

《Android 开发从入门到实战》

简单控件

让人人都能享受到高品质的教育服务

简单控件



文本显示



视图基础



常用布局



按钮触控



图像显示

设置文本的内容

- 设置文本内容有两种方式：
 - 在 XML 文件中通过属性 `android:text` 设置文本
 - 在 Java 代码中调用文本视图对象的 `setText` 方法设置文本

引用字符串资源

- 在XML文件中引用 (`@string/***`)
- 在Java代码中引用 (`R.string.***`)

设置文本的大小

- 在 Java 代码中调用 `setTextSize` 方法，即可指定文本大小。
- 在 XML 文件中则通过属性 `android:textSize` 指定文本大小，此时需要指定字号单位。
 - px: 它是手机屏幕的最小显示单位，与设备的显示屏有关。
 - dp: 它是与设备无关的显示单位，只与屏幕的尺寸有关。
 - sp: 它专门用来设置字体大小，在系统设置中可以调整字体大小。



设置文本的颜色

➤ 在 Java 代码中调用 `setTextColor` 方法即可设置文本颜色，具体色值可从 `Color` 类取。

Color类中的颜色类型	说明	Color类中的颜色类型	说明
BLACK	黑色	GREEN	绿色
DKGRAY	深灰	BLUE	蓝色
GRAY	灰色	YELLOW	黄色
LTGRAY	浅灰	CYAN	青色
WHITE	白色	MAGENTA	玫红
RED	红色	TRANSPARENT	透明

RGB颜色定义

- 在XML文件中则通过属性android:textColor指定文本颜色，色值由透明度alpha和RGB三原色（红色red、绿色green、蓝色blue）联合定义。
- 色值有**八位十六进制数**与**六位十六进制数**两种表达方式，例如八位编码FFEEDDCC中，FF表示透明度，EE表示红色的浓度，DD表示绿色的浓度，CC表示蓝色的浓度。
- 透明度为FF表示完全不透明，为00表示完全透明。RGB三色的数值越大，表示颜色越浓，也就越亮；数值越小，表示颜色越淡，也就越暗。

使用色值定义文字颜色

- 在Java代码中设置色值需要添加0x前缀表示十六进制数。
- 在XML文件中设置色值需要添加 “#” 前缀。



引用颜色资源

- 在XML文件中引用（@color/***）。
- 在Java代码中引用（R.color.***）。

设置视图的宽高

- 视图宽度通过属性`android:layout_width`表达，视图高度通过属性`android:layout_height`表达，宽高的取值主要有列三种：
- `match_parent`：表示与上级视图保持一致。
 - `wrap_content`：表示与内容自适应。
 - 以dp为单位的具体尺寸。

在代码中设置视图宽高

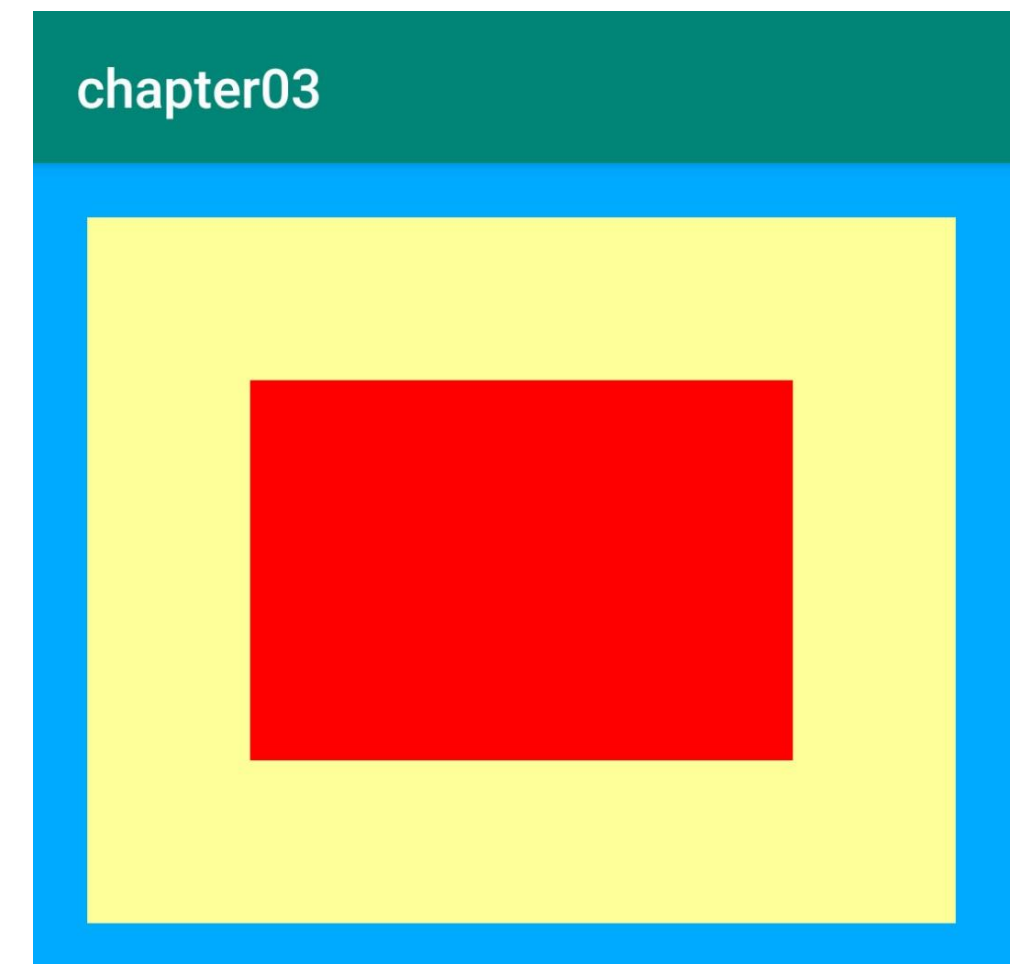
➤ 首先确保XML中的宽高属性值为wrap_content，接着打开该页面对应的Java代码，依序执行以下三个步骤：

- 调用控件对象的getLayoutParams方法，获取该控件的布局参数。
- 布局参数的width属性表示宽度，height属性表示高度，修改这两个属性值。
- 调用控件对象的setLayoutParams方法，填入修改后的布局参数使之生效。

设置视图的间距

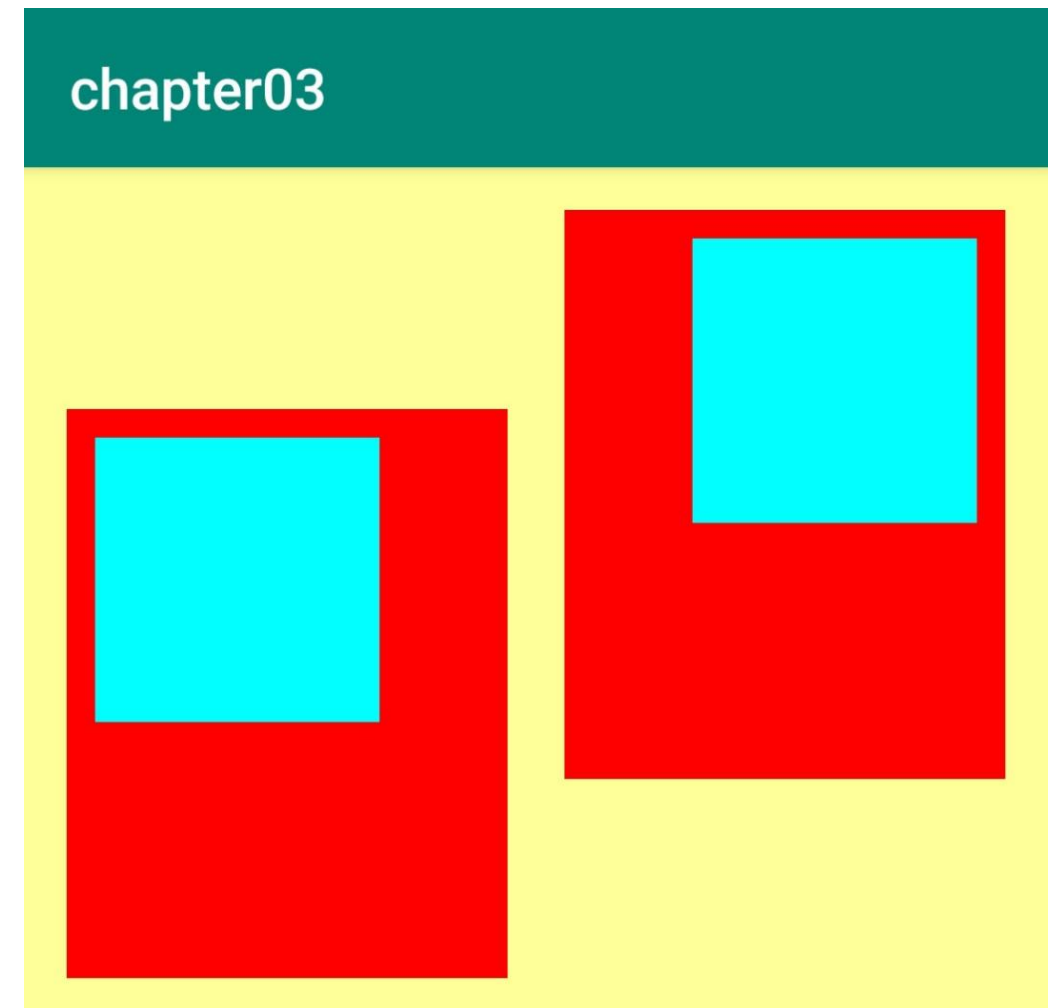
➤ 设置视图的间距有两种方式：

- 采用`layout_margin`属性，它指定了当前视图与周围同级视图之间的距离。包括`layout_margin`、`layout_marginLeft`、`layout_marginTop`、`layout_marginRight`、`layout_marginBottom`
- 采用`padding`属性，它指定了当前视图与内部下级视图之间的距离。包括`padding`、`paddingLeft`、`paddingTop`、`paddingRight`、`paddingBottom`



设置视图的对齐方式

- 设置视图的对齐方式有两种途径：
 - 采用layout_gravity属性，它指定了当前视图相对于上级视图的对齐方式。
 - 采用gravity属性，它指定了下级视图相对于当前视图的对齐方式。
- layout_gravity与gravity的取值包括：left、top、right、bottom，还可以用竖线连接各取值，例如“left|top”表示即靠左又靠上，也就是朝左上角对齐。



线性布局LinearLayout

- 线性布局内部的各视图有两种排列方式：
 - orientation属性值为horizontal时，内部视图在**水平方向**从左往右排列。
 - orientation属性值为vertical时，内部视图在**垂直方向**从上往下排列。
- 如果不指定orientation属性，则LinearLayout默认水平方向排列。

线性布局的权重

- 线性布局的权重概念，指的是线性布局的下级视图各自拥有多大比例的宽高。
- 权重属性名叫`layout_weight`，但该属性不在`LinearLayout`节点设置，而在线性布局的直接下级视图设置，表示该下级视图占据的宽高比例。
 - `layout_width`填`0dp`时，`layout_weight`表示水平方向的宽度比例。
 - `layout_height`填`0dp`时，`layout_weight`表示垂直方向的高度比例。



相对布局RelativeLayout

- 相对布局的下级视图位置由其他视图决定。用于确定下级视图位置的参照物分两种：
 - 与该视图自身平级的视图；
 - 该视图的上级视图（也就是它归属的RelativeLayout）
- 如果不设定下级视图的参照物，那么下级视图默认显示在RelativeLayout内部的左上角。

相对位置的取值

相对位置的属性取值	相对位置说明
layout_toLeftOf	当前视图在指定视图的左边
layout_toRightOf	当前视图在指定视图的右边
layout_above	当前视图在指定视图的上方
layout_below	当前视图在指定视图的下方
layout_alignLeft	当前视图与指定视图的左侧对齐
layout_alignRight	当前视图与指定视图的右侧对齐
layout_alignTop	当前视图与指定视图的顶部对齐
layout_alignBottom	当前视图与指定视图的底部对齐
layout_centerInParent	当前视图在上级视图中间
layout_centerHorizontal	当前视图在上级视图的水平方向居中
layout_centerVertical	当前视图在上级视图的垂直方向居中
layout_alignParentLeft	当前视图与上级视图的左侧对齐
layout_alignParentRight	当前视图与上级视图的右侧对齐
layout_alignParentTop	当前视图与上级视图的顶部对齐
layout_alignParentBottom	当前视图与上级视图的底部对齐

网格布局GridLayout

- 网格布局支持多行多列的表格排列。
- 网格布局默认从左往右、从上到下排列，它新增了两个属性：
 - **columnCount**属性，它指定了网格的列数，即每行能放多少个视图；
 - **rowCount**属性，它指定了网格的行数，即每列能放多少个视图；



滚动视图ScrollView

➤ 滚动视图有两种：

- **ScrollView**，它是垂直方向的滚动视图；垂直方向滚动时，`layout_width`属性值设置为`match_parent`，`layout_height`属性值设置为`wrap_content`。
- **HorizontalScrollView**，它是水平方向的滚动视图；水平方向滚动时，`layout_width`属性值设置为`wrap_content`，`layout_height`属性值设置为`match_parent`。

按钮控件Button

- 按钮控件Button由TextView派生而来，它们之间的区别有：
 - Button拥有默认的按钮背景，而TextView默认无背景；
 - Button的内部文本默认居中对齐，而TextView的内部文本默认靠左对齐；
 - Button会默认将英文字母转为大写，而TextView保持原始的英文大小写；

按钮控件的新增属性

- 与TextView相比，Button增加了两个新属性：
 - **textAllCaps**属性，它指定了是否将英文字母转为大写，为true是表示自动转为大写，为false表示不做大写转换。
 - **onClick**属性，它用来接管用户的点击动作，指定了点击按钮时要触发哪个方法；

点击事件和长按事件

- 监听器，意思是专门监听控件的动作行为。只有控件发生了指定的动作，监听器才会触发开关去执行对应的代码逻辑。
- 按钮控件有两种常用的监听器：
 - 点击监听器，通过setOnClickListener方法设置。按钮被按住少于500毫秒时，会触发点击事件。
 - 长按监听器，通过setOnLongClickListener方法设置。按钮被按住超过500毫秒时，会触发长按事件。

禁用与恢复按钮

- 在实际业务中，按钮通常拥有两种状态，即不可用状态与可用状态，它们在外观和功能上的区别如下：
 - 不可用按钮：按钮不允许点击，即使点击也没反应，同时按钮文字为灰色；
 - 可用按钮：按钮允许点击，点击按钮会触发点击事件，同时按钮文字为正常的黑色；
- 是否允许点击由enabled属性控制，属性值为true时表示允许点击，为false时表示不允许点击。

图像视图ImageView

➤ 图像视图展示的图片通常位于res/drawable***目录，设置图像视图的显示图片有两种方式：

- 在XML文件中，通过属性android:src设置图片资源，属性值格式形如 “@drawable/不含扩展名的图片名称”。
- 在Java代码中，调用setImageResource方法设置图片资源，方法参数格式形如 “R.drawable.不含扩展名的图片名称”。

图像视图的缩放类型

➤ ImageView本身默认图片居中显示，若要改变图片的显示方式，可通过scaleType属性设定，该属性的取值说明如下：

XML中的缩放类型	ScaleType类中的缩放类型	说明
fitXY	FIT_XY	拉伸图片使其正好填满视图（图片可能被拉伸变形）
fitStart	FIT_START	保持宽高比例，拉伸图片使其位于视图上方或左侧
fitCenter	FIT_CENTER	保持宽高比例，拉伸图片使其位于视图中间
fitEnd	FIT_END	保持宽高比例，拉伸图片使其位于视图下方或右侧
center	CENTER	保持图片原尺寸，并使其位于视图中间
centerCrop	CENTER_CROP	拉伸图片使其充满视图，并位于视图中间
centerInside	CENTER_INSIDE	保持宽高比例，缩小图片使之位于视图中间（只缩小不放大）

缩放类型的效果

fitCenter



chapter03

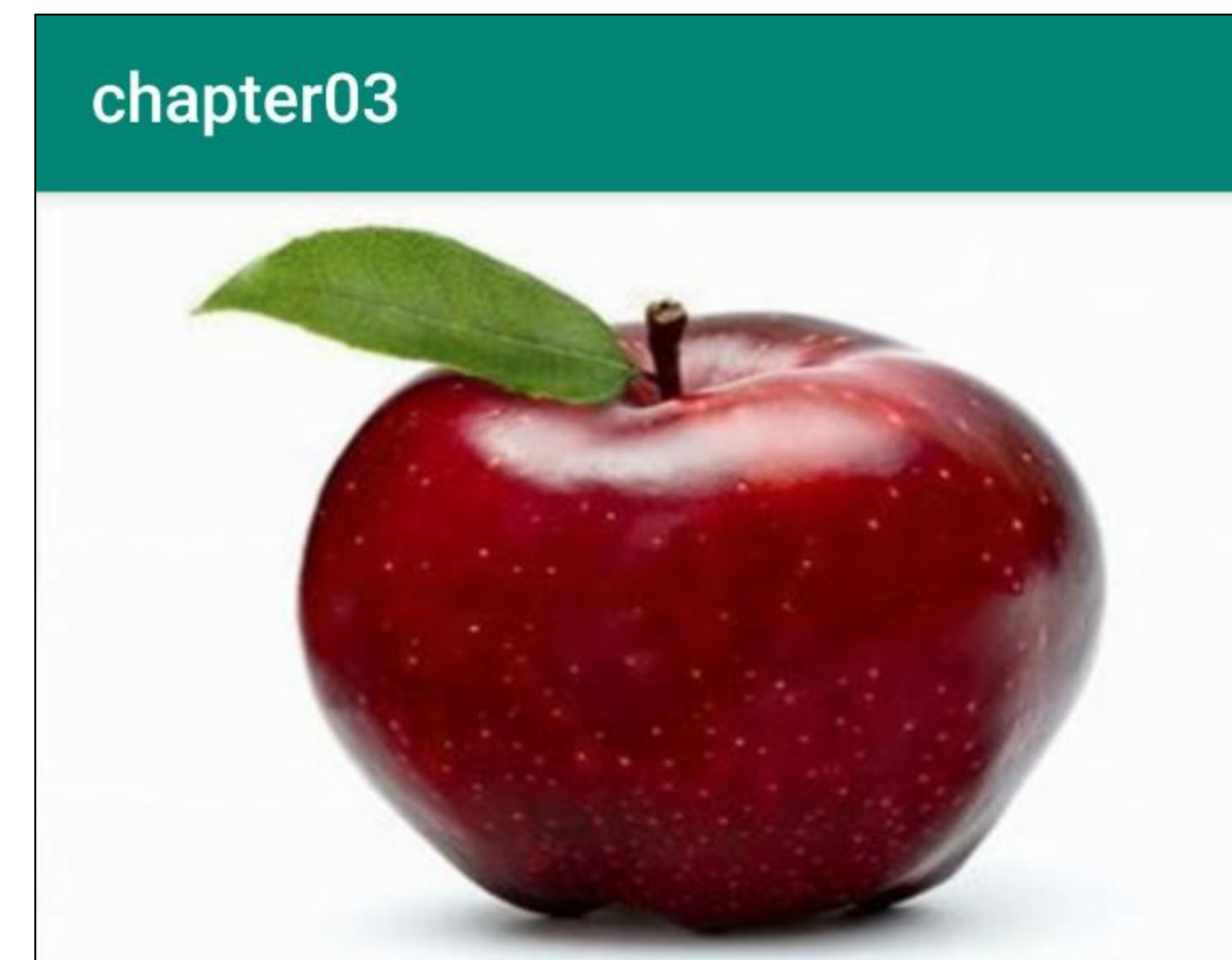


centerCrop

centerInside



chapter03



fitXY

图像按钮ImageButton

- ImageButton是显示图片的图像按钮，但它继承自ImageView，而非继承Button。
- ImageButton和Button之间的区别有：
 - Button既可显示文本也可显示图片，ImageButton只能显示图片不能显示文本。
 - ImageButton上的图像可按比例缩放，而Button通过背景设置的图像会拉伸变形。
 - Button只能靠背景显示一张图片，而ImageButton可分别在前景和背景显示图片，从而实现两张图片叠加的效果。

ImageButton的使用场合

- 在某些场合，有的字符无法由输入法打出来，或者某些文字以特殊字体展示，就适合适合先切图再放到ImageButton。例如：开平方符号 $\sqrt{\quad}$ ，等等。
- ImageButton与ImageView之间的区别有：
 - ImageButton有默认的按钮背景，ImageView默认无背景。
 - ImageButton默认的缩放类型为center，而ImageView默认的缩放类型为fitCenter。

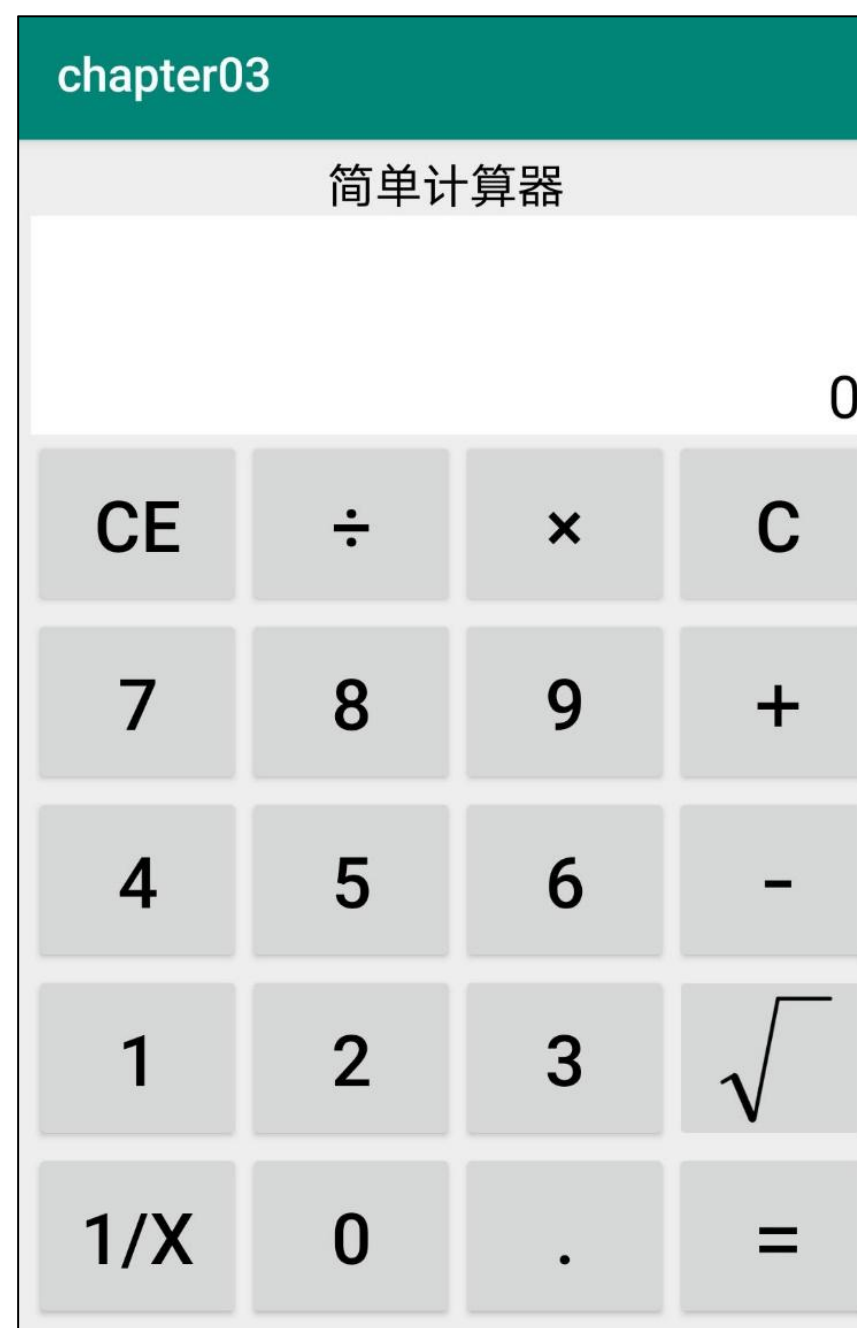
同时展示文本与图像

- 同时展示文本与图像的可能途径包括：
- （1）利用LinearLayout对ImageView和TextView组合布局。
- （2）通过按钮控件Button的drawable***属性设置文本周围的图标。
 - drawableTop: 指定文字上方的图片。
 - drawableBottom: 指定文字下方的图片。
 - drawableLeft: 指定文字左边的图片。
 - drawableRight: 指定文字右边的图片。
 - drawablePadding: 指定图片与文字的间距。



实战项目：计算器

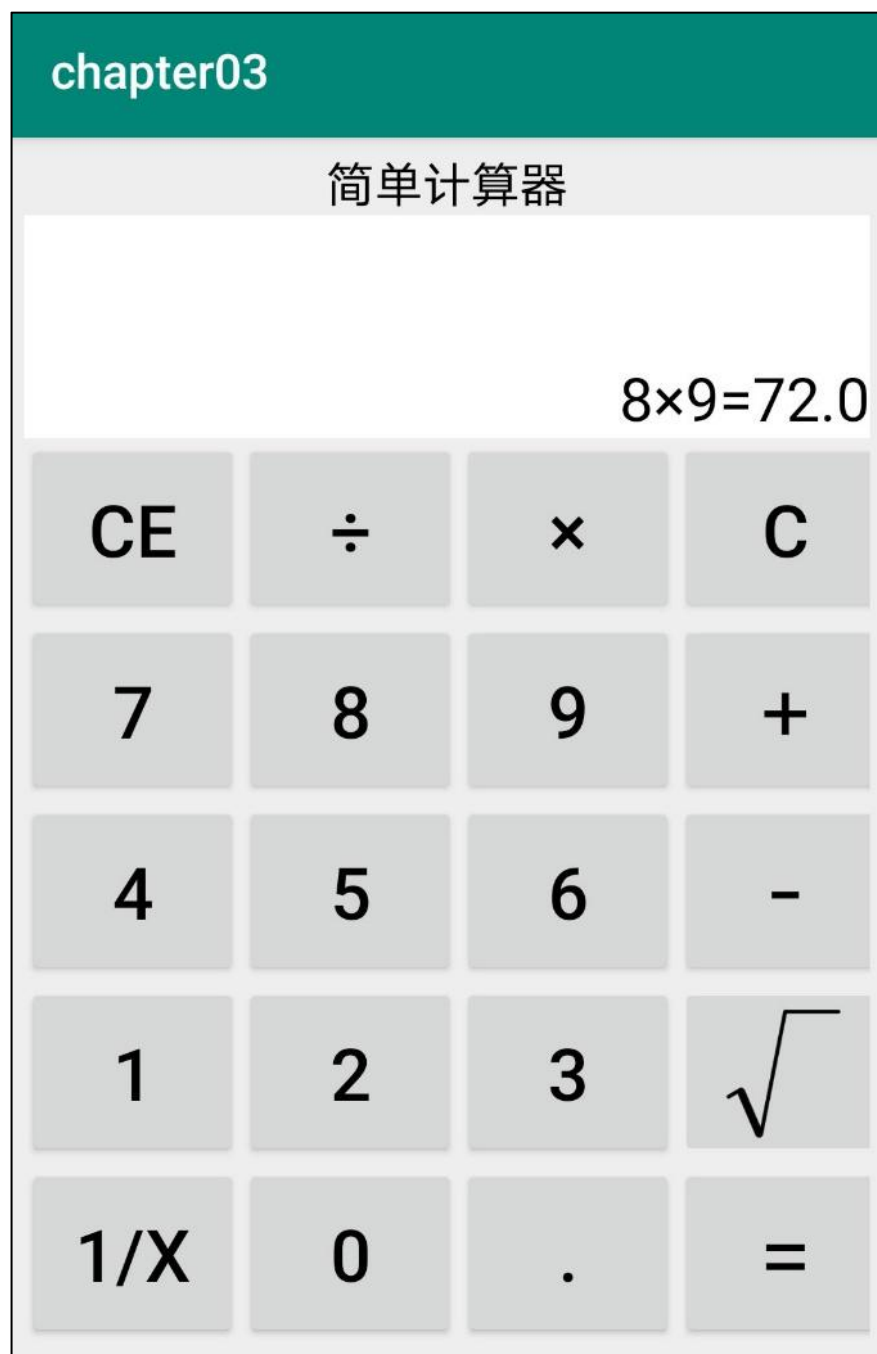
- 计算器的界面分为两大部分，第一部分是上方的计算表达式，既包括用户的按键输入，也包括计算结果数字；第二部分是下方的各个按键，例如：从0到9的数字按钮、加减乘除与等号、正负号按钮、小数点按钮、求倒数按钮、平方按钮、开方按钮，以及退格、清空、取消等控制按钮。



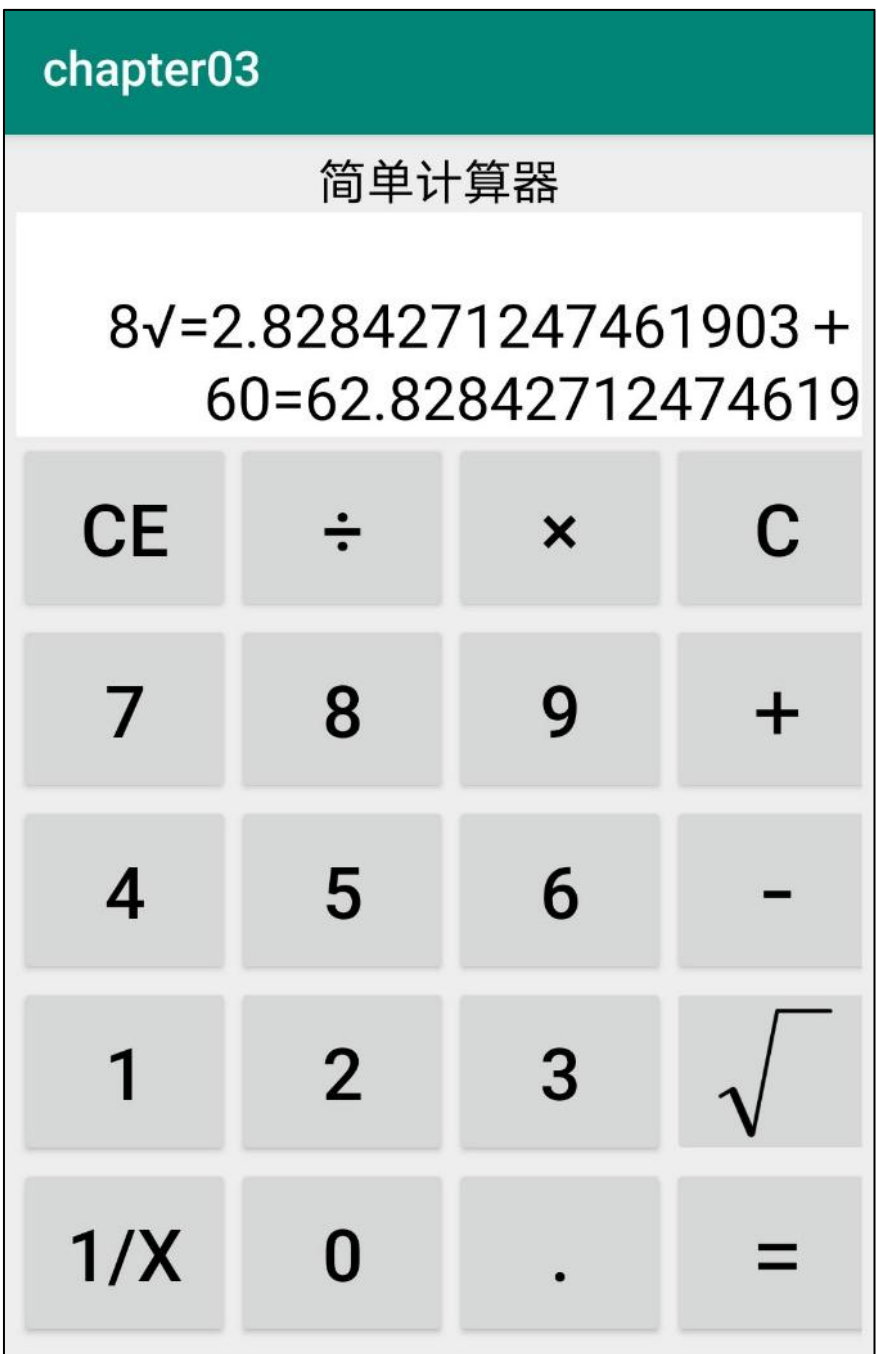
简单计算器用到的控件

- 线性布局LinearLayout: 计算器的整体布局是从上到下排列着的。
- 网格布局GridLayout: 计算器下半部分的几排按钮，正好成五行四列表格分布，适合采用GridLayout。
- 滚动视图ScrollView: 计算器界面如果超出屏幕大小，就要支持滚动。
- 文本视图TextView: 计算结果文本需要使用TextView，且文字靠下靠右显示。
- 按钮Button: 用于0-9的数字按键，以及加减乘除等运算按键。
- 图像按钮ImageButton: 开根号的运算符“ $\sqrt{\quad}$ ”虽然能够打出来，但是右上角少了一横，所以该按钮要用一张标准的开根号图片显示。

■ 计算器运行的演示效果



执行乘法运算的计算器界面



先执行开方运算再执行加法运算