

《Android 开发从入门到实战》

中级控件

让人人都能享受到高品质的教育服务

中级控件

图形定制

A

选择按钮

B

文本输入

C

对话框

D

图形Drawable

- Drawable 类型表达了各种各样的图形，包括图片、色块、画板、背景等。
- 包含图片在内的图形文件放在res目录的各个drawable目录下，其中drawable目录一般保存描述性的XML文件，而图片文件一般放在具体分辨率的drawable目录下。
- 各视图的background属性、ImageView 和 ImageButton的src属性、TextView和Button四个方向的drawable***系列属性都可以引用图形文件。

形状图形

- Shape图形又称形状图形，它用来描述常见的几何形状，包括矩形、圆角矩形、圆形、椭圆等等。
- 形状图形的定义文件是以shape标签为根节点的XML描述文件，它支持四种类型的形状：
 - rectangle：矩形。默认值
 - oval：椭圆。此时corners节点会失效
 - line：直线。此时必须设置stroke节点，否则会报错
 - ring：圆环

形状图形的规格定义

➤ 除了根节点shape标签，形状图形还拥有下列规格标签：

- size（尺寸），它描述了形状图形的宽高尺寸。
- stroke（描边），它描述了形状图形的描边规格。
- corners（圆角），它描述了形状图形的圆角大小。
- solid（填充），它描述了形状图形的填充色彩。
- padding（间隔），它描述了形状图形与周围边界的间隔。
- gradient（渐变），它描述了形状图形的颜色渐变。

圆角矩形的演示例子

- 指定形状内部的填充颜色
- 指定形状轮廓的粗细与颜色
- 指定形状四个圆角的半径

chapter05

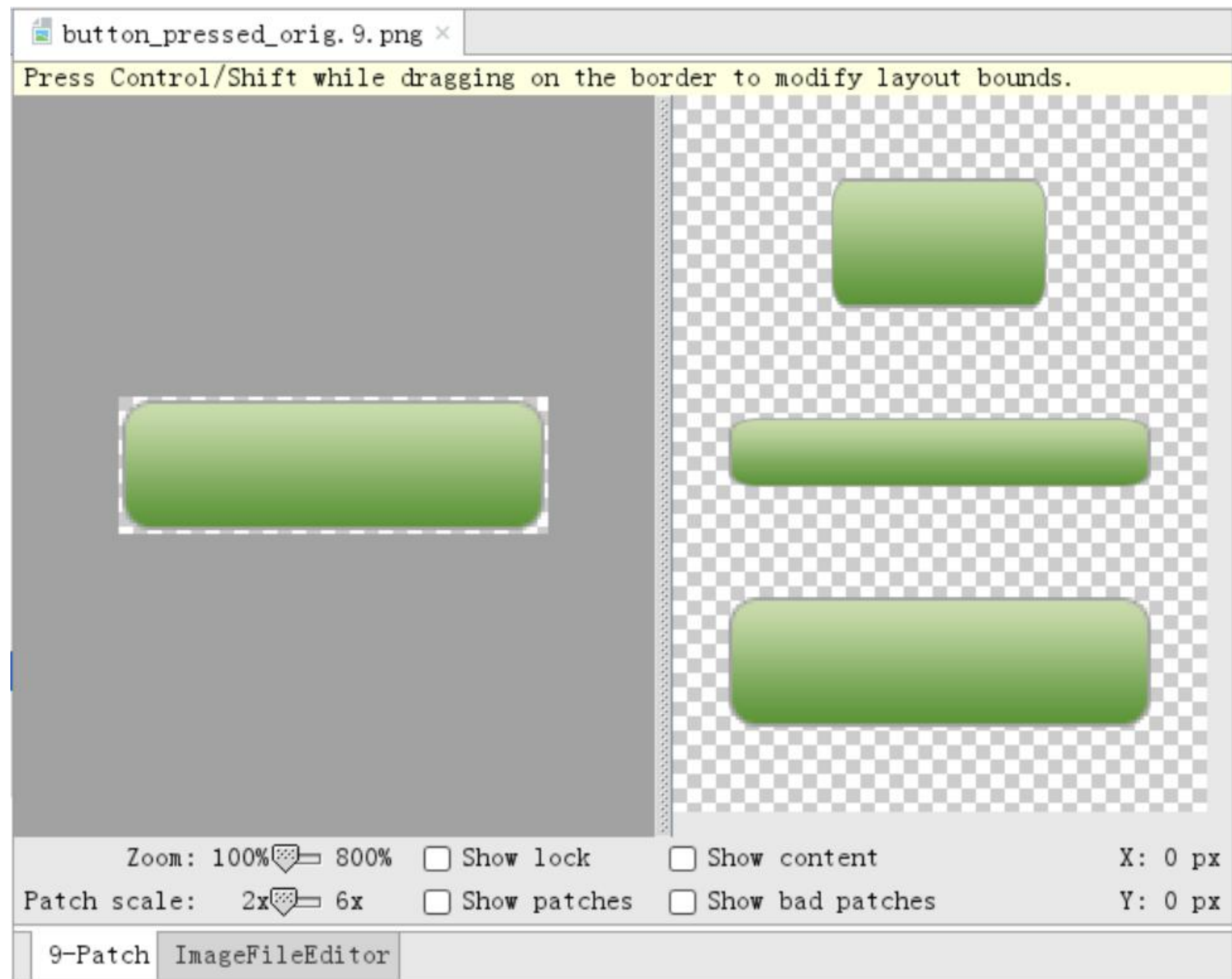


九宫格图片

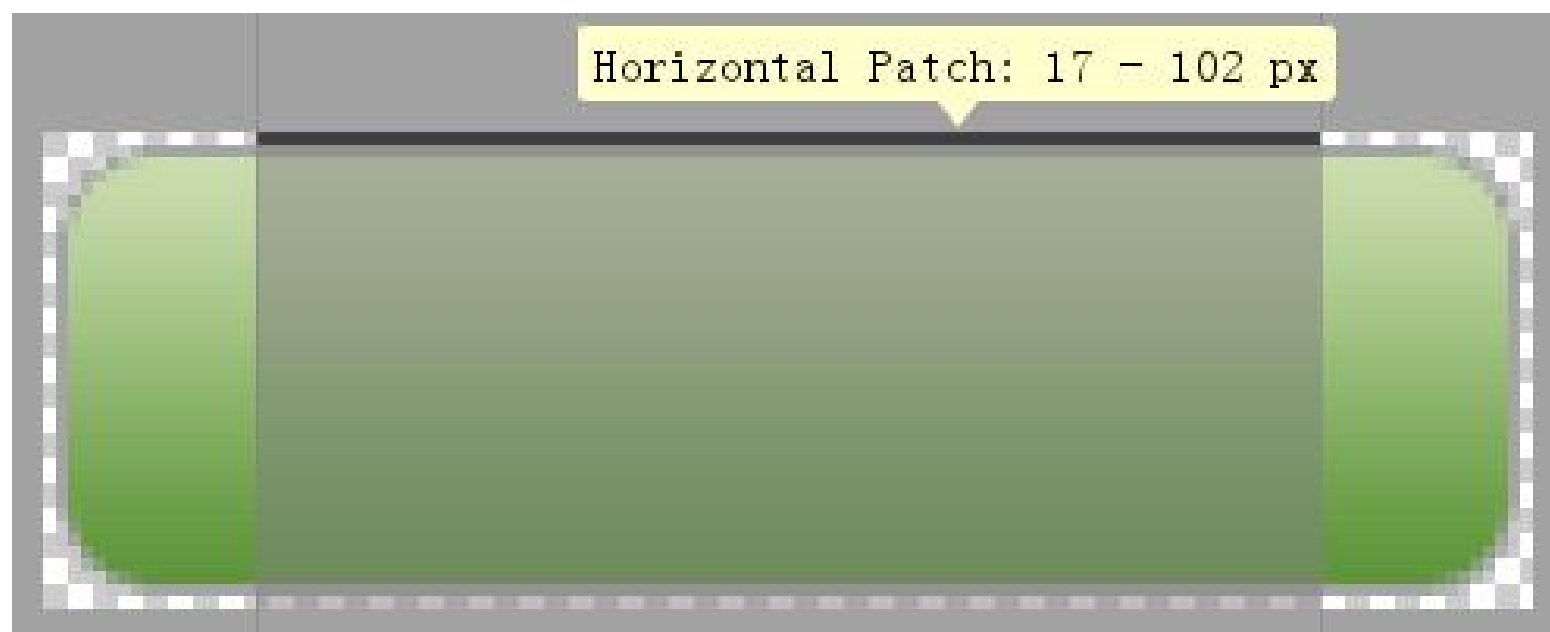
- 将某张图片设置成视图背景时，如果图片尺寸太小，则系统会自动拉伸图片使之填满背景。
- 可是一旦图片拉得过大，其画面容易变得模糊。



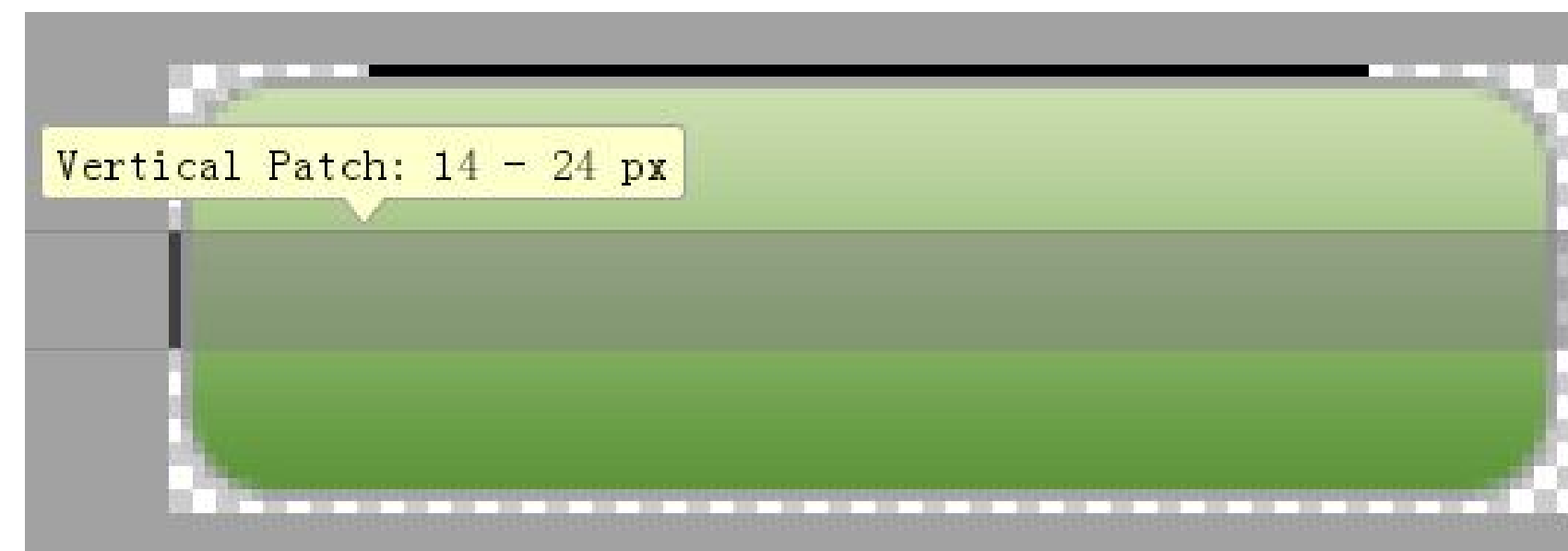
九宫格图片的制作窗口



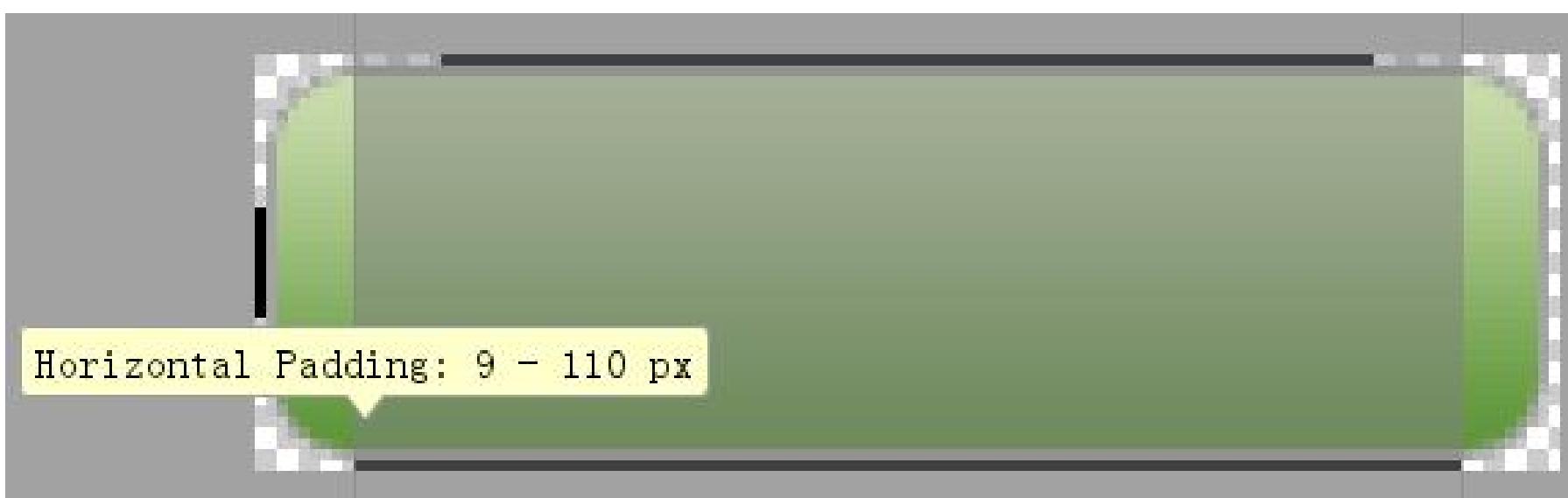
九宫格图片的四边涵义



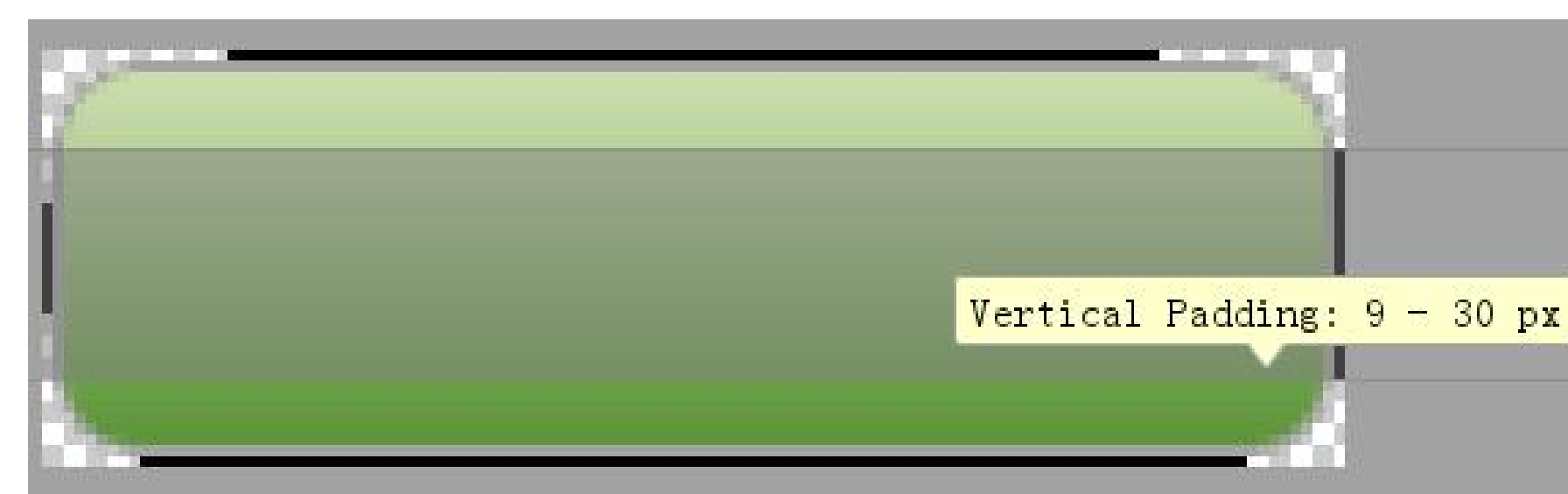
点九图片上边的边缘线



点九图片左边的边缘线



点九图片下边的边缘线



点九图片右边的边缘线

状态列表图形

- Button按钮的背景在正常情况下是凸起的，在按下时是凹陷的，从按下到弹起的过程，用户便能知道点击了这个按钮。



按下按钮时的背景样式



按钮弹起时的背景样式

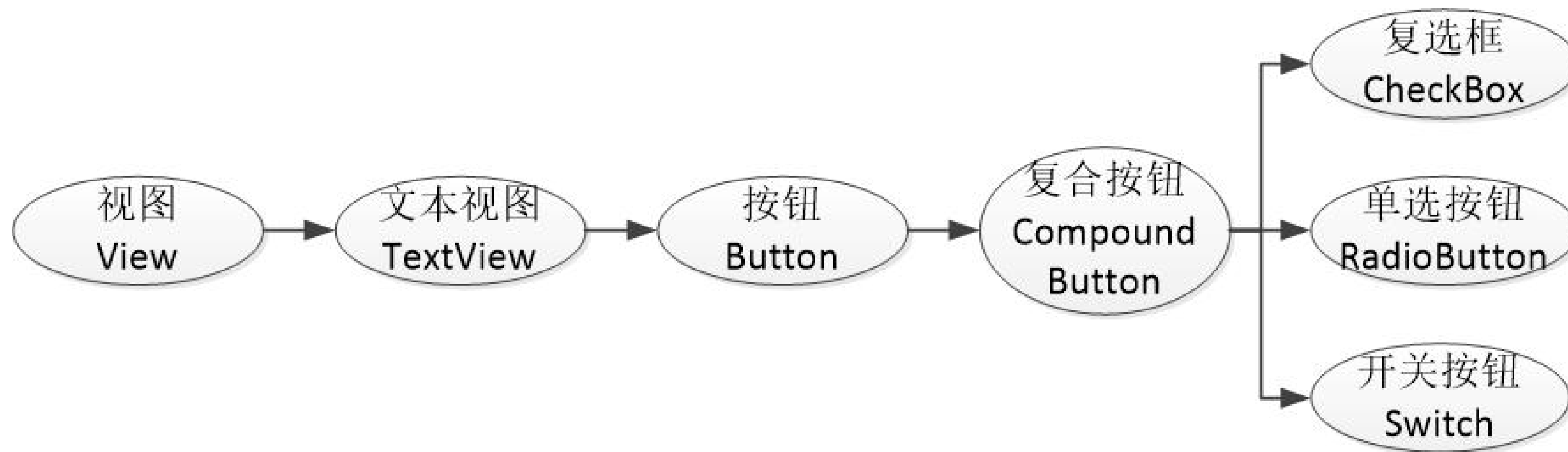
状态类型的取值说明

➤ 状态列表图形不仅用于按钮控件，还可用于其他拥有多种状态的控件。

状态类型的属性名称	说明	适用的控件
state_pressed	是否按下	按钮Button
state_checked	是否勾选	复选框CheckBox、单选按钮RadioButton
state_focused	是否获取焦点	文本编辑框EditText
state_selected	是否选中	各控件通用

复选框CheckBox

- CompoundButton类是抽象的复合按钮，由它派生而来的子类包括：复选框CheckBox、单选按钮RadioButton以及开关按钮Switch。
- 下图描述了复合按钮的继承关系：



CompoundButton的基本用法

- CompoundButton 在 XML 文件中主要使用下面两个属性。
 - **checked**: 指定按钮的勾选状态, true表示勾选, false表示未勾选。默认未勾选。
 - **button**: 指定左侧勾选图标的图形资源。如果不指定就使用系统的默认图标。
- CompoundButton 在 Java 代码中主要使用下列4种方法。
 - **setChecked**: 设置按钮的勾选状态。
 - **setButtonDrawable**: 设置左侧勾选图标的图形资源。
 - **setOnCheckedChangeListener**: 设置勾选状态变化的监听器。
 - **isChecked**: 判断按钮是否勾选。

复选框的演示效果

chapter05

☐ 这是系统的CheckBox

初始的复选框界面

chapter05

☒ 您勾选了这个CheckBox

首次点击后的复选框

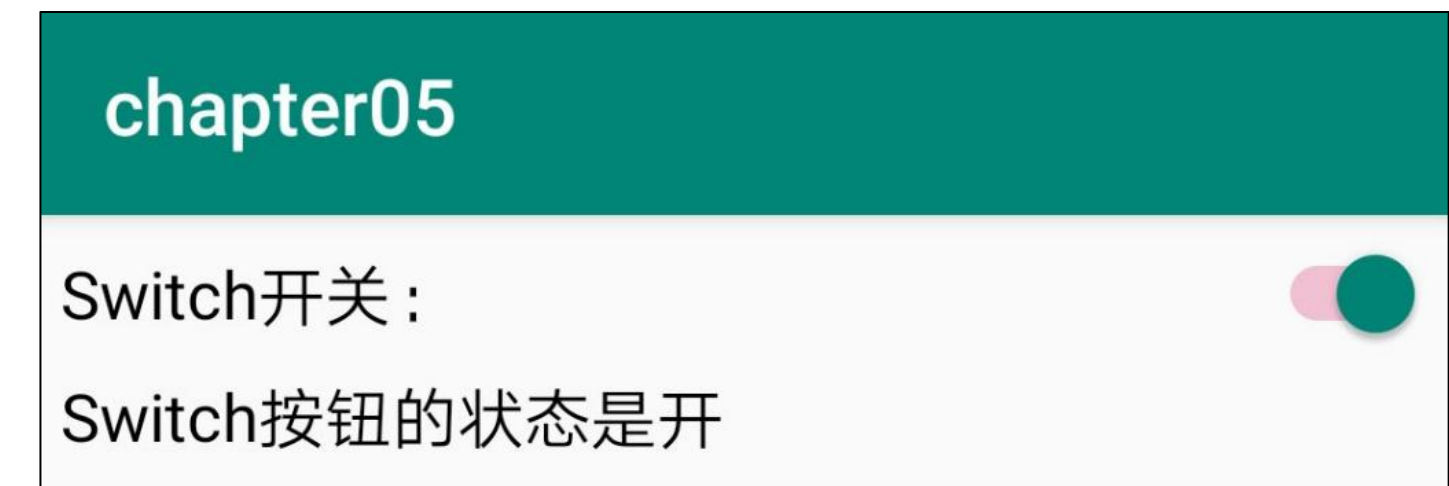
chapter05

☐ 您取消勾选了这个CheckBox

再次点击后的复选框

开关按钮Switch

- Switch 是开关按钮，它在选中与取消选中时可展现的界面元素比复选框丰富。
- Switch 控件新添加的XML属性说明如下：
 - textOn: 设置右侧开启时的文本。
 - textOff: 设置左侧关闭时的文本。
 - track: 设置开关轨道的背景。
 - thumb: 设置开关标识的图标。



仿iOS的开关按钮

- 借助状态列表图形StateListDrawable，分别定义已选中时候的“开”图形，以及未选中时候的“关”图形。
- 然后把CheckBox控件的background属性设置为该状态图形。



仿iOS按钮的“关”状态



仿iOS按钮的“开”状态

单选按钮RadioButton

- 单选按钮要在的一组按钮中选择其中一项，并且不能多选，这要求有个容器确定这组按钮的范围，这个容器便是单选组RadioGroup。
- **RadioGroup实质上是个布局**，同一组RadioButton都要放在同一个RadioGroup节点下。
除了RadioButton，也允许放置其他控件。

单选组的用法

- 判断选中了哪个单选按钮，通常不是监听某个单选按钮，而是监听单选组的选中事件。
- 下面是RadioGroup常用的3个方法：
 - check：选中指定资源编号的单选按钮。
 - getCheckedRadioButtonId：获取选中状态单选按钮的资源编号。
 - setOnCheckedChangeListener：设置单选按钮勾选变化的监听器。

单选按钮的演示效果

chapter05

请选择您的性别

☐ 男

☐ 女

初始的单选按钮界面

chapter05

请选择您的性别

☒ 男

☐ 女

哇哦，你是个帅气的男孩

选中左边按钮的单选界面

chapter05

请选择您的性别

☐ 男

☒ 女

哇哦，你是个漂亮的女孩

选中右边按钮的单选界面

编辑框EditText

- EditText 是文本编辑框，用户可在此输入文本等信息。
- EditText 的常用属性说明如下：
 - `inputType`：指定输入的文本类型。若同时使用多种文本类型，则可使用竖线 “|” 把多种文本类型拼接起来。
 - `maxLength`：指定文本允许输入的最大长度。
 - `hint`：指定提示文本的内容。
 - `textColorHint`：指定提示文本的颜色。

输入类型的取值说明

输入类型	说明
text	文本
textPassword	文本密码。显示时用圆点 “.” 代替
number	整型数
numberSigned	带符号的数字。允许在开头带负号 “-”
numberDecimal	带小数点的数字
numberPassword	数字密码。显示时用圆点 “.” 代替
datetime	时间日期格式。除了数字外，还允许输入横线、斜杆、空格、冒号
date	日期格式。除了数字外，还允许输入横线 “-” 和斜杆 “/”
time	时间格式。除了数字外，还允许输入冒号 “:”

焦点变更监听器

- 编辑框点击两次后才会触发点击事件，因为第一次点击只触发焦点变更事件，第二次点击才触发点击事件。
- 若要判断是否切换编辑框输入，应当监听焦点变更事件，而非监听点击事件。
- 调用编辑框对象的setOnFocusChangeListener方法，即可在光标切换之时（获得光标和失去光标）触发焦点变更事件。

监听焦点变更事件的演示效果

- 手机号码未输满11位，就点击密码框，此时校验不通过，一边弹出提示文字，一边把焦点拉回手机框。

chapter05

123456

请输入6位密码

登录

请输入11位手机号码

文本变化监听器

- 判断手机号输入满11位后自动关闭软键盘，或者密码输入满6位后自动关闭软键盘，此时要注册文本变化监听器。
- 达到指定位数便自动关闭键盘的功能，可以再分解为两个独立的功能点
 - 如何关闭软键盘；
 - 如何判断已输入的文字达到指定位数；

文本变化监听器的用法

- 调用编辑框对象的 `addTextChangedListener` 方法即可注册文本监听器。
- 文本监听器的接口名称为 **TextWatcher**，该接口提供了3个监控方法，具体说明如下。
 - `beforeTextChanged`：在文本改变之前触发。
 - `onTextChanged`：在文本改变过程中触发。
 - `afterTextChanged`：在文本改变之后触发。

监听文本位数自动关闭软键盘的演示效果



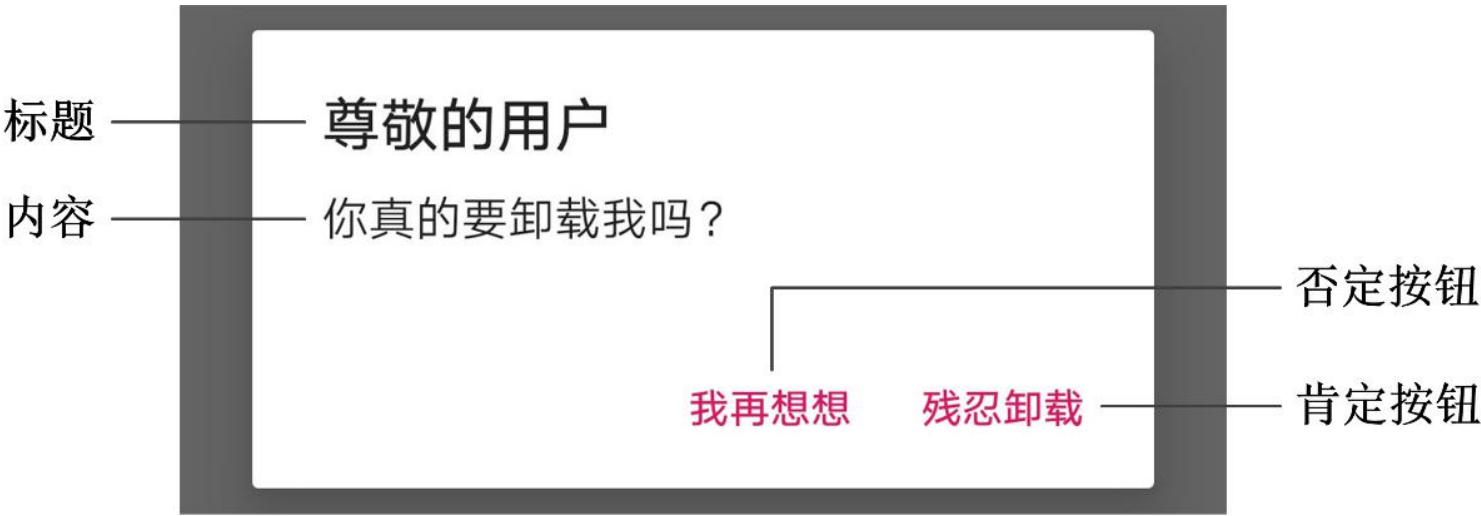
输入10位手机号码，软键盘未关闭



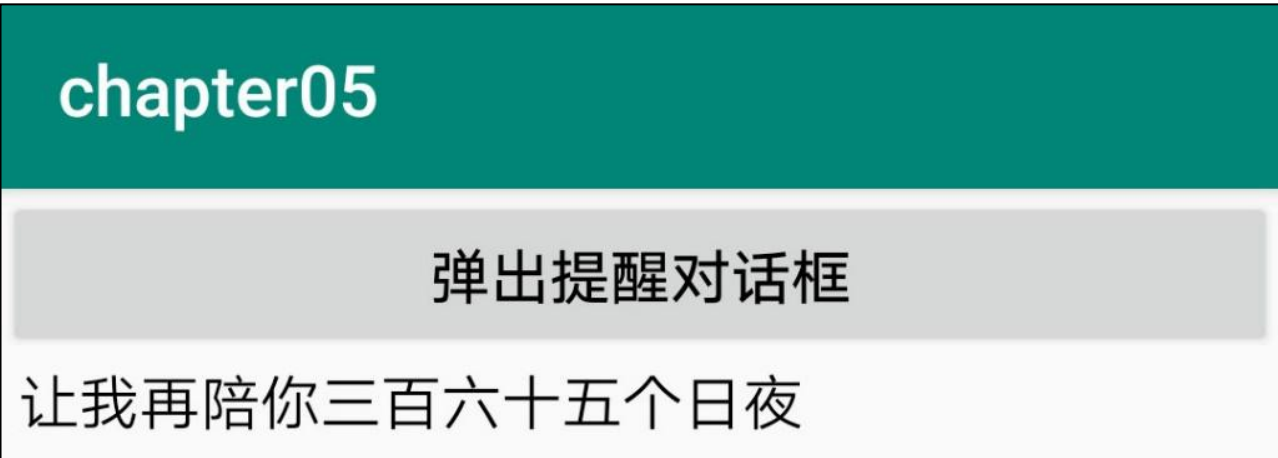
输入11位手机号码，软键盘已关闭

提醒对话框AlertDialog

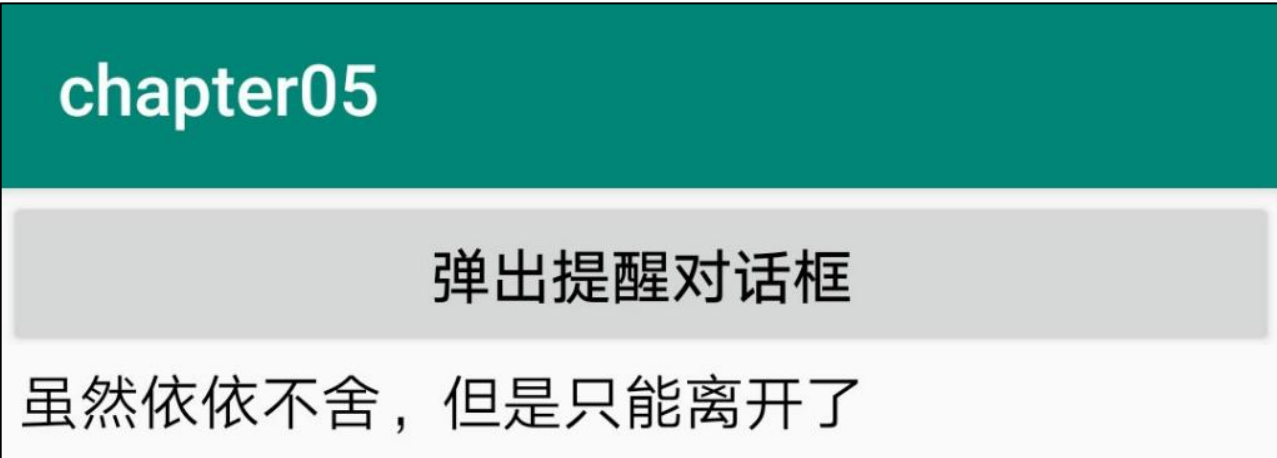
- AlertDialog 可以完成常见的交互操作，例如提示、确认、选择等功能。AlertDialog借助建造器 AlertDialog.Builder 才能完成参数设置。
- 调用建造器的 create 方法生成对话框实例，再调用对话框实例的show方法，在页面上弹出提醒对话框。



提醒对话框的效果图



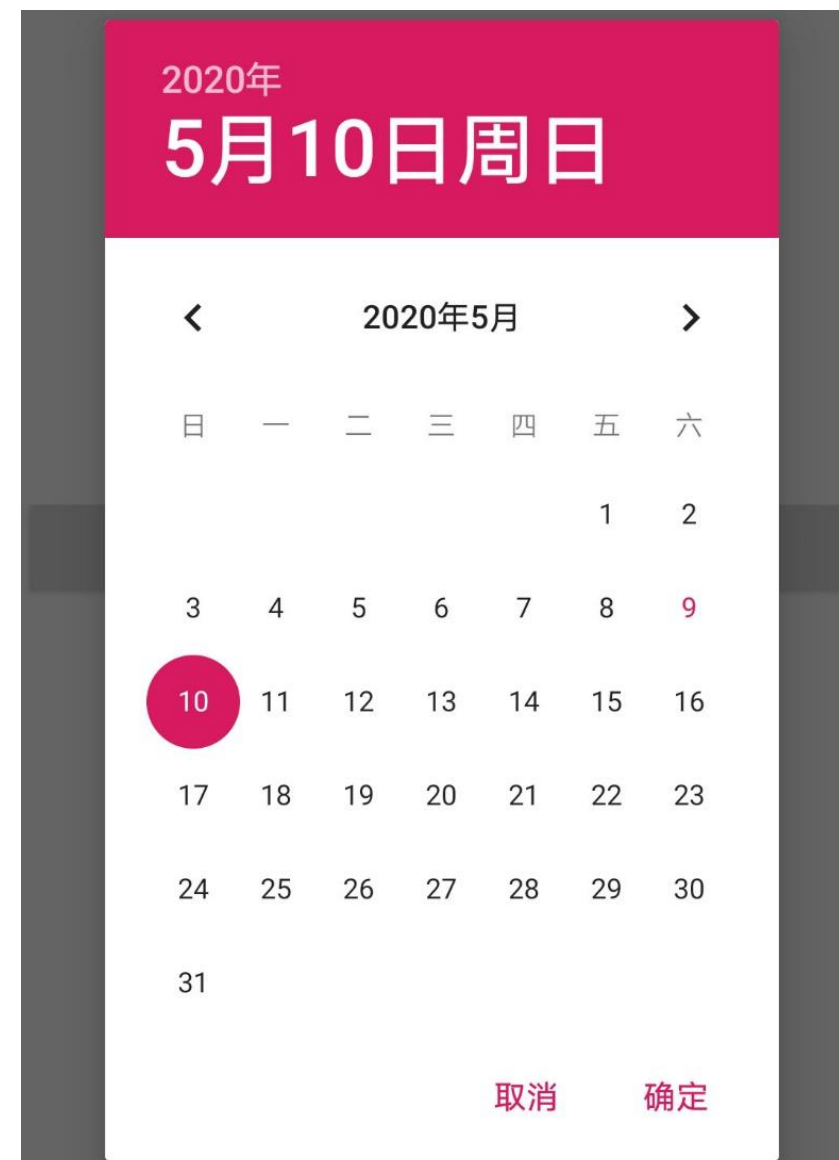
点击“我再想想”的截图



点击“残忍卸载”的截图

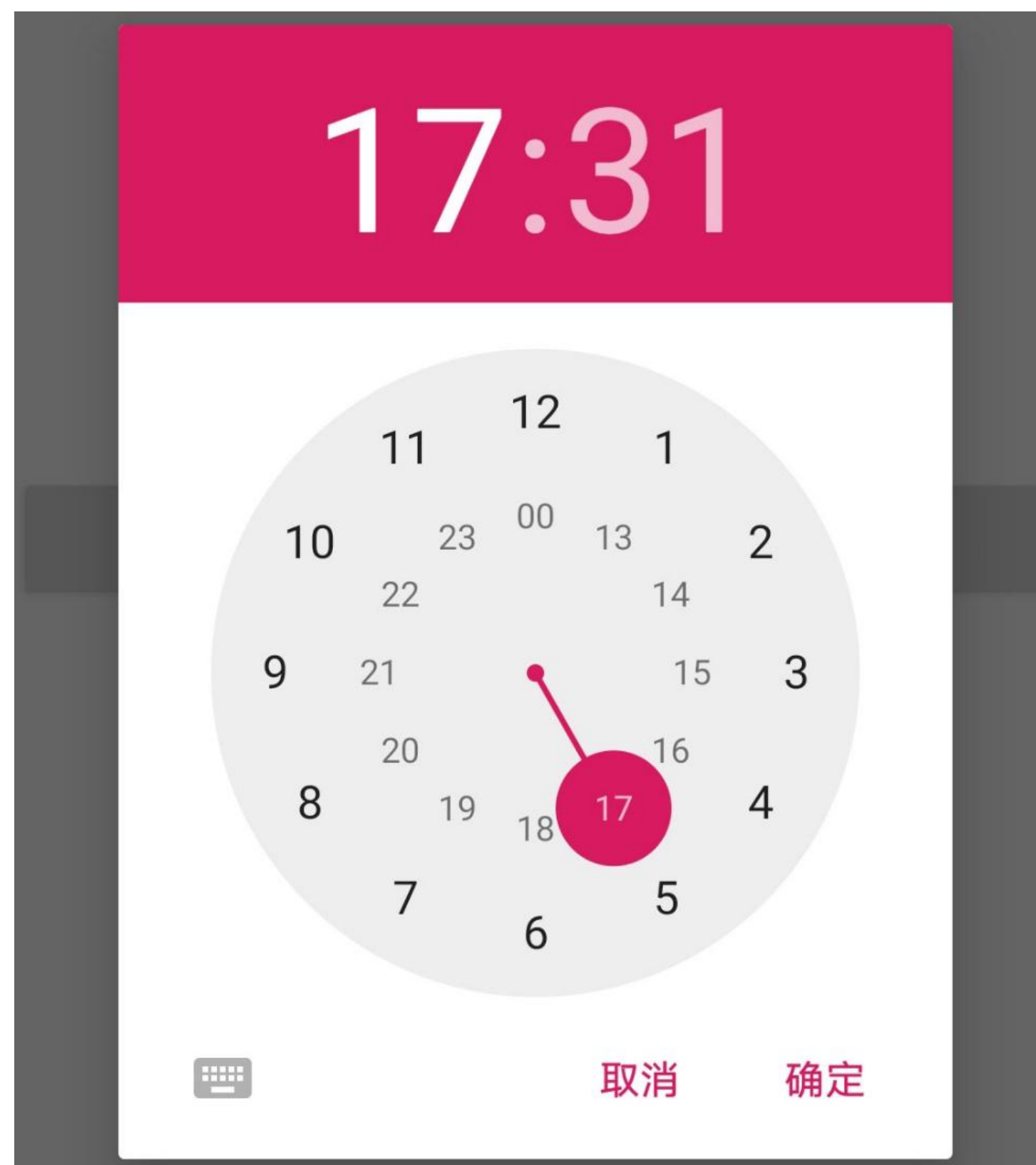
日期对话框DatePickerDialog

- 日期选择器 DatePicker 可以让用户选择具体的年月日。
- 但 DatePicker 并非弹窗模式，而是在当前页面占据一块区域，并且不会自动关闭。
- DatePickerDialog 相当于在 AlertDialog 上装载了 DatePicker，日期选择事件则由监听器 OnDateSetListener 负责响应，在该监听器的 onDateSet 方法中，开发者获取用户选择的具体日期，再做后续处理。



时间对话框TimePickerDialog

- 时间选择器 TimePicker 可以让用户选择具体的小时和分钟
- TimePickerDialog 的用法类似 DatePickerDialog。



实战项目：找回密码

➤ 登录页面一般有两种方式：

- 用户名与密码组合登录；
- 手机号与验证码组合登录；

chapter05

☒ 密码登录 ☐ 验证码登录

手机号码:

登录密码: [忘记密码](#)

☐ 记住密码

登 录

chapter05

☐ 密码登录 ☒ 验证码登录

手机号码:

验证码: [获取验证码](#)

登 录

两种登录方式的区别

- 密码输入框和验证码输入框的左侧标题以及输入框内部的提示语各不相同；
- 如果是密码登录，则需要支持找回密码；如果是验证码登录，则需要支持向用户手机发送验证码；
- 密码登录可以提供记住密码功能，而验证码的数值每次都不一样，无需也没法记住验证码；

找回密码的功能说明

- 如果用户忘记密码，则需跳到单独的找回密码页面，在该页面输入和确认新密码，并校验找回密码的合法性（通过短信验证码检查）。

chapter05

输入新密码：

请输入新密码

确认新密码：

请再次输入新密码

验证码：

请输入验证码

获取验证码

确 定

界面设计

- 单选按钮 RadioButton：用来区分是密码登录还是验证码登录。
- 文本视图 TextView：输入框左侧要显示此处应该输入什么信息。
- 编辑框 EditText：用来输入手机号码、密码和验证码。
- 复选框 CheckBox：用于判断是否记住密码。
- 按钮 Button：包含“登录”、“忘记密码”和“获取验证码”三个按钮。
- 线性布局 LinearLayout：界面从上往下排列，用到了垂直方向的线性布局。
- 相对布局 RelativeLayout：忘记密码的按钮与密码输入框是叠加的，且“忘记密码”与上级视图右对齐。
- 单选组 RadioGroup：放置密码登录和验证码登录这两个单选按钮。
- 提醒对话框 AlertDialog：通过提醒对话框向用户反馈结果。

找回密码之时的参数传递

- 整个登录模块由登录页面和找回密码页面组成，因此这两个页面之间需要进行数据交互。
 - 从登录页面跳到找回密码页面，要携带唯一标识的手机号码作为请求参数；
 - 从找回密码页面回到登录页面，要将修改之后的新密码作为应答参数传回去；