3.1.用XML为布局添加控件

Android 提倡用 XML 的代码方式开发 UI，辅之以 Java 代码。用 Java 代码创建 UI，因其具有灵活性也较常见。后面的 UI 开发主要用 XML 代码方式。

在 XML 中添加控件如图-2 所示，建议每行只写一个 XML 命令。

3.1.1.TextView控件

3.1.1.1.概述

TextView 是一个标签控件，用于显示提示信息。

TextView 是 ViewGroup 的子类，该类是 EditText 和 Button 的父类。

3.1.1.2.常用属性

属性名	Java 代码	说明
layout_height	setHeight(int)	设置文本框高度
layout_width	setWidth(int)	设置文本框宽度
hint	setHint(int)或 setHint(Charsequence)	设置提示的字符串
text	setText(String)	标签显示的文本信息
cursorVisible	setCursorVisible(boolean)	光标是否可见 true:可见 false:不可见
editable		设置是否允许编辑
gravity	setGravity(int)	设置文本框内文本的对齐方式
layout_gravity		设置标签在父容器中的对齐方式
visibility	setVisibility(int)	是否可见(visible/invisible)
textColor	setTextColor	设置文字的颜色
textSize	setTextSize	设置文字大小
background		设置背景图

图-1

3.1.2.EditText控件

3.1.2.1 概述

EditText 是文本编辑框控件，该控件继承自 TextView 类，常用来进行文本输入、编辑。

3.1.2.2 常用属性

EditText 是 TextView 类的子类，因此上表中的属性 EditText 同样具有。

3.1.2.3.常用方法

String　getText().toString();
　　作用：获取控件的 text 属性的值。
void setError(String message);
　　作用：显示错误提示信息。

3.3.Button控件

3.1.3.1.概述

　　Button 控件的父类是 TextView 控件。作用是接收并响应单击事件。
　　Button 控件可显示文字，并且可将图片设置为背景。

3.1.3.2.常用属性

　　Button 是 TextView 类的子类，因此上表中的属性 EditText 同样具有。

属性名	说明
drawableTop	位于文字之上的图片
drawableLeft	位于文字左边的图片
drawableRight	位于文字右边的图片
drawableBottom	位于文字右边的图片

3.1.4.ImageView控件

3.1.4.1.概述

　　ImageView 是 View 类的子类，该控件用于显示图像（包括动画）。

3.1.4.2 常用属性

XML 属性	相关方法	说明
src	setImageResource(int)	设置 ImageView 所显示的图片

图-2

3.1.5.控件的相关说明

3.1.5.1.textColor

文本颜色，#+8 位十六进制数，前两位表示 Alpha（透明度），3、4 位：红色值，5、6 位：绿色，7、8 位：蓝色。

例如：#ffffff：白色，#000000：黑色

3.1.5.2.textSize

设置文字大小，单位可以是 px（像素值），也可以是 sp

3.1.5.3.dp、px和sp

dip: device independent pixels(设备独立像素). 不同设备有不同的显示效果,这个和设备硬件有关，为了支持 WVGA、HVGA 和 QVGA 推荐使用 dip，不依赖像素。

px: pixels(像素). 不同设备显示效果相同，HVGA 代表 320x480 像素，这个用得比较多。

sp: scaled pixels(与刻度无关的像素). 主要用于字体显示。

(4)关于 dip 和 sp

过去常以像素为单位设计用户界面。这样处理的问题在于，如果在一个每英寸点数（dip）更高分辨率的显示器上运行该程序，则用户界面会显得很小。在有些情况下，用户界面可能会小到难以看清内容。

与分辨率无关的度量单位可以解决这一问题。Android 支持下列所有单位。

px（像素）：屏幕上的点。

in（英寸）：长度单位。

mm（毫米）：长度单位。

pt（磅）：1/72 英寸。

dp（与密度无关的像素）：一种基于屏幕密度的抽象单位。在每英寸 160 点(标准分辨率)的显示器上，1dp = 1px。

在每英寸 240 点的显示器(高分辨率)上，1dp=240/160=1.5px

在每英寸 120 点的显示器上(低分辨率)上，1dp=120/160=0.75px

dip：与 dp 相同,sp（与刻度无关的像素）：与 dp 类似，但是可以根据用户的字体大小首选项进行缩放。

为了使用户界面能够在现在和将来的显示器类型上正常显示，建议始终使用 sp 作为文字大小的单位，将 dip 作为图形尺寸的单位。

3.1.5.4.设置控件的ID值

命令格式：@+id/ID 值，示例：android:id=”@+id/tv”

3.1.5.5.控件的宽度和高度

几乎每一个 UI 组件都必须设置宽和高。

3.2.构建Android界面对应的Java代码

3.2.1.设置布局文件为窗口的显示内容

【示例代码】setContentView(R.id.main);
说明：R.id.main 是 R.java 文件中定义的索引值，通过该索引值在 res/layout 文件夹下查找到 main.xml 文件。

3.2.2.通过查找文件中的ID值实例化控件

【示例代码】TextView tv=(TextView)findViewById(R.id.tv);
说明：R.id.tv 是 R.java 中定义的控件的索引值，通过该索引值将在 main.xml 中查找到 id 值是 tv 的控件，并以该控件来实例化 tv 所引用的对象。

3.2.3.getText()：获取控件中的字符串

示例代码：String text=tv.getText().toString();*//将 tv 控件中的 text 属性的值赋值给 text 变量*

Log.d 命令的作用是用于调试程序，Android 开发推荐使用 Log.d，而不推荐使用控制台输出命令，主要原因是信息输出在日志窗口能保留调试的信息而不丢失。

操作步骤如下

步骤 1、Log.d（“UI”,” Hello”);

提示：以上命令在输入至 Log 时，按 Alt+/并按回车（目的是导包），再输入后续命令。

步骤 2、按下图所示操作，打开日志视图：

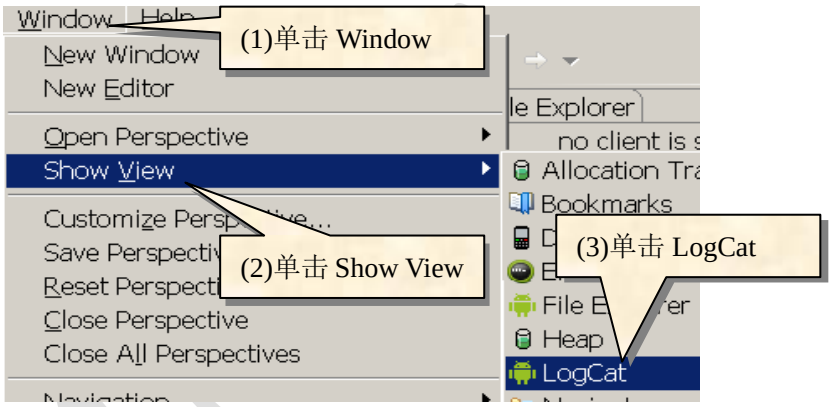

图-3

步骤 3、按下图所示操作，在日志窗口中增加一个过滤标签，该标签只显示字符串是”UI”条件的 Log.d 输出的信息。

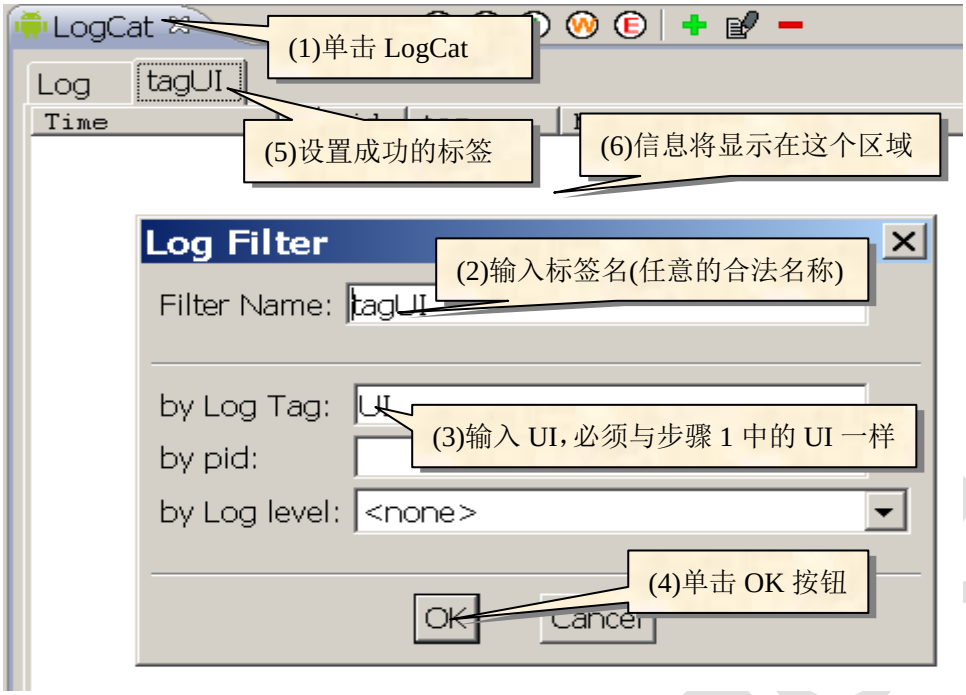

图-4

运行当前项目，则 Hello 将显示在上图中(6)所指向的区域中。

3.4 线性布局

3.4.1.概述

UI 是 User Interface(用户界面)的简称，在 Android 应用开发中，UI 设计是非常重要的，良好的 UI 设计会给用户一个惊喜，容易得到用户的认可。

Android SDK 为 UI 设计提供了强力的支持，SDK 中包含了丰富的控件，还允许自定义控件实现特殊效果。

3.4.2.布局管理器

Android 提供了布局管理器，用来管理用户界面中的各个控件，布局管理器本身也是控件，但该控件也是一个容器控件，可存放其它的控件。

布局管理器是 ViewGroup 的子类，还是 View 类的间接子类，图-5 是布局管理器的类图树：

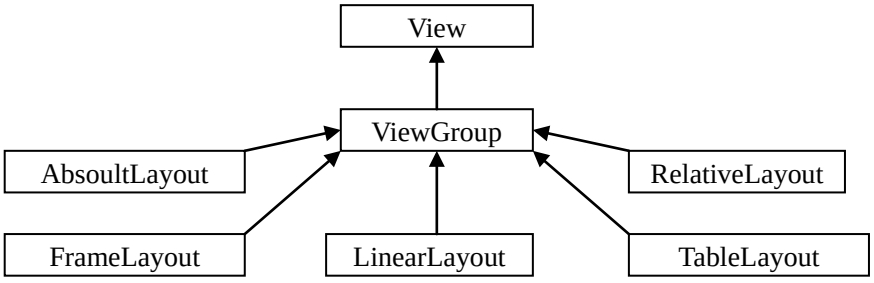

图-5

其中：

●LinearLayout：线性布局
●RelativeLayout；相对布局
●TableLayout：表格布局
●FrameLayout：帧布局
●Absolayout：绝对布局

3.4.3.LinearLayout布局

LinearLayout(线性布局)本身比较简单，该布局中的控件出现的先后顺序依次排列，线性布局有两种排列方式（水平排列和垂直排列，默认水平排列控件）。

2、图 2 列出 LinearLayout 中常用的 XML 属性和方法：

gravity	设置布局管理器内控件的对齐方式。支持：top、bottom、left、right、center_vertical、center_horizontal、fill_horizontal、fill、clip_horizontal 属性值，并支持组合属性
Layout_gravity	设置在父容器中的对齐方式，属性值与 gravity 属性相同
orientation	设置布局管理器控件的排列方式，有 Vertical(垂直)和 horizontal（水平）两种。
marginLeft	与左边控件的居间
marginTop	与上边控件的间距
marginBottom	与下边控件的间距
marginRight	与右边控件的间距