。

【例 5-6】 (代码位置: \5\button)

文件 mylayout.xml:

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    ><!--描述一个线性布局-->
<Button
    android:text="单击按钮 "
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/mybutton"
    android:textStyle="bold"
    android:textSize="20px"
    ><!--在线性布局中添加一个 Button 控件-->
</Button>
</LinearLayout>

```

文件 test.js:

```

load("/sdcard/com.googlecode.rhinoforandroid/extras/rhino/android.js");
var droid=new Android();
var layout=file_get_contents("/sdcard/sl4a/scripts/mylayout.xml");
droid.fullShow(layout);
droid.eventWait(10000);

function file_get_contents(fileName) {
    var file=new java.io.File(fileName);
    var reader=new java.io.BufferedReader(new java.io.FileReader(file));
    var tempString=null;
    var fileString="";
    // 一次读入一行,直到读入 null 为文件结束
    while ((tempString=reader.readLine()) !=null) {
        fileString=fileString+tempString ;
    }
    reader.close();
    return fileString;
}

```

在文件 mylayout.xml 中,首先定义了一个垂直线性布局,然后在该布局中通过 Button 标签添加了一个按钮控件,文本内容为“单击按钮”,id 为“mybutton”。

程序运行之前,先把文件 mylayout.xml 和文件 test.js 复制到手机或模拟器中的 /sdcard/sl4a/scripts/ 目录,然后再运行 test.js,屏幕上会出现一个按钮,显示文本为“单

击按钮”。

5.4 ImageButton

ImageButton 控件是图片按钮控件,是 Android 提供的另一种按钮控件,与 Button 控件不同的是,ImageButton 控件可以在按钮上加载一个图片,从而控件上显示的是图片。ImageButton 控件能实现自定义风格,使按钮外观更美观。

Android 界面中,ImageButton 可以引用图像资源、字符串和图标等不同类型的资源,引用资源的格式如下。

@ 包名:资源类型/资源名

ImageButton 以文件 URL 的形式引用图片资源,支持的图像文件格式是 png 和 jpg 格式。

ImageButton 引用图片的来源,可以是自定义的图片文件资源,也可以是 Android 系统提供的标准图片资源,这些标准资源是所有应用程序共享的,关于可用系统资源的完整列表,请参阅 <http://developer.android.com/reference/android/R.html>。

ImageButton 可以通过 android:src 属性来定义引用的图片,它有两种定义格式,具体如表 5-3 所示。

表 5-3 android:src 格式及含义

格 式	含 义
android:src="@android:drawable/资源名"	引用系统提供的标准图片资源,例如,android:src="@android:drawable/stat_sys_phone_call_on_hold"
android:src="file:///目录和文件名"	引用图像文件资源,例如,android:src="file:///sdcard/download/panda72.png"

下面通过实例说明 ImageButton 控件。该按钮上显示的图片引用了系统提供的标准图片资源。

【例 5-7】 (代码位置: \5\imagebutton1)

文件 mylayout.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="#000000"
    ><!--描述一个线性布局-->
<ImageButton
```