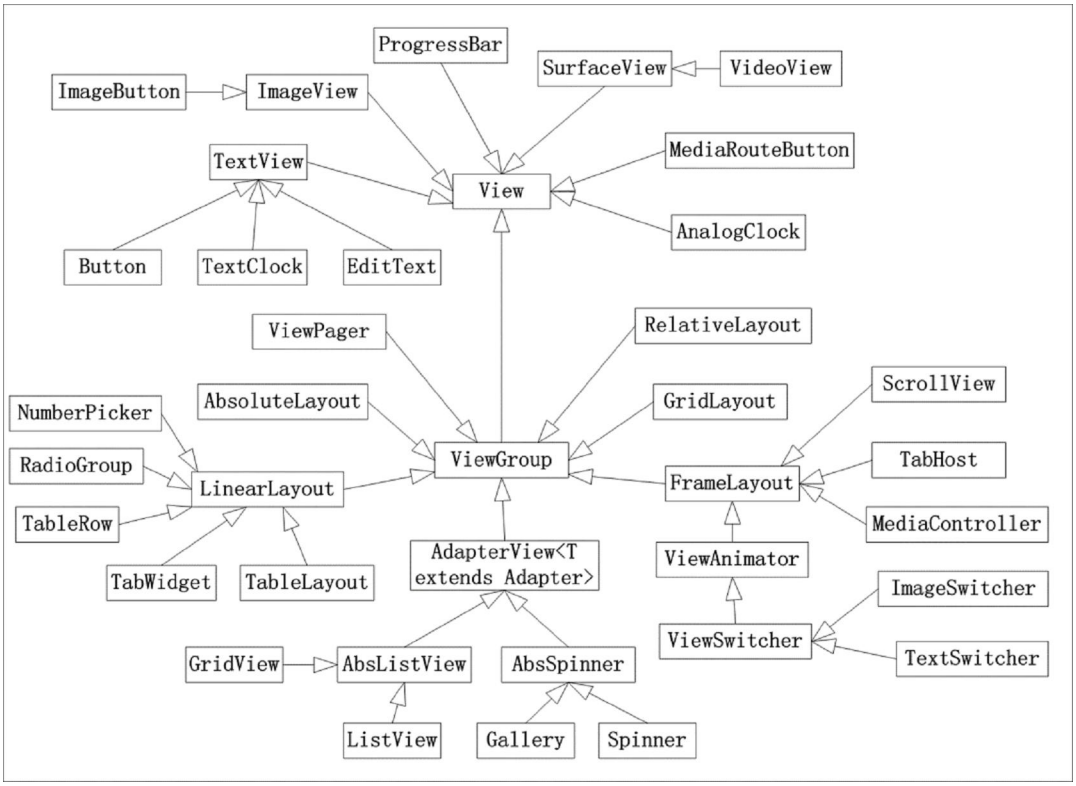
摘要:

TextView,EditText,Button,事件处理,ImageView,ImageButton,
CheckBox,RadioButton,ToggleButton,Switch

一、View类的常用xml属性：【了解】

①.Android中所有的UI（用户界面）元素都是使用View和ViewGroup对象建立的
②.View是一个可以将一些信息绘制在屏幕上并与用户产生交互的对象
③.ViewGroup是一个能够包含多个的View或ViewGroup的容器。
④.Android提供了一系列的View和ViewGroup的子类，开发者可以灵活地组合使用它们来完成界面布局、界面元素绘制和与用户交互等工作
⑤.开发者还可以选择性地继承一些系统提供的View，来自定义View，把自己定义的界面元素显示给用户。

UI视图结构



（一）、类结构：

java.lang.Object

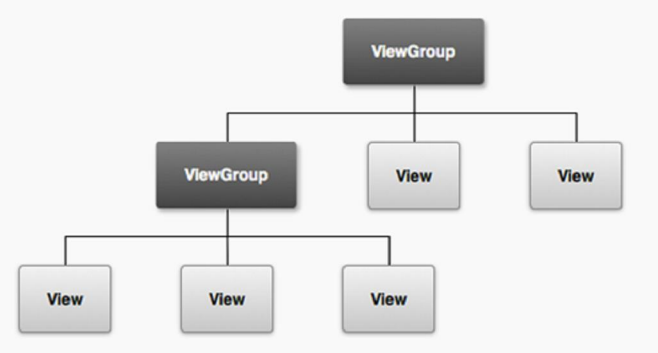
↳ android.view.View

（二）、View及其子元素常用属性：（各种布局及控件的共同属性）

1. android:id 注意:在同一个布局文件中,不能有重复id的组件
2. android:background 背景图或背景色
3. android:onClick 为该控件的单击事件绑定监听器
4. android:padding 设置控件的内间距，即组件内部的内容 与组件边界之间的 距离.
5. android:layout_margin 设置子组件的外边距,一个组件与其他组件,或一个组件与父容器之间的距离
6. android:visibility 设置该控件是否可见
7. android:alpha 设置该组件透明度
8. android:layout_height 子组件的布局高度
9. android:layout_width 子组件的布局宽度

（三）.Android中UI布局的嵌套【掌握】

Android的UI开发使用层次模型来完成，一般都是在一个ViewGroup中嵌套多层ViewGroup，每一层中含有任意数目的View。



注意:嵌套层次不要超过10层，否则会大幅降低运行效率，上图为3层

二、Android UI控件：

（一）、控件名称：【标红色的为常用的】

1. TextView 文本视图
2. EditText 文本编辑框
3. Button 按钮
4. ImageView 图像视图
5. ImageButton 图片按钮
6. ToggleButton 开关按钮 、Switch开关
7. RadioButton 单选按钮
8. CheckBox 多选框
9. Spinner 下拉列表
10. AutoCompleteTextView 自动完成文本框
11. ProgressBar 进度条
12. SeekBar 拖动条
13. RatingBar 星级评分条
14. TimePicker、DatePicker 时间选择器、日期选择器
15. AnalogClock、DigitalClock 模拟时钟、数字时钟
16. Dialog（AlertDialog提示对话框、ProgressDialog进度对话框、TimePickerDialog时间选择对话框、DatePickerDialog日期选择对话框）
17. ListView列表视图【**最重要的UI组件之一**】、GridView 网格视图
18. ScrollView 滚动视图
19. ExpandableListView 可展开列表视图
20. WebView 网页视图
21. SearchView 搜索框
22. TabHost 书签选项卡
23. Notification 通知
24. Toast 吐司（短时提醒）
25. Menu（OptionMenu 选项菜单、ContextMenu上下文菜单, PopupMenu弹出菜单）
26. ActionBar 动作导航条

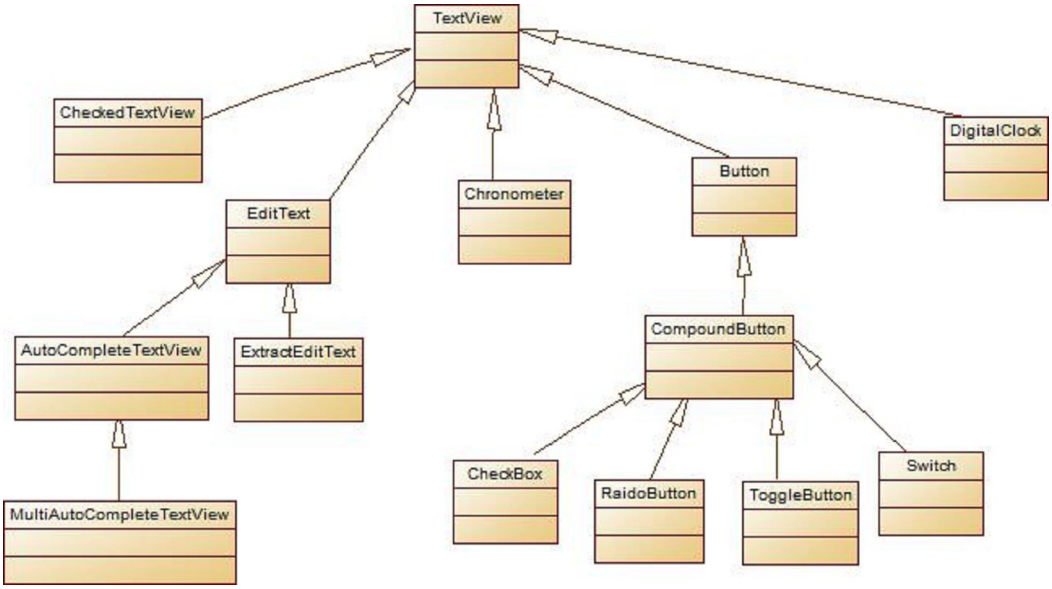
三、基本控件：——TextView【掌握】

（一）、TextView类结构：

java.lang.Object

↳ android.view.View

↳ android.widget.TextView



（二）、TextView 常用属性：

1、andorid:text　　设置文本的内容

2、 android:textColor　　设置文本的颜色

3、 android:textSize　　设置文本的字体大小（sp）

4、 andorid:height　　设置文本的高度，以像素为单位

5、 android:width　　设置文本的宽度，以像素为单位

6、 android:inputType　　设置文本的类型。不指定即是普通文本，可选的值有textPassword, number等。

7、 android:ems　　设置TextView的宽度为N个字符的宽度

8、android:gravity　　设置文本框内文本(相对于文本框)的对齐方式。
可选项有：top、bottom、left、right、center、center_vertical、center_horizontal等等。这些属性值也可以同时指定，各属性值之间用竖线隔开。例如right|bottom

9、android:drawableLeft　用于在文本框左侧绘制图片。该属性值通过“@drawable/图片文件名”来设置。

10、android:drawableRight　用于在文本框右侧绘制图片。该属性值通过“@drawable/图片文件名”来设置。

11、android:drawableTop　用于在文本框上侧绘制图片。该属性值通过“@drawable/图片文件名”来设置。

12、android:drawableBottom　用于在文本框下侧绘制图片。该属性值通过“@drawable/图片文件名”来设置。

13、android:hint　设置当文本框内文本内容为空时，默认显示的提示性文字。

14、 android:textStyle：设置字形[normal(普通) , bold(粗体) , italic(斜体) , bold|italic(又粗又斜)] 可以设置一个或多个，用“|”隔开

15、android:ellipsize[ɪ'lɪpsɪz]：设置当文字过长时,该控件该如何显示。有如下值：
"start"—省略号显示在开头;
"end" ——省略号显示在结尾;
"middle"—-省略号显示在中间;
"marquee" ——以跑马灯的方式显示(动画横向移动)

想要实现文字的跑马灯效果，以下五个属性必须全写
android:ellipsize="marquee"
android:marqueeRepeatLimit="marquee_forever"
在ellipsize指定marquee的情况下，设置重复滚动的次数，当设置为 marquee_forever时表示无限次。
android:focusable="true"
android:focusableInTouchMode="true"
android:singleLine="true"

16、android:maxLength：限制显示的文本长度，超出部分不显示。
17、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也显示。
18、android:singleLine　　设置文本是否是单行显示(默认false, 超出部分会换行)。如果设置为true,则超出部分不显示, 只显示省略号...

四、基本控件：——EditText【掌握】

API查询路径:Develop>API Guides>User Interface>Input Controls>TextFields>EditText

（一）、 EditText 类结构：

java.lang.Object

↳ android.view.View

↳ android.widget.TextView

↳ android.widget.EditText

所以 EditText 继承了TextView的所有属性。常用属性及EditText类的对应方法:

属性名称	对应方法	说明
android:cursorVisible	setCursorVisible(boolean)	设置光标是否可见，默认可见
android:lines	setLines(int)	设置编辑文本的行数
android:maxLines	setMaxLines(int)	设置编辑文本的最大行数
android:minLines	setMinLines(int)	设置编辑文本的最小行数
android:password	setTransformationMethod (TransformationMethod)	设置文本框中的内容是否显示为密码
android:phoneNumber	setKeyListener(KeyListener)	设置文本中的内容只能是电话号码
android: scrollHorizontally	setHorizontallyScrolling(bool)	是否设置为水平滚动
android: selectAllOnFocus	setSelectAllOnFocus	当文本获得焦点时自动选中全部文本内容
android:singleLine	setTransformationMethod (TransformationMethod)	设置文本为单行模式
android:maxLength	setFilters(InputFilter)	设置最大显示长度

其中android:password属性不推荐使用,可以用inputType="textPassword"替代

1.在程序中设置EditText以明文显示:

```
et.setInputType(InputType.TYPE_CLASS_TEXT);
```

2.在程序中设置EditText以密码显示:

```
et.setInputType(InputType.TYPE_CLASS_TEXT | InputType.TYPE_TEXT_VARIATION_PASSWORD);
```

（二）、android:inputType的可选项：

1. android:inputType="text" 默认的
2. android:inputType="textPersonName"
3. android:inputType="textPassword"　　文本密码
4. android:inputType="numberPassword"　只能输入数字的密码
5. android:inputType="textEmailAddress"　电邮地址
6. android:inputType="phone"　　电话号码
7. android:inputType="textPostalAddress"　邮政地址
8. android:inputType="time"
9. android:inputType="date"
10. android:inputType="number"　数字

Figure 1. The default text input type.

Figure 2. The textEmailAddress input type.

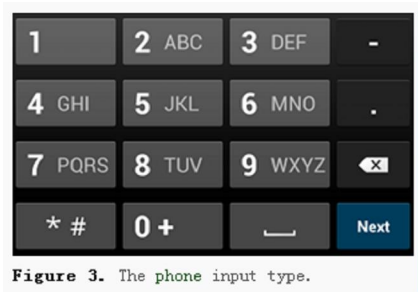
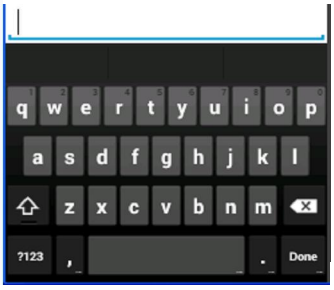
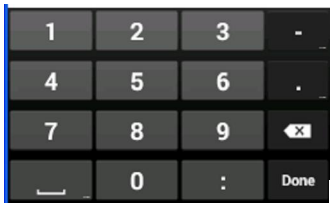


Figure 3. The phone input type.





textPersonName time date

textPostalAddress

（三）、EditText常用方法：

1、void setText (int resid);//传入字符串资源,例如R.string.xxx
void setText(char[] text, int start, int len);
void setText(CharSequence text);//CharSequence 是一个接口,可以传入String,StringBuffer,StringBuilder等.
2、Editable getText ();//根据需要toString()转换为字符串

五、基本控件：——Button【掌握】

（一）、Button类结构：

java.lang.Object
↳ android.view.View
↳ android.widget.TextView
↳ android.widget.Button

Button继承了TextView的所有属性。
重要属性:
android:background 指定按钮的背景图片
android:text 指定按钮上显示的文字
android:drawableLeft 在按钮左侧绘制图片
android:onClick 指定在Activity中的响应该按钮的点击事件的方法名.
同时onClick属性指定的方法有以下4点要求：
1）此方法的访问修饰符必须是public
2）此方法的返回值类型必须是void
3）此方法的参数列表，必须有且只有一个View类型的参数
4）此方法的方法名必须与onClick属性的属性值完全一致

六、Android事件处理的概述【掌握】

（一）、概念：

在Android中，我们可以通过事件处理使UI与用户互动（UI Events）。

具体形式则是以事件监听器（event listener）的方式来“监听”用户的动作。

Android提供了非常良好的UI事件处理机制。View是绘制UI的类，每个View对象都可以向Android注册事件监听器。每个事件监听器都包含一个回调方法（callback method），这个回调方法主要的工作就是回应或处理用户的操作。

（二）、Android UI事件处理中基于监听和回调的事件处理机制（event listener）：

就是为Android中的控件绑定特定的事件监听器。一旦该控件监听到有相应的动作发生，则该动作会触发事件监听器，而该监听器会自动调用回调方法做出相应的响应。事件监听器的核心就是它内部包含的回调方法。

（三）、Android种常用的事件监听器 及其回调方法（callback method）：

View.OnClickListener（View类中的内部接口） ： onClick() 单击事件

View.OnLongClickListener ： onLongClick() 长按事件

View.OnFocusChangeListener ：onFocusChange() 焦点改变事件

View.OnKeyListener ： onKey() 按键事件

View.OnCreateContextMenuListener： onCreateContextMenu() 创建上下文菜单事件

【备注：】常用的事件监听器：

1、RadioGroup.OnCheckedChangeListener 单选按钮组的勾选项改变监听器
2、CompoundButton.OnCheckedChangeListener 多选框勾选项改变监听器
3、AdapterView.OnItemSelectedListener 下拉列表框(Spinner)条目被选中监听器
4、AdapterView.OnItemClickListener ListView的条目单击监听器
5、AdapterView.OnItemClickListener GridView的条目单击监听器
6、DatePicker.OnDateChangedListener DatePicker的日期改变监听器
7、AbsListView.OnScrollListener ListView的滑动监听器

七、事件监听器的几种写法示例：

【备注：】所谓事件监听器，就是实现了一个特定接口的Java类的实例。

事件源--->事件--->监听器

以按钮的单击事件为例,实现事件监听器有5种形式：

1、内部类形式；实现OnClickListener接口
2、普通类形式；实现OnClickListener接口
3、自定义的Activity本身实现监听器OnClickListener接口，自身作为事件监听器类；
4、匿名内部类的形式创建事件监听器OnClickListener的实现类，并实现监听器内的方法。
5、在xml布局文件中指定时间处理方法android:onClick="xxx",在代码中添加相应的public void xxx(View v){...}

练习&作业:

[基础版:]在页面中添加一个按钮，一个TextView，要求实现：每点击一下按钮将TextView中的文字改成当前的点击次数，如：点击一次按钮，TextView上显示1，第二次点击按钮，TextView上显示2.

[改进版:]再定义一个减一按钮,每次点击,将TextView中的数值减一.

练习&作业:

请实现下面的UI,每次点击"提交"按钮时,打印日志Log.i(...).弹出吐司Toast.

import android.util.Log;
Log.i(TAG标签,"具体日志信息");//会打印在LogCat视图中,TAG通常是所在类的类名

[弹出吐司举例:]Toast.makeText(this,//上下文,一般传所在的Activity的this对象
"是按钮5干的!",//要显示的文本内容
Toast.LENGTH_SHORT//持续的时间
).show();//一定要调用一个show()方法,显示吐司

练习&作业:
完成计算器的基本运算功能.[预备知识:多个按钮的switch-case]

八、基本控件：——ImageView：

(一).ImageView的类结构

java.lang.Object

↳ android.view.View

↳ android.widget.ImageView

(二).ImageView的常用属性:

①、andorid:src 设置图片来源(显示什么图片)。格式 android:src="@drawable/图片名称"
一般使用png或jpg格式的图片,**图片名称可以使用的字符有:[a-z0-9_],不带后缀名,不能以数字开头,不能使用大写字母**,
如果要在代码中设置显示哪张图片:用ImageView对象调用setImageResource(R.drawable.xxx);

②、android:adjustViewBounds 用于设置 ImageView 是否调整自己的边界，来保持所显示图片的长宽比例(不会截断图片)。默认false.

③、android:maxHeight 设置 ImageView 的最大高度。需要先设置android:adjustViewBounds为true，否则不起作用。

④、andorid:maxWidth 设置 ImageView 的最大宽度。需要先设置android:adjustViewBounds为true，否则不起作用。

限定宽度,边界自动调整

限定高度,边界自动调整

限定高度和宽度,边界自动调整

备注:如果想设置图片为固定大小,并且要保持图片的宽高比,则:

1. 设置adjustViewBounds属性为true;

2. 设置maxHeight,maxWidth;

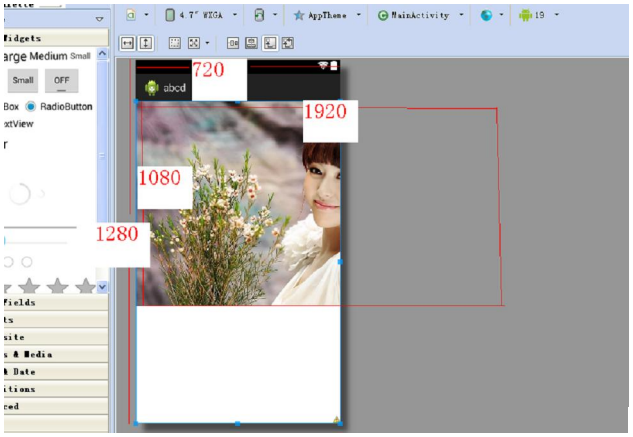
3. 设置layout_width="wrap_content",layout_height="wrap_content".

⑤、 android:background 设置背景,src上的图片会覆盖在background的上方,
在代码中设置用 ImageView调用setBackgroundResource(R.drawable.xxx);设置里面显示的图片.

⑥、android:scaleType : 缩放类型 设置所显示的图片如何缩放或移动，以适应ImageView的大小。

设置所显示的图片如何缩放或移动，以适应ImageView的大小。可选项：fitCenter**(默认值)**、fitStart 、fitEnd、 fitXY 、 center、centerCrop居中裁剪、centerInside、matrix
注意:要把layout_width和layout_height设置为 match_parent 或 固定值 时
android:scaleType属性才有效. 如果设置为wrap_content则该属性无效.

- matrix(原尺寸多余截掉): 保持原图大小、从ImageView的左上角的点开始，以矩阵形式绘图。 超出部分会被截掉.
- fitXY (伸缩填满)：把图片按照指定的大小在View中显示，拉伸(压缩)显示图片，不保持原比例，填满View. 如果未指定长宽,则受限于手机屏幕.

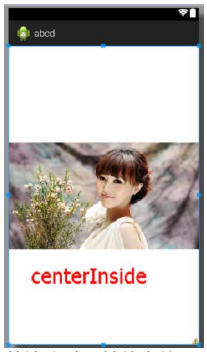


- fitStart：把图片**按比例**扩大(缩小)到View的宽度(或高度)，不截断，显示在View的上部分位置
- fitCenter **(默认值)**：把图片**按比例**扩大(缩小)到View的宽度(或高度),不截断，居中显示
- fitEnd ：把图片**按比例**扩大(缩小)到View的宽度(或高度)，不截断，显示在View的下部分位置

- center(原尺寸居中,多余的截掉) ： 以原图的几何中心点和ImagView的几何中心点为基准，按图片的原来size居中显示，不缩放. 当图片长/宽超过View的长/宽，则截取图片的居中部分显示. 当图片小于View 的长宽时，只显示原图片的尺寸,不拉伸。
- centerCrop(按比例伸缩,多余截掉并使 图填满Imageview) ：以原图的几何中心点和ImagView的几何中心点为基准，按比例扩大(图片小于View的宽时)图片的size。 居中显示，使得图片长 (宽)等于或大于View的长(宽),并按View的大小截取图片。 当原图的size大于ImageView时，按比例缩小图片，使得长宽中有一向等于ImageView,另一向大于ImageView。 实际上，使得原图的size大于等于ImageView .

  简言之,均衡的缩放图像（保持图像原始比例），使图片的两个坐标（宽、高）都大于等于 相应的视图坐标。使图像位于视图的中央。
- centerInside(大图按比例缩小后完整居中,小图不变) ：以原图的几何中心点和ImagView的几何中心点为基准，将图片的内容完整居中显示， 如果原图较大,通过按比例缩小原来的size使得图片长(宽)等

于或小于ImageView的长(宽);原图较小则不放大(fitCenter会将小图放大).



总结:上述属性值中的1,2,5

1个:不保证尺寸和比例,只填满,不截: fitXY

2个: 保持原图尺寸超出部分截掉的:matrix,center

5个:按比例缩放的:

fitCenter(不截,小图会放大) , fitStart(不截), fitEnd(不截),

centerCrop(满后超截) , centerInside(不截,小图不放大)

用电子表格比较:

比较方面	比例不变	尺寸不变	超出截掉	小图可放大	大图可缩小	填满高度或宽度	填
fitCenter	y	n	n	y	y	y	n
centerInside	y	n	n	n	y	y	n

根据需求来决定使用何种缩放模式.

(三) 、ImageView常用方法 :

1、setImageBitmap(Bitmap); //用输入流或字节数组把图片设置到ImageView中去.
可以利用BitmapFactory.decodeStream(.....), BitmapFactory.decodeByteArray(...)等方法返回Bitmap
iv.setImageBitmap(BitmapFactory.decodeStream(getAssets().open("b.png")));//b.png要放在assets目录下

2、setImageDrawable(getResources().getDrawable(R.drawable.xxx));//设置图片源
3、setImageResource(R.drawable.xxx);// res/drawable/hdpi/xxx.png 或jpg 图片的名称必须不以数字开头,可以由[a-z_.0-9]组成
4、setBackgroundResource(R.drawable.xxx);//设置背景
5、setScaleType(ScaleType.FIT_XY);//设置缩放模式

练习1:

点击按钮换图;setImageResource(R.drawable.xxx)

提示:setImageResource(int resourceId),用一个int数组保存若干张图片的资源id

练习2：

点击按钮切换8种缩放模式

提示：ImageView的方法setScaleType(ScaleType.xxx)

注意：如果是设置为matrix类型，ImageView需要同时设置setImageMatrix(new Matrix());

/Day04_UI/src/com/qf/day04_ui/ScaleTypeActivity.java

/Day04_UI/res/layout/scaletype.xml

九、基本控件：——ImageButton【了解】：

ImageView的子类

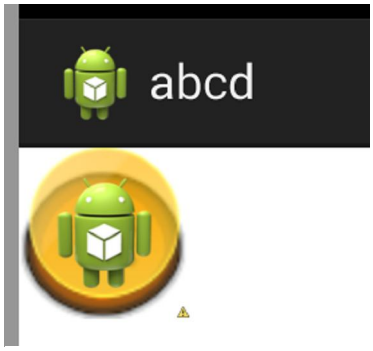
java.lang.Object

↳ android.view.View

↳ android.widget.ImageView

↳ android.widget.ImageButton

常用属性:
android:src="@drawable/xxx"属性指定按钮中顶层的图片,
android:background="@drawable/xxx"属性指定按钮底层的背景图片.



在不同资源文件夹里面放的图标大小参考:
mdpi:48*48
hdpi:72*72
xhdpi:96*96
xxhdpi:144*144

十、基本控件：——CheckBox：

（一）、 类结构介绍：

java.lang.Object
↳ android.view.View
↳ android.widget.TextView
↳ android.widget.Button
↳ android.widget.CompoundButton
↳ android.widget.CheckBox

CheckBox（间接）继承于Button，所以具有普通按钮的各种属性，但是与普通按钮不同的是， CheckBox 提供了可选中的功能。
【备注：】CheckBox的父类是CompoundButton，所以在使用监听器的时候要注意跟单选按钮进行区别。

（二）、核心代码：

A. UI的代码：

```
<CheckBox
  android:id="@+id/checkBox_main_hobby1"
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="学习" />

    <CheckBox
        android:id="@+id/checkBox_main_hobby2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="玩耍" />
```

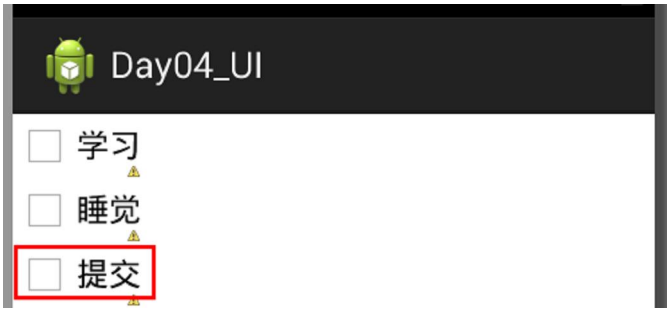
B. java代码：

```
    // 因为有多个复选框都注册在同一个监听器上，所以单独定义一个有名字的监听器对象以便重用。
OnCheckedChangeListener listener = new OnCheckedChangeListener() {//OnCheckedChangeListener复选框的选中状态发生改变时触发.
    //多选框中的    OnCheckedChangeListener是CompoundButton中定义的内部接口.import的时候注意不要引入错了.
    //单选按钮中的 OnCheckedChangeListener是RadioGroup中定义的内部接口.
        @Override
        public void onCheckedChanged(CompoundButton buttonView , boolean isChecked) {
        // 该方法中的第一个参数是当前被监听的多选项按钮对象，第二个参数是勾选状态改变后的状态，（如果选中是true，否则是false）。
                    if (isChecked) {
                        Log.i("====","==您勾选了：==>" + buttonView.getText());
                    } else {
                        Log.i("====","==您取消了：==>" + buttonView.getText());
                    }

            }
        };
//给多选按钮注册监听器,注册之后,一旦按钮的选中状态发生改变,将自动执行监听器实现类中的方法.
checkBox_main_hobby1.setOnCheckedChangeListener(listener);
checkBox_main_hobby2.setOnCheckedChangeListener(listener);
```

多选按钮不仅可以使用OnCheckedChangeListener监听器来获取被勾选的内容，也可以通过按钮的单击监听器OnClickListener来监听。

例如:



xml布局文件中增加了一个CheckBox "提交",希望点击提交的时候在日志中显示上面选中的复选框中的内容(学习睡觉....)

```
    <CheckBox
        android:id="@+id/button_main_submit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="提交" />
//在Activity中获取UI组件:
  button_main_submit = (CheckBox) findViewById(R.id.button_main_submit);
  checkBox_main_hobby1 = (CheckBox) findViewById(R.id.checkBox_main_hobby1);
  checkBox_main_hobby2 = (CheckBox) findViewById(R.id.checkBox_main_hobby2);
//CheckBox组件也可以注册OnClickListener监听器
  button_main_submit.setOnClickListener(new OnClickListener() {
   @Override
   public void onClick(View v) {
    // 实例化一个StringBuilder对象，则勾选的各项内容组成一个字符串。
    StringBuilder sb = new StringBuilder();
```

```
    // 多选框中有一个方法isChecked()，是判断该选项是否被勾选。如果勾选则返回true，如果取消勾选则返回false
    // 多选项中没有多选项Group这个概念，所以要逐一去判断每一个选项的勾选状态
    if (checkBox_main_hobby1.isChecked()) {
     sb.append(checkBox_main_hobby1.getText().toString());
    }
    if (checkBox_main_hobby2.isChecked()) {
     sb.append(checkBox_main_hobby2.getText().toString());
    }
    Log.i("====", "==您的爱好是：==>" + sb.toString());
   }
  });
```

练习&作业:
实现下面程序功能(Day03_CheckBox)

提示:
CheckBox的方法:
void setChecked(boolean) 设置选中状态
void toggle()切换选中状态
CharSequence getText()获取文本内容
boolean isChecked()获取选中状态

十一、基本控件：——RadioButton：

（一）、类结构介绍：

```
java.lang.Object
  ↳ android.view.View
   ↳ android.view.ViewGroup
    ↳ android.widget.LinearLayout
     ↳ android.widget.RadioGroup默认垂直线性布局,可以设置其orientation属性

java.lang.Object
  ↳ android.view.View
   ↳ android.widget.TextView
    ↳ android.widget.Button
     ↳ android.widget.CompoundButton
      ↳ android.widget.RadioButton
```

RadioButton(单选按钮)继承于Button，所以具有普通按钮的各种属性，但是与普通按钮不同的是，RadioButton提供了互斥式选中的功能。在使用RadioButton的时候，要使用RadioGroup(单选按钮组)来包围起这些RadioButton。
【备注：】RadioGroup是LinearLayout的子类，所以RadioGroup本质上是一个存放RadioButton的布局容器。
RadioGroup默认是垂直线性布局的.

（二）、重点记忆的方法：

1、RadioGroup类中的getCheckedRadioButtonId(),获取被选中的单选按钮的资源id
2、RadioGroup类中的setOnCheckedChangeListener(...),该按钮组中的被选中的单选按钮发生变化时触发.
自定义的监听器类需要重写OnCheckedChangeListener接口中的void onCheckedChanged(RadioGroup group, int checkedId) 方法,注意区分于CompoundButton中定义的内部接口OnCheckedChangeListener.

RadioButton重点方法:

boolean isChecked()获取选中状态

void setChecked(boolean)设置选中状态

void toggle()设置为"选中"状态,不能切换为"未选中"状态

RadioGroup重点方法:

int getCheckedRadioButtonId()获取被选择的单选按钮的id

void setOnCheckedChangeListener(OnCheckedChangeListener)注意是RadioGroup中定义的内部接口,不是CompondButton中定义的内部接口.

clearCheck()清空本RadioGroup里选中的RadioButton

案例:


```
public class RadioButtonActivity extends Activity {
 private RadioGroup rg;
 @Override
 protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_main);
  rg = (RadioGroup) findViewById(R.id.radioGroup1);
  rg. setOnCheckedChangeListener(new OnCheckedChangeListener() {
   /* 参数一：当前单选组的group对象 参数二：被选择项的资源id值,如R.id.rb1*/
   @Override
   public void onCheckedChanged(RadioGroup group, int checkedId) {
    switch (checkedId) {
    case R.id.rb1://布局文件中RadioButton的名字
     Log.i("====", "=======选择了女的");
     break;
    case R.id.rb2:
     Log.i("====", "=======选择了男的");
     break;
    case R.id.rb3:
     Log.i("====", "=======选择了你猜");
     break;
    }
   }
  });
 }
//接着定义"清空按钮"按钮和"选择你猜"按钮点击事件的处理方法,两个按钮都使用这个方法进行事件处理
//在布局中,两个Button都定义了android:onClick="select";
 public void select(View v) {
  switch (v.getId()) {
  case R.id.button1:    // 点击了"清空按钮"
       rg.clearCheck();//清空选中的RadioButton
       break;
  case R.id.button2:    // 点击了"选择你猜"按钮
   RadioButton rb3 = (RadioButton) findViewById(R.id.rb3);
   // rb3.setChecked(true);//设置为选中状态
   rb3. toggle();// 如果已经被选中了,将不会切换为未选中状态.如果未选中,将切换为选中状态.
   break;
  }
 }
}
```

ToggleButton控件:
切换按钮(一种Form Widgets)

```
java.lang.Object
  ↳ android.view.View
    ↳ android.widget.TextView
      ↳ android.widget.Button
        ↳ android.widget.CompoundButton
          ↳ android.widget.ToggleButton
```

重点方法:

void setOnCheckedChangeListener(OnCheckedChangeListener);//注意:使用CompoundButton中的OnCheckedChangeListener接口

核心代码:

给ToggleButton注册点击监听器,在事件处理方法中加入:

```
if(toggleBitton.isChecked()){
        editText.setInputType(InputType.TYPE_CLASS_TEXT);//密码显示为明文
}else{
        editText.setInputType(InputType.TYPE_CLASS_TEXT
        | InputType.TYPE_TEXT_VARIATION_PASSWORD);//密码显示为密文
}
```

Switch控件:

java.lang.Object

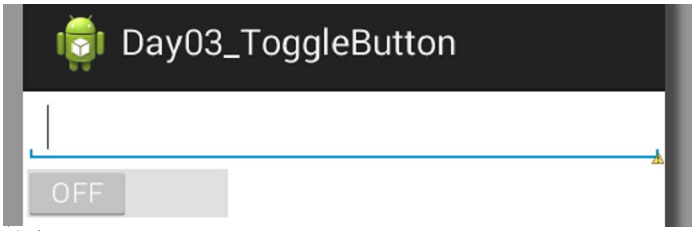
↳ android.view.View

↳ android.widget.TextView

↳ android.widget.Button

↳ android.widget.CompoundButton

↳ android.widget.Switch



重点方法:

void setOnCheckedChangeListener(OnCheckedChangeListener);//注意:使用CompoundButton中的OnCheckedChangeListener接口

作业:

注册时检测两次输入的密码是否一致,不一致用Toast弹出提示信息.

在日志中打印注册的信息.