Android程序设计

第一章、概述和基本控件

2019.3.24

isszym   sysu.edu.cn

目录

* 详细见每个部分目录和附录5

官方文档（中文） 官方文档（英文）
runoob cnblogs adroid.widget

安卓系统

概述

安卓系统结构

Dalvik虚拟机

- .dex JIT ART模式

app的四大组件

- Activity
- Service
- ContentProvider
- BroadcastReceiver

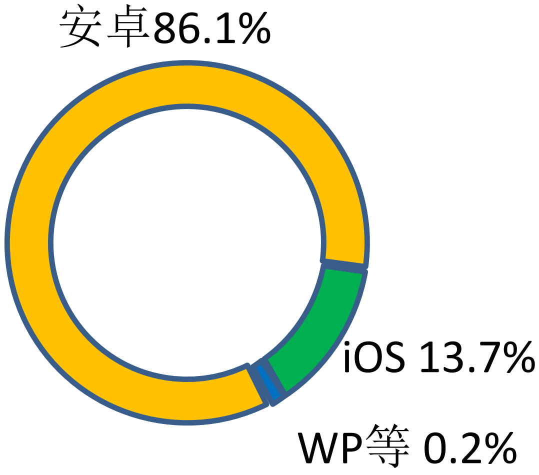
app的基本结构

- 界面程序
- 界面配置
- 清单文件
- 资源文件

第一个Android程序

概述

- Android操作系统最初由Andy Rubin开发，主要支持手机。2005年8月由Google收购注资。2007年11月，Google与84家硬件制造商、软件开发商及电信营运商组建开放手机联盟共同研发和改良Android系统。
- 随后Google以Apache开源许可证的授权方式，发布了Android的源代码。第一部Android智能手机发布于2008年10月。Android逐渐扩展到平板电脑及其他领域上，如电视、数码相机、游戏机等。
- 2011年第一季度，Android在全球的市场份额首次超过塞班系统，跃居全球第一。
- 据美国高德纳咨询公司统计，2017年第一季度中，安卓智能手机销量为3.271亿台，Android的市场份额高达86.1%，iOS系统智能手机销量为5190万台，iOS系统的市场份额为13.7%。



安卓系统结构

软件栈

原生应用程序和第三方应用程序(Java编写)

Java API(包)

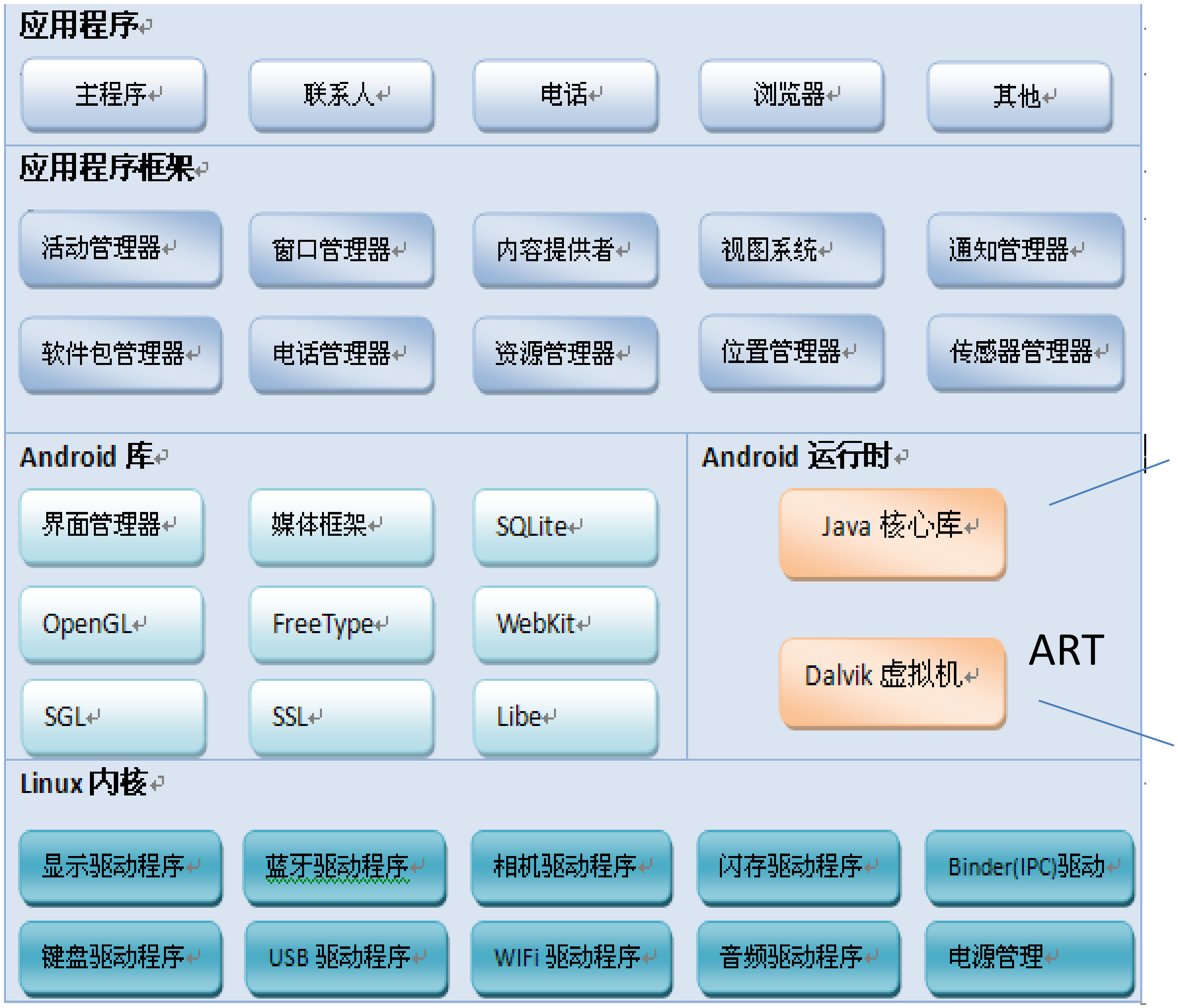
为应用程序提供所需要的包(Java编写)

JNI

Android库(C/C++编写)

可以编写C++应用程序通过NDK直接访问核心库

Linux 2.6 Android针对移动设备特点进行了优化



应用程序

- 主程序
- 联系人
- 电话
- 浏览器
- 其他

应用程序框架

- 活动管理器
- 窗口管理器
- 内容提供者
- 视图系统
- 通知管理器
- 软件包管理器
- 电话管理器
- 资源管理器
- 位置管理器
- 传感器管理器

Android 库

- 界面管理器
- 媒体框架
- SQLite
- OpenGL
- FreeType
- WebKit
- SGL
- SSL
- Libe

Android 运行时

- Java 核心库
- Dalvik 虚拟机

ART

Linux 内核

- 显示驱动程序
- 蓝牙驱动程序
- 相机驱动程序
- 闪存驱动程序
- Binder(IPC)驱动
- 键盘驱动程序
- USB 驱动程序
- WIFi 驱动程序
- 音频驱动程序
- 电源管理

提供Java语言的基本功能和扩展功能(包)

应用程序运行环境

NDK --Native Development Kit JNI—Java Native Interface ART模式-Android Runtime

Dalvik虚拟机

- 为了使应用程序开发脱离特定的硬件实现，安卓使用虚拟机来承载应用程序的执行。为了应用程序可以高效运行和最小限度地占用内存，安卓没有直接使用Java虚拟机，而是重新开发了虚拟机，即Dalvik虚拟机。**Dalvik虚拟机**上的执行文件为字节码程序(.dex文件)。
- Dalvik使用设备的底层Linux内核来处理基本功能，包括进程和线程管理、内存管理和安全管理。
- Dalvik采用即时编译（Just-In-Time，JIT）运行程序，每次运行字节码程序时都会进行一次编译，即转化为机器码程序。这大大降低了程序的运行速度。
- 谷歌在Android4.4中新加入了**ART模式**（Android Runtime），采用了**Ahead-Of-Time**（AOT）技术，系统在安装应用程序的时候会进行一次预编译，将字节码程序转换为机器码程序存储在本地，这样在运行程序时就不会每次都进行一次编译了，执行效率大大提升。 ART同时也改善了效能、垃圾回收(Garbage Collection)、应用程序除错与性能分析。

参考

安卓app的四大组件

Activity

显示用户界面并可以响应用户操作的程序。

Service

一种没有界面、在后台运行的程序。

Content Provider

因为每个SQLite数据库都只对创建它的应用程序可见，Content Provider是提供给其他应用程序访问数据库的一种方法。联系人数据就是通过ContentProvider提供给外部访问的。

Broadcast Receiver

一种Intent广播的侦听器。如果应用程序侦听到与预设的过滤标准匹配的Intent广播，就会立即进行响应。

Intent

提供了一种在应用程序之间或者同一个应用程序的不同组件之间传递消息的机制。

参考

安卓app的基本结构

界面程序
MainActivity.java

Activity

界面配置(静态)

activity_main.xml

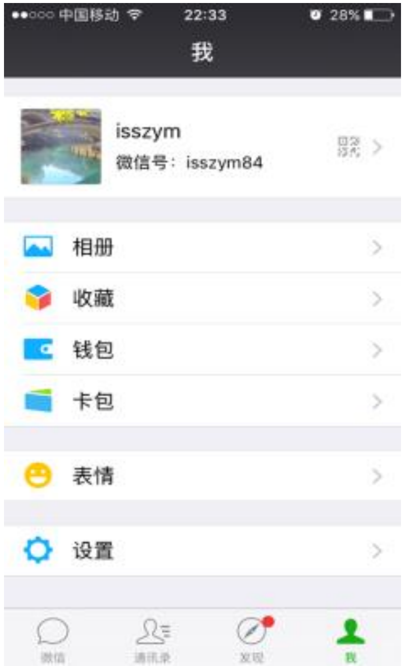
布局(layout)
控件(widge)

资源文件(图片等)

AndroidManifest.xml
清单文件

app的配置文件

定义了app的名称和图标等，并说明本app包含了哪些Activity、Service等组件。



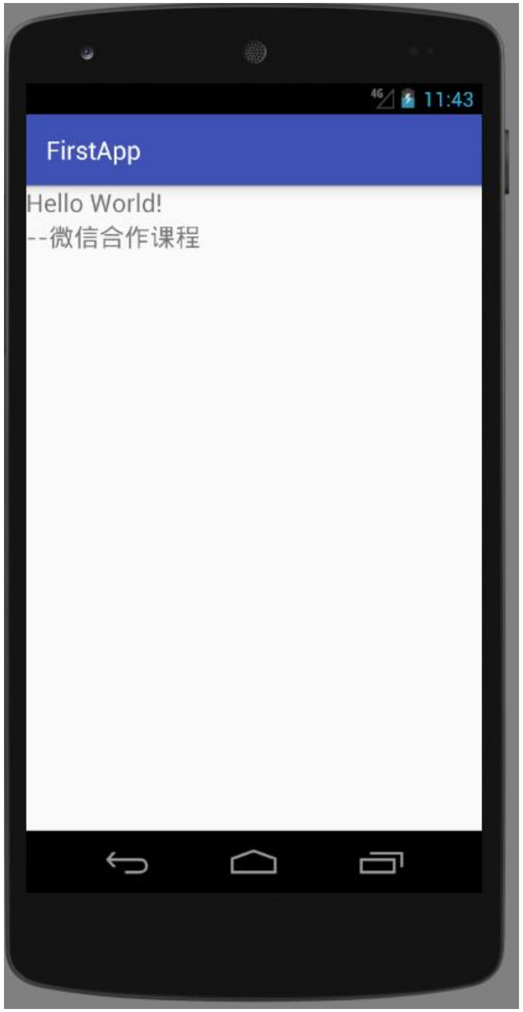
第一个Android程序

- **Activity的程序：**

项目： FirstApp

MainActivity.java

```
package com.example.isszym.firstapp;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

  @Override
  protected void onCreate(Bundle savedInstanceState){
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
  }
}
```



Hello World!
--微信合作课程

- savedInstanceState是上次退出本Activity时保存的状态。
- super.onCreate(savedInstanceState)用来把恢复的状态传递给父类(super)。
- setContentView用于设置本Activity使用的界面。

- MainActivity的界面文件：

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:text="Hello World!" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20sp"
        android:text="--微信合作课程" />
</LinearLayout>
```

LinearLayout(线性布局)
内部控件垂直或水平依次摆放

匹配双亲的宽度
也可以取一个具体值

文本框控件：显示一段文字

文字大小

包裹住内容的宽度

- android:id="@+id/activity_main"中activity_main为布局的id。+表示增加一个新ID。

控件--View（视图），Widget（窗口小部件），Control
GroupView 也是View，可以包含其他的View

- 程序清单：

AndroidManifest.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.isszym.firstapp">
    <application
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>
</manifest>
```

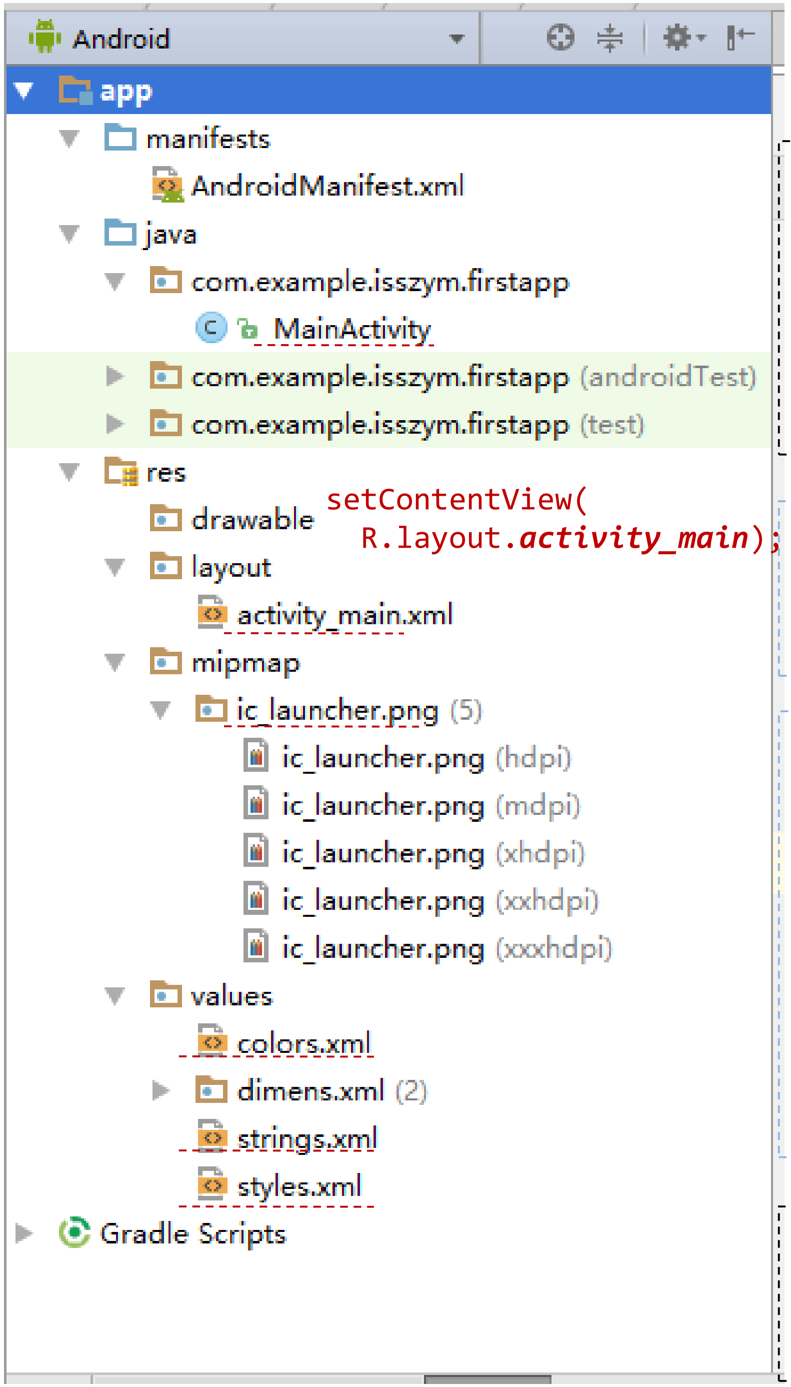
包名

app所用图标

app所用标签

设置activity的启动条件

action和category指出本activity为主activitiy。



```
setContentView(
   R.layout.activity_main);
```

颜色：#RRGGBBAA(四个字节，红绿蓝透明度)

color.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
   <color name="colorPrimary">#3F51B5</color>
   <color name="colorPrimaryDark">#303F9F</color>
   <color name="colorAccent">#FF4081</color>
</resources>
```

string.xml

```
<resources>
   <string name="app_name">FirstApp</string>
</resources>
```

style.xml

```
<resources>
   <!-- Base application theme. -->
   <style name="AppTheme"
      parent="Theme.AppCompat.Light.DarkActionBar">
      <!-- Customize your theme here. -->
      <item name="colorPrimary">@color/colorPrimary</item>
      <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
      <item name="colorAccent">@color/colorAccent</item>
   </style>
</resources>
```

dimension.xml

```
     <dimen name=”activity_horizontal_margin”>16dp</dimen>
     <dimen name=”activity_vertical_margin”>16dp</dimen>
</resources>
```

控件的基本属性

定义id

android:id="@+id/tv"

定义颜色

Color.argb()
0x80FF0000
#80FF0000
@color/colorPrimary

尺寸单位

px dp sp pt

设置字体

fontFamily textStyle
typeFace

设置宽度和高度

height width
padding layout_margin
lines ems letterSpacing

布局比重

layout_weight

文本设置

text textColor textSize
textStyle
shadowDx shadowDy
shadowColor shadowRadius
autolink linksClickable
textColorLink
textAllCaps textIsSelectable
textColorHighlight
textSacaleX

文本对齐

gravity
(top left
center_vertical center)
textAlignment
（textStart center
viewStart gravity）

省略文本

ellipsize
(none marquee
start end middle）

背景设置

background
tv.setBackgroundColor()
drawableLeft
drawableTop
bitmap tileMode repeat
shape solid stroke

定义id

- 每个控件可以定义界面中的唯一的id，通过id可以取到该控件。

```
<TextView
    android:layout_width="240dp"
    android:layout_height="wrap_content"
    android:text="Hello World!"
    android:textSize="24sp"
    android:id="@+id/tv"
 />
```

@+id/tv表示增加一个取名tv的新的id
@id/tv表示引用已存在的名叫tv的id

```
TextView tv = (TextView) findViewById(R.id.tv);
tv.setText("Hello!");
```

编译后的资源类

- **android:id**="@android:id/tabhost" 表示调用系统内部的ID “tabhost”

颜色

参考

- **通过ARGB构建颜色**

```
int color = Color.argb(127,255,0,0);//半透明的红色.Color.rgb(255,0,0)
    a-不透明度(alpha，0~255)   0-完全透明  255-完全不透明
    rgb – Red，Green，Blue，取值0~255
```

```
int color = 0x80FF0000;     // argb方式：0xFFFFAA00等同0xFFA0
tv.setTextColor(color);     // rgb方式：  0xFFAA00等同0xFA0
```

- 安卓系统颜色

```
android:textColor="@android:color/holo_red_dark"
int color = Color.BLUE;     // 安卓颜色(与android.graphics.Color.BLUE相同)
```

项目颜色：@color/colorAccent

主题颜色:?attr/colorPrimary

- **使用XML资源文件(res/values/colors.xml)**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="mycolor">#80FF0000</color>
</resources>
```

＊ 安卓命名颜色见附录

```
android:background="@color/mycolor"
```

在XML中使用

```
int color=getResources().getColor(R.color.mycolor);
tv.setBackgroundColor(color);
```

在Java中使用

尺寸单位

- **px**

  对应屏幕上的实际像素点（Pixels）。例如，320*480的屏幕在横向有320个象素，在纵向有480个象素。

- **dp，dip**

  与设备无关的像素（ device independent pixels），是一种逻辑长度单位。在160 dpi （dot per inch）屏幕上，1dp=1/160英寸。随着密度变化，对应的像素数量也会变化，但物理长度始终保持为1/160英寸。dip与dp相同。

- **sp**

  缩放像素（scaled pixels），是与屏幕密度无关的像素，常用于设置字体大小，1sp=1dp。

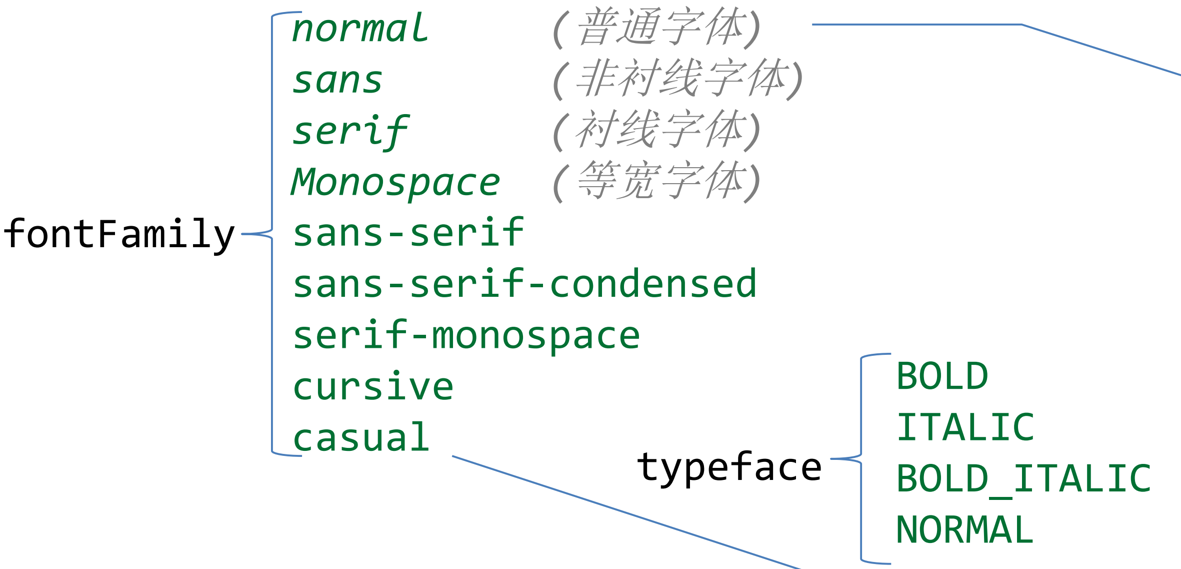
- **pt**

  屏幕物理长度单位，磅（points）。1pt=1/72英寸。

  其它物理单位：in-英寸（ inches ），mm-毫米（Millimeters ）。

设置字体

- 控件的字体是通过fontFamily或typeface设置的。



fontFamily

normal *(普通字体)*

sans *(非衬线字体)*

serif *(衬线字体)*

Monospace *(等宽字体)*

sans-serif

sans-serif-condensed

serif-monospace

cursive

casual

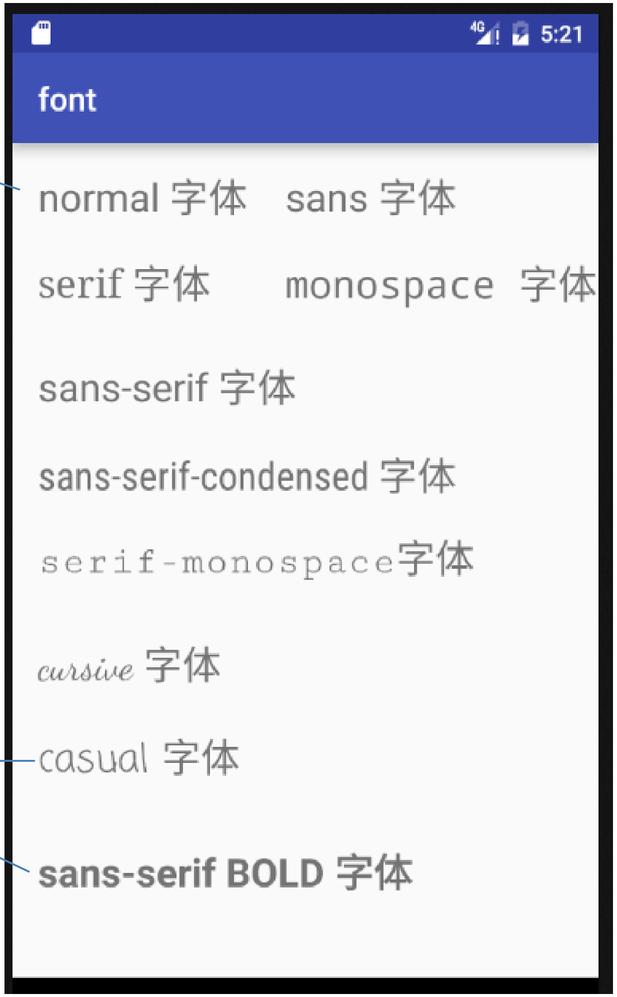
typeface

BOLD

ITALIC

BOLD_ITALIC

NORMAL



5:21

font

normal 字体 sans 字体

serif 字体 monospace 字体

sans-serif 字体

sans-serif-condensed 字体

serif-monospace字体

cursive 字体

casual 字体

sans-serif BOLD 字体

- 在xml中设置字体

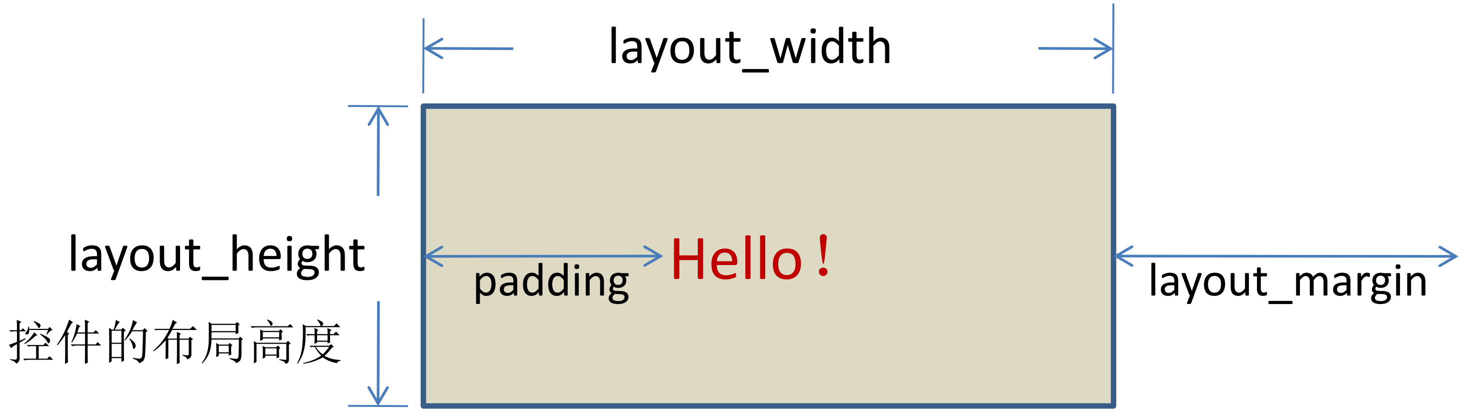
```
android:fontFamily="sans-serif"
android:textStyle="bold"
```

- 在Java程序中设置字体

```
TextView tw1 = (TextView)findViewById(R.id.tw1);
tw1.setTypeface(Typeface.create("sans-serif",  Typeface.BOLD) );
```

设置宽度和高度

- **控件的高度和宽度**



layout_width

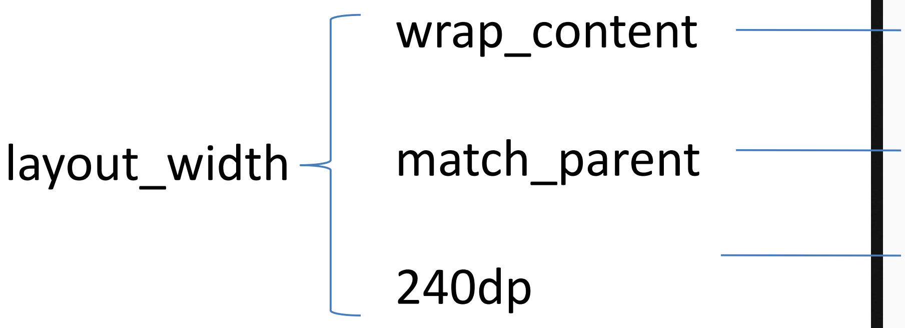
layout_height

控件的布局高度

padding

Hello！

layout_margin

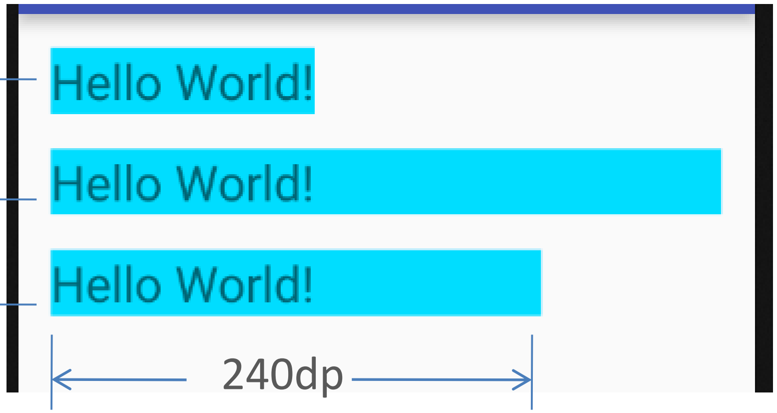


layout_width

wrap_content

match_parent

240dp



Hello World!

Hello World!

Hello World!

240dp

android:layout_width="wrap_content"

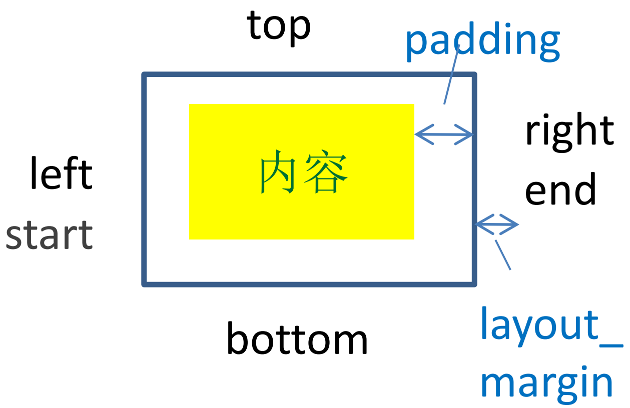
* fill_parent已过时

- **padding和layout_margin**

padding是内容与控件边界之间保留的空白，layout_margin为边界之外保留的空白。

padding
内边距

- android:padding （同时设置四个方向）
- android:paddingTop
- android:paddingBottom
- android:paddingLeft
- android:paddingRight
- android:paddingStart
- android:paddingEnd



layout_margin
外边距

- android:layout_margin （同时设置四个方向）
- android:layout_marginTop
- android:layout_marginBottom
- android:layout_marginLeft
- android:layout_marginRight
- android:layout_marginStart
- android:layout_marginEnd

* 在从左到右模式（默认）下，Start和Left一样，End和Right一样，在从右到左的模式下，Start和Right一样，End和Left一样。android:textDirection="rtl"设置从右到左。

- **width和layout_width的区别**
  - width只关注控件，而layout_width可以扩展到布局，所以，layout_width可以取值wrap_content和match_parent，而width只能取固定值(dp)。
  - 如果width和layout_width同时设置了，哪个会起作用呢？

    (1) 在layout_width设置为具体数值 width其实就无效了。
    (2) 在layout_width设置成wrap_content的时候
    - 如果设置了width，控件的宽度就取决于width
    - 如果没设置width，那么系统就会根据控件的内容来自行测量大小。

- **控件的最小宽度和最大宽度**

控件的宽度
- width
- maxWidth
- minWidth

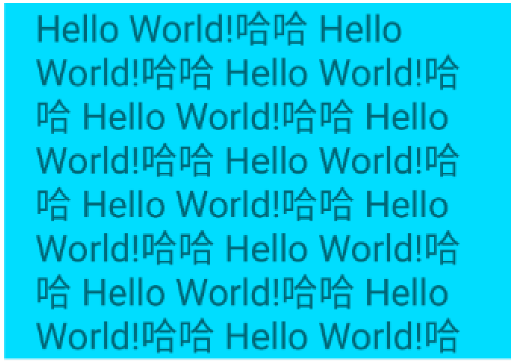
控件的高度
- height
- minHeight
- maxHeight

- width和layout_width都是整个控件的宽度。
- 当wrap_content并且内容太少或太多时控件宽度不会小于minWidth也不会大于maxWidth。内容太多时会导致自动折行（wrap）。
- 当wrap_content时，如果高度大于maxHeight，内容将不会被显示出来。

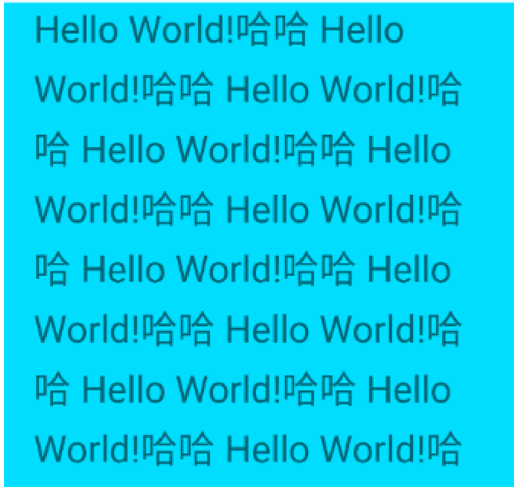
- 用行数设置控件的高度

lines	设置文本的行数。
maxLines	设置文本的最大显示行数，超出行数将不显示。
minLines	设置文本的最小行数。
lineSpacingExtra	行间距
lineSpacingMultiplier	设置行间距的倍数。如”1.2”
letterSpacing	字符间的空隙

```
android:layout_height="wrap_content"

android:lineSpacingExtra="0dp"                    "10dp"

android:lines="8"
```



Hello World!哈哈 Hello
World!哈哈 Hello World!哈
哈 Hello World!哈哈 Hello
World!哈哈 Hello World!哈
哈 Hello World!哈哈 Hello
World!哈哈 Hello World!哈
哈 Hello World!哈哈 Hello
World!哈哈 Hello World!哈



Hello World!哈哈 Hello
World!哈哈 Hello World!哈
哈 Hello World!哈哈 Hello
World!哈哈 Hello World!哈
哈 Hello World!哈哈 Hello
World!哈哈 Hello World!哈
哈 Hello World!哈哈 Hello
World!哈哈 Hello World!哈

- 用字符数设置控件的宽度

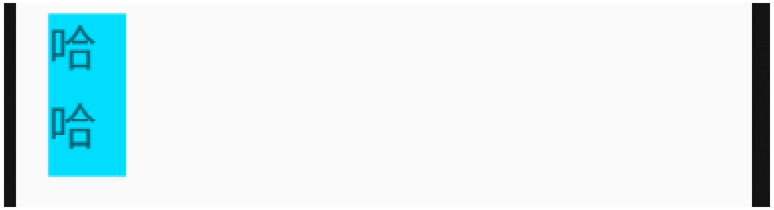
ems　设置TextView的宽度为N个字符的宽度。

maxEms　设置TextView的宽度为最长为N个字符的宽度。

minEms　设置TextView的宽度为最短为N个字符的宽度。

maxLength　限制显示的文本长度，超出部分不显示。

```
android:text="哈哈哈哈哈哈哈哈"
android:layout_width="wrap_content"
android:textSize="24sp"
android:ems="1"
```



哈
哈

- 动态设置控件的高度和宽度

```
TextView tv = (TextView) findViewById(R.id.tv);
RelativeLayout.LayoutParams  lp=new RelativeLayout.LayoutParams(
                      RelativeLayout.LayoutParams.WRAP_CONTENT,  // width
                      RelativeLayout.LayoutParams.WRAP_CONTENT); // height
lp.setMargins(0,20,0,0);
tv.setLayoutParams(lp);
tv.setText("Hello!");
```

可以设置具体值

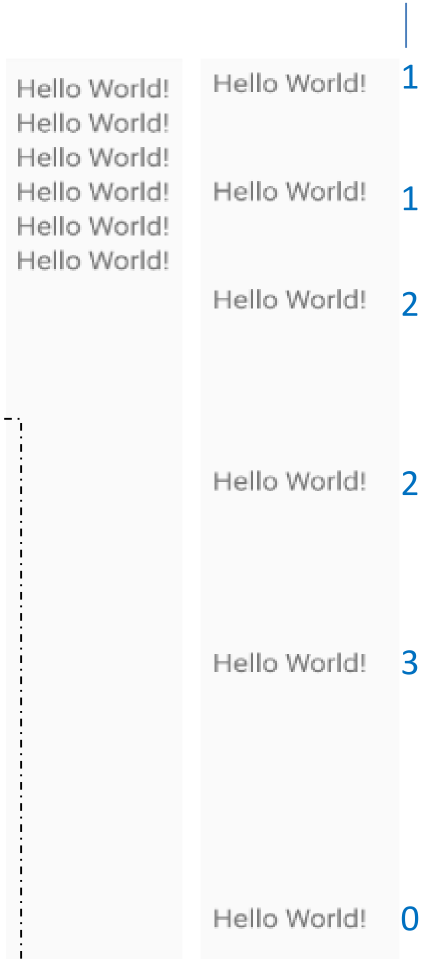
获得控件高度：参考1 参考2

布局比重

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android=http://schemas.android.com/apk/res/android
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Hello World!" />
    <TextView
        android:layout_weight="1"
        android:text="Hello World!" />
    <TextView
        android:layout_weight="2"
        android:text="Hello World!" />
    <TextView
        android:layout_weight="2"
        android:text="Hello World!" />
    <TextView
        android:layout_weight="3"
        android:text="Hello World!" />
    <TextView
        android:text="Hello World!" />
</LinearLayout>
```

- 省略了layout_width和layout_height，与第一个TextView一样。
- layout_weight根据父控件的余留空白按比例进行分配。这里是分配高度。
- Layout可以设置一个总比重，以确定未设置的layout_weight值，例如，android:weightSum=5

android:layout_weight



Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!

Hello World!
Hello World!
Hello World!
Hello World!
Hello World!
Hello World!

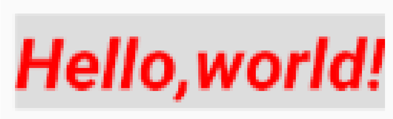
没有weight或weight等于0

按比例分配高度

文本设置

- **设置文本显示样式**

很多控件都可以显示文字，例如：TextView（文本框），EditView（编辑框），CheckBox（复选框），RadioButton（单选按钮），Button（按钮）等都有文字显示，下面是设置它们样式的基本属性。

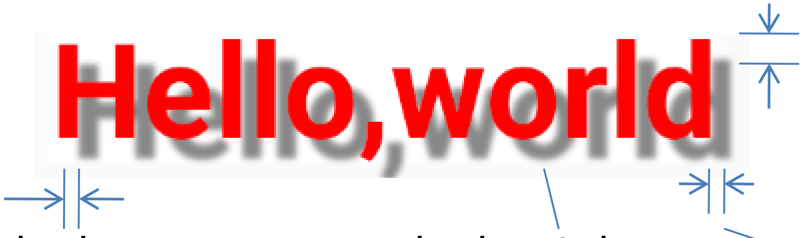


Hello,world!

`android:text="Hello,world!"`	设置文本内容
`android:textColor="#FF0000"`	设置文字颜色
`android:textSize="24f"`	设置文字大小(sp)
`android:textStyle="bold|italic"`	设置文字样式
`android:background="#DDD"`	设置背景颜色

```
myTextView.setTextSize(12f);   //浮点数，单位sp
myTextView.setTextSize(TypedValue.COMPLEX_UNIT_SP,12);  //单位sp
myTextView.setTextSize(TypedValue.COMPLEX_UNIT_DIP,12); //单位dp
myTextView.setTextSize(TypedValue.COMPLEX_UNIT_PX,12);  //单位px
```

- 文本阴影



shadowDy
阴影的垂直偏移量

shadowDx
阴影的水平偏移量

shadowColor
文本的阴影颜色

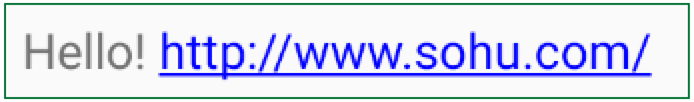
shadowRadius
文本阴影的模糊半径

```
<TextView
    android:text="Hello,world!"
    android:textColor="#FF0000"
    android:textSize="64sp"
    android:textStyle="bold"
    android:shadowDx="30"
    android:shadowDy="30"
    android:shadowColor="#888"
    android:shadowRadius="10"
    android:background="#FFF" />
```

设置阴影

- 文本中的链接

```
android:autoLink="web"
android:linksClickable="true"
android:textColorLink="#0000FF"
android:text="Hello! http://www.sohu.com/"
```



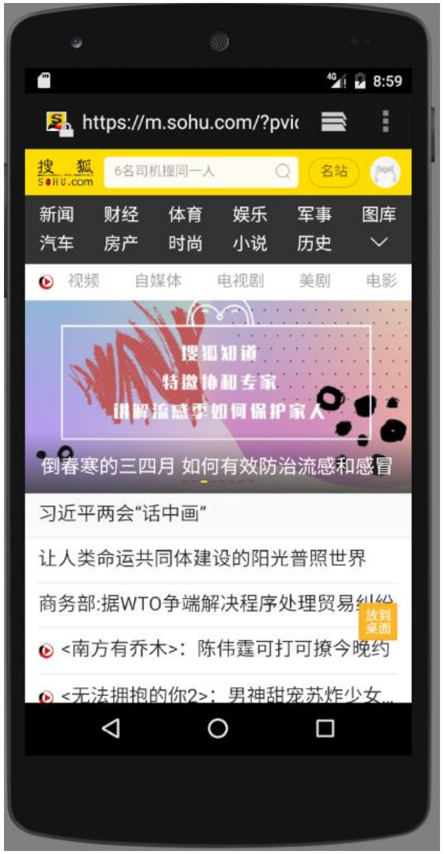
Hello! http://www.sohu.com/

linksClickable　　设置链接是否可以点击。默认可点击

autoLink　　自动找到地址并显示链接：

web|email|phone|map|all|none

textColorLink　　文本中链接的颜色



倒春寒的三四月 如何有效防治流感和感冒

习近平两会"话中画"

让人类命运共同体建设的阳光普照世界

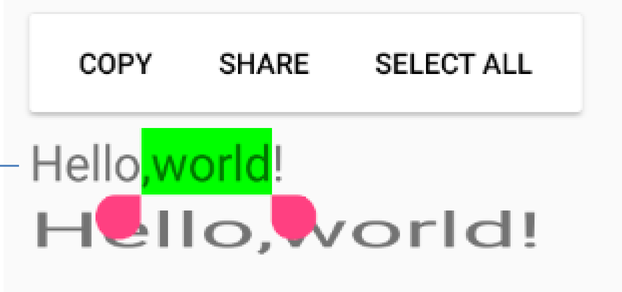
商务部:据WTO争端解决程序处理贸易纠纷

<南方有乔木>：陈伟霆可打可撩今晚约

<无法拥抱的你2>：男神甜宠苏炸少女...

- 选择文本的颜色和全部大写字母

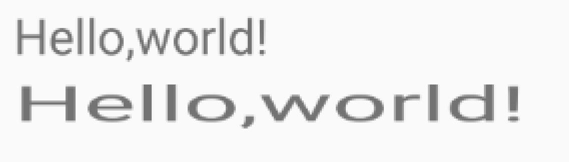
```
<TextView
    android:textSize="24sp"
    android:text="Hello,world!"
    android:textAllCaps="true"
    android:textIsSelectable="true"
    android:textColorHighlight="#0F0"/>
```



textAllCaps	文本是否全部变为大写（true,false）
textIsSelectable	是否可以选择文本
textColorHighlight	选择的文本的颜色

- x方向放大文本的倍数

```
<TextView
    android:textSize="24sp"
    android:text="Hello,world!"
    android:textScaleX="2" />
```



x方向放大文本的倍数

文本对齐

gravity(重心)用于控件内容的对齐。

- 取值： top、bottom、left、right、center_vertical、center、fill_vertical、center_horizontal、fill_horizontal、center、fill、clip_vertical 。（参见FrameLayout）

- 默认取值位top、left。
- clip剪切掉超出部分；fill增大控件，直到把内容填满控件。

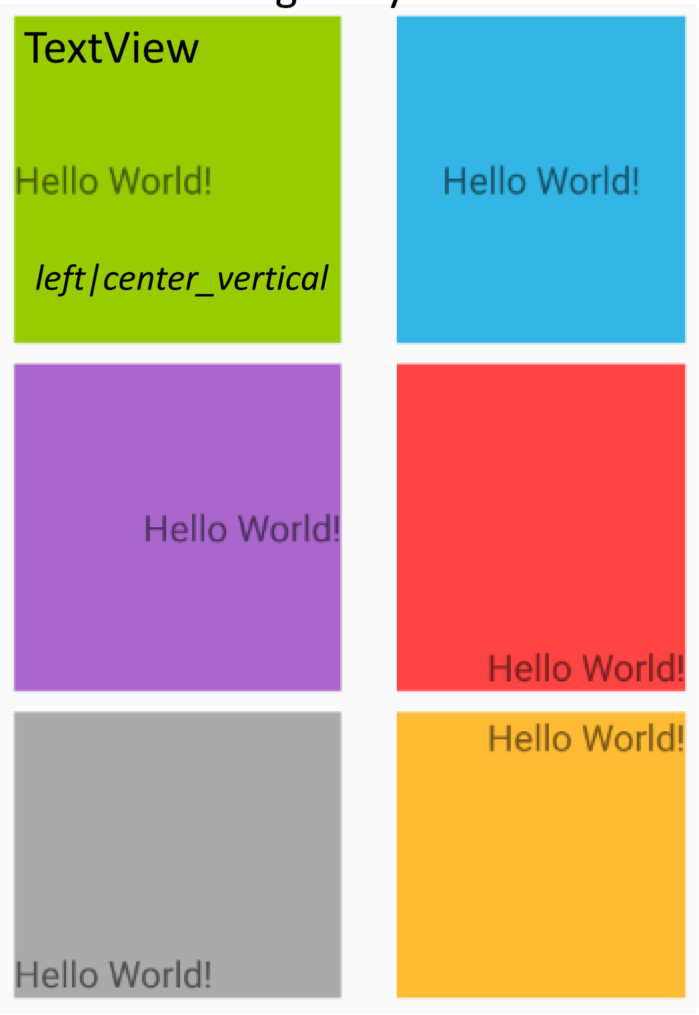
```
android:gravity="top|right"
```

textAlignment也可以用于控件中的文本对齐。

- 取值： inherit、textStart、textEnd、center、viewStart、viewEnd、gravity。
- 取值为gravity时gravity起作用，设置值与gravity有矛盾时本属性起作用。
- view对齐控件，text对齐文本。

```
android:textAlignment="textEnd"
```

gravity



* android:layout_gravity用于在父元素中对齐，具体的建下一章的FrameLayout

gravity取值：

值	说明
top	将对象放在其容器顶部，不改变其大小。
bottom	将对象放在其容器底部，不改变其大小。
left	将对象放在其容器左边缘，不改变其大小。
right	将对象放在其容器右边缘，不改变其大小。
center_vertical	将对象放在其容器的垂直中心，不改变其大小。
fill_vertical	按需要扩展对象的垂直大小，使其完全适应其容器。
center_horizontal	将对象放在其容器的水平中心，不改变其大小。
fill_horizontal	按需要扩展对象的水平大小，使其完全适应其容器。
center	将对象放在其容器的水平和垂直轴中心，不改变其大小。
fill	按需要扩展对象的垂直大小，使其完全适应其容器。这是默认值。
clip_vertical	可设置为让子元素的上边缘和/或下边缘裁剪至其容器边界的附加选项。裁剪基于垂直重力：顶部重力裁剪上边缘，底部重力裁剪下边缘，任一重力不会同时裁剪两边。
clip_horizontal	可设置为让子元素的左边和/或右边裁剪至其容器边界的附加选项。裁剪基于水平重力：左边重力裁剪右边缘，右边重力裁剪左边缘，任一重力不会同时裁剪两边。

Toast显示定位

- setGravity(int gravity, int xOffset, int yOffset)
- setMargin(float horiMargin, float vertMargin)

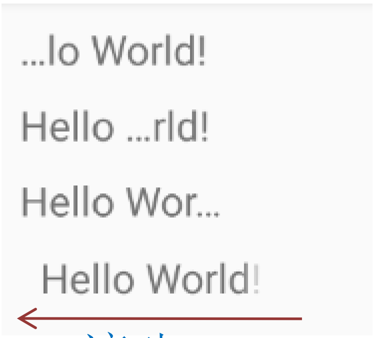
省略文本

ellipsize设置当因宽度限制而文字过长时,该控件该如何显示。

取值：none（截断）|marquee（跑马灯模式，聚焦时）|start（省略号放在前面）|end（省略号放在后面）|middle（省略号放在中间）

marqueeRepeatLimit 在ellipsize指定marquee的情况下，设置重复滚动的次数。

```
<TextView
    android:text="Hello World! "
    android:layout_width="100dp"
    android:ellipsize="start" />
<TextView
    android:text="Hello World! "
    android:layout_width="100dp"
    android:ellipsize="middle"/>
<TextView
    android:text="Hello World! "
    android:layout_width="100dp"
    android:ellipsize="end"/>
<TextView
    android:text="Hello World! a b c d e f"
    android:ellipsize="marquee"
    android:layout_width="120dp"
    android:marqueeRepeatLimit="marquee_forever"/>
```



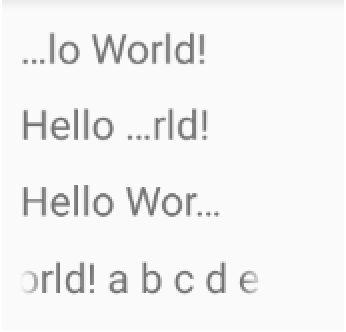
...lo World!

Hello ...rld!

Hello Wor...

Hello World!

滚动



...lo World!

Hello ...rld!

Hello Wor...

orld! a b c d e

上面的**TextView**都省略了：`android:maxLines="1"`

跑马灯模式必须采用编程方式实现才有效：

```
TextView textView=(TextView)findViewById(R.id.TextView4);
textView.setEllipsize(TextUtils.TruncateAt.MARQUEE);
textView.setSingleLine(true);
textView.setSelected(true);
textView.setFocusable(true);
textView.setFocusableInTouchMode(true);
```

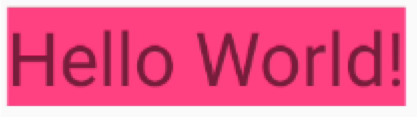
• selected	是否可以选择文本
• focusable	是否可获得焦点(聚焦)
• focusableInTouchMode	在触摸方式下是否可获得焦点

当具有focusableInTouchMode属性的控件聚焦时，如果另一个控件没有此属性，则它可以保持聚焦

背景设置

- **背景颜色**

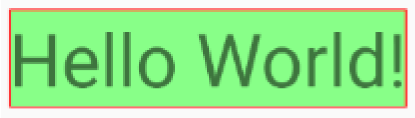
```
android:background=″@color/colorAccent″
tv.setBackgroundColor(Color.argb(255,0,0,255));
```


Hello World!

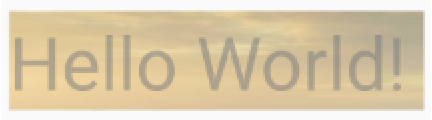
- **背景图像和透明度 (alpha:0~1)**

```
android:background=″@drawable/shape″
```

加边框


Hello World!

```
android:background=″@drawable/bk″
android:alpha=″0.5″
android:layout_height=″48dp″
```


Hello World!

透明度: 0~1
1表示完全不透明

res\drawable\bk.png

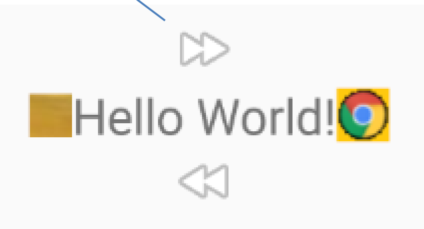
drawable\shape.xml

```
<?xml version=″1.0″ encoding=″UTF-8″?>
<shape xmlns:android=″http://schemas.android.com/apk/res/android″>
    <solid android:color=″#8F8″/>
    <stroke android:width=″1dip″ android:color=″#F00″ />
</shape>
```

命名空间，可提示输入

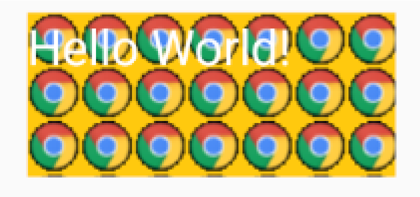
- **背景图的位置**

安卓内置图像



Hello World!

```
<TextView
    android:text="Hello World!"
    android:drawableLeft="@drawable/bk1"
    android:drawableRight="@drawable/bk2"
    android:drawableTop="@android:drawable/ic_media_ff"
    android:drawableBottom="@android:drawable/ic_media_rew" />
```



Hello World!

```
<TextView
    android:text="Hello World!"
    android:background="@drawable/bk"
    android:textColor="@android:color/background_light" />
```

res\drawable\bk1.png res\drawable\bk2.png

drawable\bk.xml

```
<?xml version="1.0" encoding="utf-8"?>
<bitmap xmlns:android="http://schemas.android.com/apk/res/android"
    android:src="@drawable/bk2"
    android:tileMode="repeat" />
```

基本控件

切换按钮(ToggleButton)

- android:textOff=""
- android:textOn=""
- android:checked="true"
- setOnCheckedChangeListener()

开关按钮(Switch)

- android:thumb
- android:track
- android:switchMinWidth
- android:checked
- setOnCheckedChangeListener()

复选框(CheckBox)

- android:checked
- setOnCheckedChangeListener()

单选按钮(RadioButton)

- RadioGroup
- radiogroup.setOnCheckedChangeListener()
- DisplayToast()

可选文本框(CheckedTextView)

- android:checked
- android:clickable
- android:checkMark

评价条(RatingBar)

- android:stepSize
- android:numStars
- android:rating
- setOnRatingBarChangeListener()

拖动条(SeekBar)

- android:max
- android:progress
- setOnSeekBarChangeListener()

参考

进度条(ProgressBar)

android:max

android:progress

android:secondaryProgress

android:progressDrawable

android:indeterminate

android:indeterminateDrawable

shape

CirclePgBar（自定义控件）

图像框(ImageView)

ImageButton

android:src　android:scaleType

app:srcCompat

android:background

编辑框(EditText)

android:inputType

(textPassword textMultiLine

textAutoComplete)

android:hint

android:textColorHint

android:textCursorDrawable

android:cursorVisible

android:selectAllOnFocus

android:enabled

联系人控件(QuickContactBadge)

assignContactFromPhone()

日期选择器(DatePicker)

setMinDate()

setMaxDate()

init()

OnDateChangedListener()

时间选择器(TimePicker)

setIs24HourView()

setCurrentHour()

setCurrentMinute()

setHour()

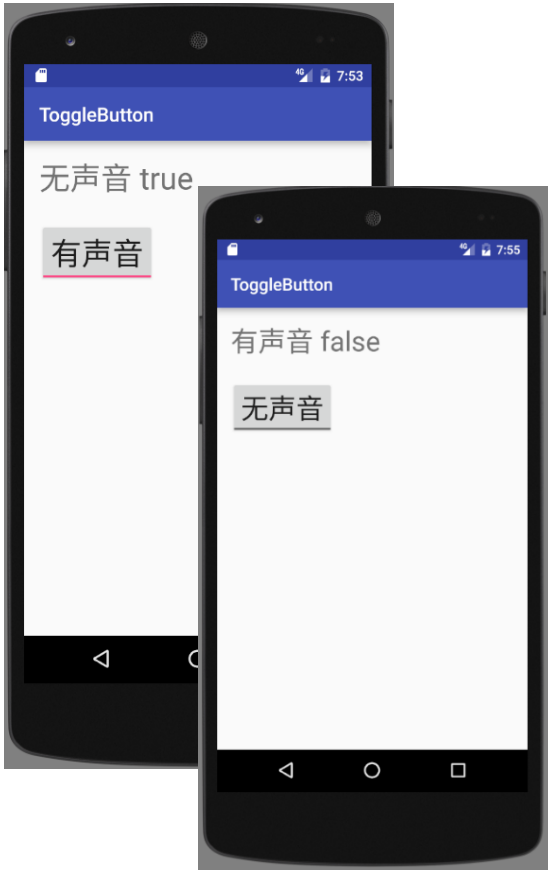
setMinute()

setOnTimeChangedListener()

切换按钮 (ToggleButton)

参考

```
<?xml version=”1.0” encoding=”utf-8”?>
<RelativeLayout xmlns:android=”http://schemas.android.com/apk/res/android”
    xmlns:tools=http://schemas.android.com/tools
    tools:context=”com.example.isszym.togglebutton.MainActivity”
    android:id=”@+id/activity_main”>

    <TextView
        android:text=”Hello World!”
        android:id=”@+id/textView” />

    <ToggleButton
        android:text=”ToggleButton”
        android:textOff=””
        android:textOn=””
        android:checked=”true”
        android:id=”@+id/toggleButton” />
</RelativeLayout>
```



无声音 true

有声音 false

*getText()得到的显示文本是状态变化前的

```
public class MainActivity extends AppCompatActivity {
    ToggleButton toggleButton1;
    TextView textView1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        toggleButton1=(ToggleButton)findViewById(R.id.toggleButton);
        textView1=(TextView)findViewById(R.id.textView);
        toggleButton1.setChecked(true);
        toggleButton1.setTextOff("无声音");
        toggleButton1.setTextOn("有声音");
        toggleButton1.setOnCheckedChangeListener(
           new CompoundButton.OnCheckedChangeListener() {
             @Override
             public void onCheckedChanged(CompoundButton v, boolean b) {
                 textView1.setText(toggleButton1.getText()
                                  +" "+(toggleButton1.isChecked()?"true":"false"));
             }
         });
    }
}
```

onCheckedChanged的参数：v就是toggleButton1
b就是toggleButton1.isChecked()

为什么toggleButton1定义在onCreate中必须定义为final的？

```
public abstract class CompoundButton extends Button implements Checkable {
     private boolean mChecked;
        ...
     public CompoundButton(Context context) {
                   this(context, null);
         }
    public void toggle() {
        throw new RuntimeException("Stub!");
    }

    @ExportedProperty
    public boolean isChecked() {
       setChecked(!mChecked);
    }

    public void setChecked(boolean checked) {
         if (mChecked != checked) {
              mChecked = checked;
               ...
         }
    }

    public void setOnCheckedChangeListener(CompoundButton.OnCheckedChangeListener listener){
        ...
    }

    public interface OnCheckedChangeListener {
        void onCheckedChanged(CompoundButton var1, boolean var2);
    }
}
```

* 完整定义参见附录6

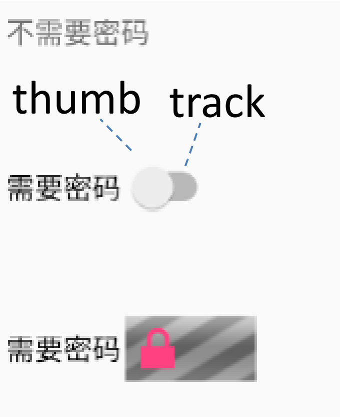
```java
public class ToggleButton extends CompoundButton {
    private CharSequence mTextOn;
    private CharSequence mTextOff;
    private Drawable mIndicatorDrawable;
    private static final int NO_ALPHA = 0xFF;
    private float mDisabledAlpha;

    public ToggleButton(Context context) {
        this(context, null);
    }
    ......
    @Override
    public void setChecked(boolean checked) {
        super.setChecked(checked);
        syncTextState();
    }
    public CharSequence getTextOn() {
        return mTextOn;
    }
    public void setTextOn(CharSequence textOn) {
        mTextOn = textOn;
    }
    public CharSequence getTextOff() {
        return mTextOff;
    }
    @Override
    public void setBackgroundDrawable(Drawable d) {
        super.setBackgroundDrawable(d);
        updateReferenceToIndicatorDrawable(d);
    }
}
```

* 完整定义参见附录6

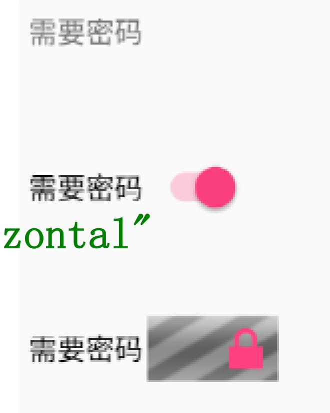
开关按钮 (Switch)

参考

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android=http://schemas.android.com/apk/res/android
    android:id="@+id/activity_main">
    <TextView
        android:text="Hello World!"
        android:id="@+id/textView" />

    <Switch
        android:text="需要密码"
        android:id="@+id/switch1"
        android:checked="false" />

    <Switch
        android:text="需要密码"
        android:id="@+id/switch2"
        android:thumb="@android:drawable/ic_lock_lock"
        android:track="@android:drawable/progress_indeterminate_horizontal"
        android:switchMinWidth="60dp"
        android:checked="false" />
</RelativeLayout>
```

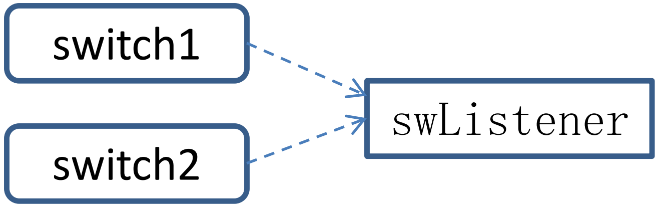


两个开关同时开关



```
public class MainActivity extends AppCompatActivity {
    TextView textview1;  Switch switch1;   Switch switch2;
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        textview1=(TextView)findViewById(R.id.textView);
        switch1=(Switch)findViewById(R.id.switch1);
        switch2=(Switch)findViewById(R.id.switch2);
        Switch.OnCheckedChangeListener swListener=
                                    new Switch.OnCheckedChangeListener(){
            public void onCheckedChanged(CompoundButton btn, boolean b){
                if(btn.getId()==R.id.switch1){
                    switch2.setChecked(switch1.isChecked());
                } else {
                    switch1.setChecked(switch2.isChecked());
                }
                if(switch1.isChecked())
                    textview1.setText("需要密码");
                else
                    textview1.setText("不需要密码");
            }
        };
        switch1.setOnCheckedChangeListener(swListener);
        switch2.setOnCheckedChangeListener(swListener);
        switch1.setChecked(true);
    }
}
```

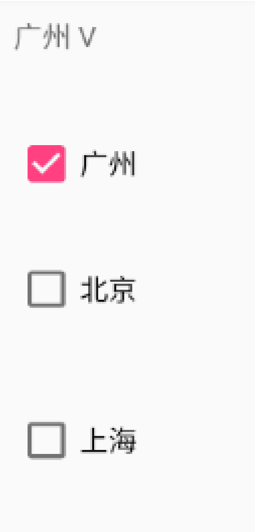
参数btn为事件对象，b为btn.isChekced()



复选框(CheckBox)

参考

```
<?xml version=”1.0” encoding=”utf-8”?>
<RelativeLayout xmlns:android
     =http://schemas.android.com/apk/res/android
    android:id=”@+id/activity_main”>
    <TextView
        android:text=”Hello World!”
        android:id=”@+id/textView” />
    <CheckBox
        android:text=”广州”
        android:checked=”false”
        android:id=”@+id/checkBox1” />
    <CheckBox
        android:text=”北京”
        android:id=”@+id/checkBox2”
        android:checked=”false” />
    <CheckBox
        android:text=”上海”
        android:checked=”false”
        android:id=”@+id/checkBox3” />
</RelativeLayout>
```

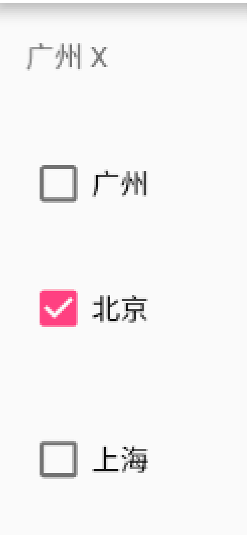


广州 V

☑ 广州

□ 北京

□ 上海



广州 X

□ 广州

☑ 北京

□ 上海

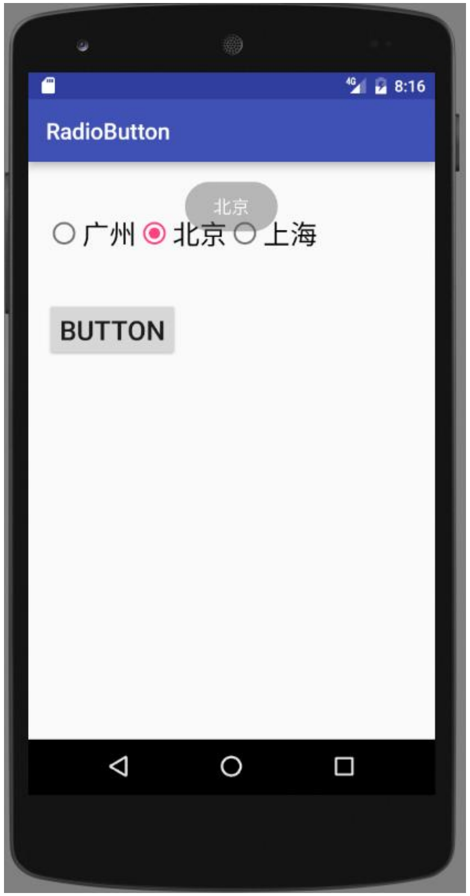
```java
public class MainActivity extends AppCompatActivity {
    CheckBox chk[];
    TextView tv1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tv1=(TextView)findViewById(R.id.textView);
        chk = new CheckBox[3];
        chk[0]=(CheckBox) findViewById(R.id.checkBox1);
        chk[1]=(CheckBox) findViewById(R.id.checkBox2);
        chk[2]=(CheckBox) findViewById(R.id.checkBox3);
        CheckBox.OnCheckedChangeListener cbListener
            = new CheckBox.OnCheckedChangeListener(){
          public void onCheckedChanged(CompoundButton btn, boolean b){
              tv1.setText(btn.isChecked()?(btn.getText()+” V”)
                                           :(btn.getText()+” X”));
          }
        };
        for(int i=0;i<3;i++){
            chk[i].setChecked(false);
            chk[i].setOnCheckedChangeListener(cbListener);
        }
    }
}
```

单选按钮 (RadioButton)

参考 参考

activity_main.xml

```
<?xml version=″1.0″ encoding=″utf-8″?>
<RelativeLayout xmlns:android=http://schemas.android.com/apk/res/android
    android:id=″@+id/activity_main″>
    <RadioGroup
        android:layout_weight=″1″
        android:id=″@+id/radiogroup1″
        android:orientation=″horizontal″
        android:layout_x=″3px″>
        <RadioButton
            android:text=″广州″
            android:id=″@+id/radioButton1″/>
        <RadioButton
            android:text=″北京″
            android:id=″@+id/radioButton2″ />
        <RadioButton
            android:text=″上海″
            android:id=″@+id/radioButton3″ />
    </RadioGroup>
    <Button
        android:text=″Button″
        android:id=″@+id/button″ />
</RelativeLayout>
```

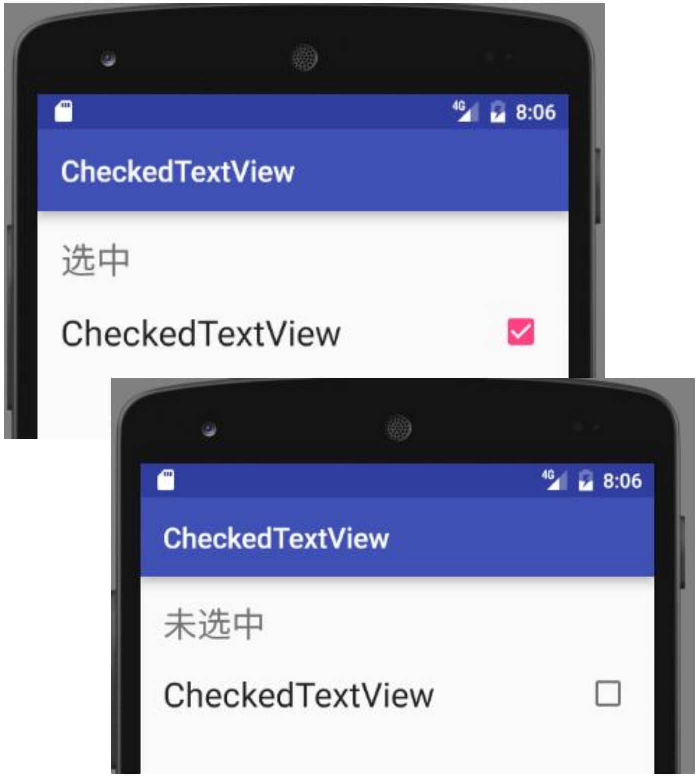


```
public class MainActivity extends AppCompatActivity {
    RadioGroup radiogroup;  Button btn;
    protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      btn=(Button)findViewById(R.id.button);
      radiogroup=(RadioGroup)findViewById(R.id.radiogroup1);
      btn.setOnClickListener(new Button.OnClickListener(){
        public void onClick(View vw){
          RadioButton rb =
                (RadioButton) findViewById(radiogroup.getCheckedRadioButtonId());
                DisplayToast(rb.getText().toString());
            }
      });
      radiogroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener(){
        public void onCheckedChanged(RadioGroup group, int checkedId) {
               RadioButton rb = (RadioButton) findViewById(checkedId);
               DisplayToast(rb.getText().toString());
        }
      });
    }
    public void DisplayToast(String str){
        Toast toast= Toast.makeText(this, str, Toast.LENGTH_SHORT);
        toast.setGravity(Gravity.TOP,0,220);    toast.show();
    }
}
```

可选文本框 (CheckedTextView)

参考

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android=http://schemas.android.com/apk/res/android
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        android:id="@+id/textView"/>

    <CheckedTextView
        android:text="CheckedTextView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/checkedTextView"
        android:checked="false"
        android:clickable="true"
        android:checkMark="?android:attr/listChoiceIndicatorMultiple"/>
</RelativeLayout>
```



CheckedTextView

选中

CheckedTextView ☑

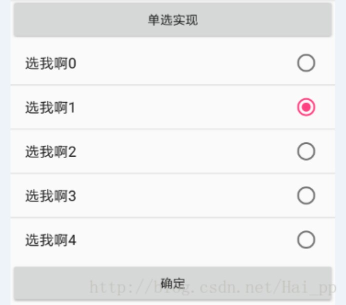
CheckedTextView

未选中

CheckedTextView □

listChoiceIndicatorSingle(单选)

```java
public class MainActivity extends AppCompatActivity {
    TextView tv1;
    CheckedTextView checkedTextView1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tv1=(TextView)findViewById(R.id.textView);
        checkedTextView1=(CheckedTextView)findViewById(R.id.checkedTextView);
        checkedTextView1.setChecked(true);
        tv1.setText("选中");
        checkedTextView1.setOnClickListener(new CheckedTextView.OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                checkedTextView1.toggle();
                tv1.setText(checkedTextView1.isChecked()?"选中":"未选中");
            }
        });
    }
}
```



单选实现

选我啊0 ○

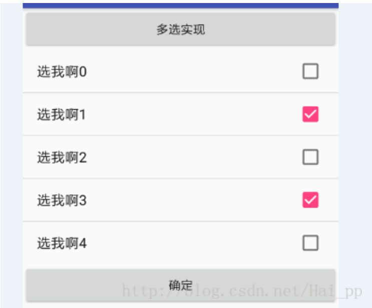
选我啊1 ●

选我啊2 ○

选我啊3 ○

选我啊4 ○

确定



多选实现

选我啊0 ☐

选我啊1 ☑

选我啊2 ☐

选我啊3 ☑

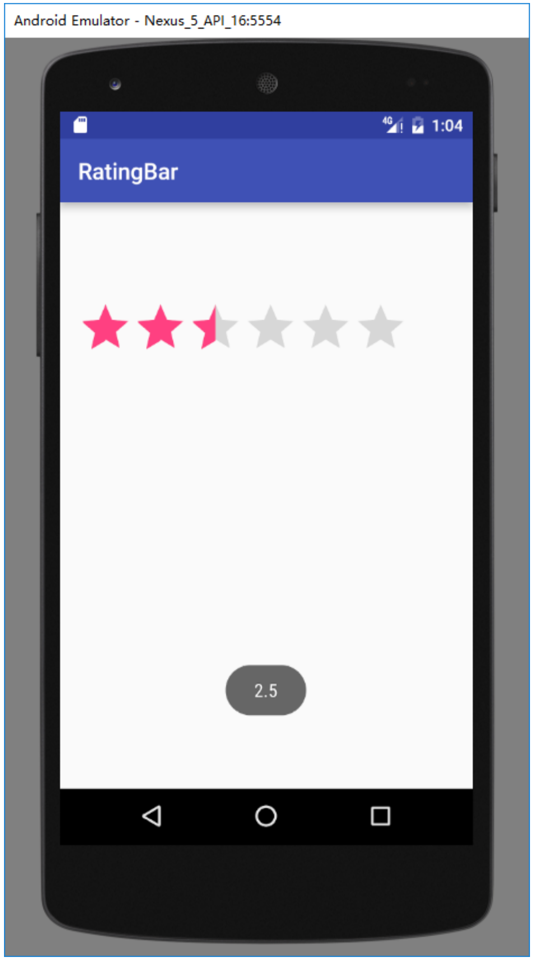
选我啊4 ☐

确定

评价条(RatingBar)

参考

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <RatingBar
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/ratingBar"
        android:stepSize="0.5"
        android:numStars="6"
        android:rating="4" />
</RelativeLayout>
```



初始化后点击了一次

```java
public class MainActivity extends AppCompatActivity {
    RatingBar ratingBar1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ratingBar1 = (RatingBar) findViewById(R.id.ratingBar);
        ratingBar1.setRating(4);
        ratingBar1.setOnRatingBarChangeListener(
             new RatingBar.OnRatingBarChangeListener() {
              // 第三个参数 如果评分改变是由用户触摸手势或方向键轨迹球移动触发的,
              //  则返回true
            public void onRatingChanged(RatingBar ratingBar, float rating,
                                        boolean paramBoolean){
                Toast.makeText(MainActivity.this,""+ratingBar1.getRating(),
                                Toast.LENGTH_SHORT).show();

            }
        });
    }
}
```

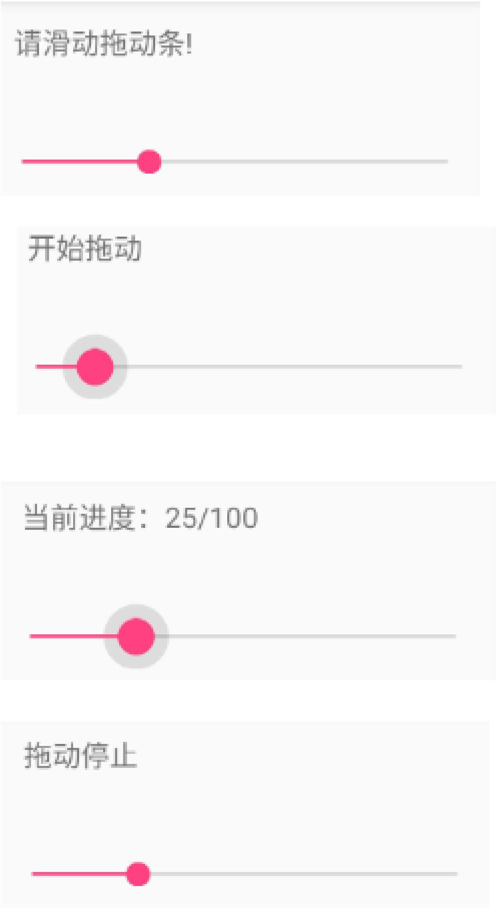
* ratingBar1.getRating()、参数rating、ratingBar.getRating()都得到相同的值。

拖动条(SeekBar)

参考

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android=http://schemas.android.com/apk/res/android
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="请滑动拖动条!"
        android:id="@+id/textView" />

    <SeekBar
        android:layout_height="wrap_content"
        android:layout_width="240dp"
        android:max="100"
        android:progress="30"
        android:id="@+id/seekBar"  />

</RelativeLayout>
```

拖动条



请滑动拖动条!

开始拖动

当前进度：25/100

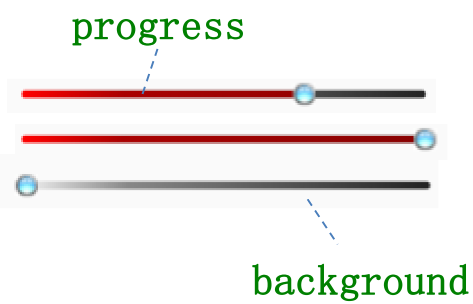
拖动停止

```java
public class MainActivity extends AppCompatActivity {
    private SeekBar seekBar1;      private TextView tv1;
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        seekBar1 = (SeekBar) findViewById(R.id.seekBar);
        seekBar1.setProgress(30);
        tv1 = (TextView) findViewById(R.id.textView);
        seekBar1.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {
                tv1.setText(“拖动停止”);
            }
            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {
                tv1.setText(“开始拖动”);
            }
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress,
                                          boolean fromUser) {
                tv1.setText(“当前进度：” + seekBar1.getProgress()
                                    + “/” + seekBar1.getMax());
            }
        });
    }
}
```

- 自定义拖动条控件的按钮、背景和进度条

```
<SeekBar
    android:progressDrawable="@drawable/seekbar_light"
    android:thumb="@drawable/submit"
    android:maxHeight="4dp"
    android:minHeight="4dp"
    android:max="10"
    android:progress="7"
 />
```

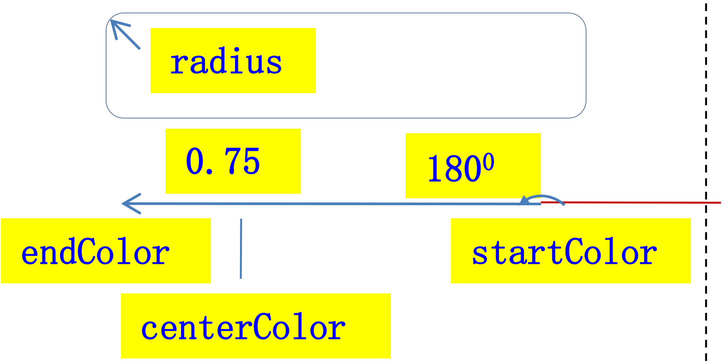
drawable\submit.png



drawable\seekbar_light.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@android:id/background">
        <shape>
            <corners android:radius="2dip" />
            <gradient
                android:angle="180"
                android:startColor="#222"
                android:centerColor="#888"
                android:centerX="0.75"
                android:endColor="#EEE" />
        </shape>
    </item>
```

圆角和渐变(gradient)



```
 <item android:id=″@android:id/progress″>
        <clip>
            <shape>
                <corners android:radius=″2dip″ />
                <gradient
                    android:angle=″180″
                    android:startColor=″#800″
                    android:centerColor=″#A00″
                    android:centerX=″0.75″
                    android:endColor=″#F00″ />
            </shape>
        </clip>
    </item>
</layer-list>
```

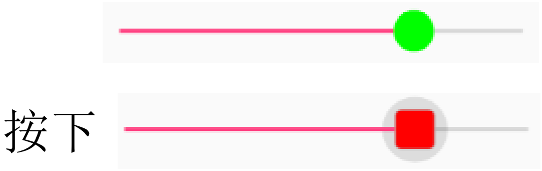
*clip表示剪切一段，不是整条线

四个圆角分别定义：

```
<corners
   android:topLeftRadius=″1dp″
   android:topRightRadius=″2dp″
   android:bottomLeftRadius=″0dp″
   android:bottomRightRadius=″3dp″ />
```

- 自定义拖动条控件的按钮（放开和按下）

```
<SeekBar
    android:thumb=″@drawable/seekbar_thumb″
 />
```



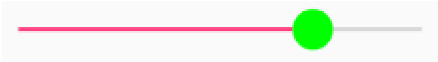
drawable\seekbar_thumb.xml

```
<?xml version=″1.0″ encoding=″utf-8″?>
<selector xmlns:android=″http://schemas.android.com/apk/res/android″>
    <item android:drawable=″@drawable/rectangle″ android:state_pressed=″true″ />
    <item android:drawable=″@drawable/circle″ android:state_pressed=″false″ />
</selector>
```

＊ 另一个可用的选项：android:state_focused=″false″

drawable/circle.xml

```
<?xml version=″1.0″ encoding=″utf-8″?>
<shape xmlns:android=″http://schemas.android.com/apk/res/android″
  android:shape=″oval″>

  <solid android:color=″#FF00FF00″/>
  <stroke android:width=″1dp″
          android:color=″#FF00FF00″/>
  <size android:width=″20dp″
        android:height=″20dp″/>
</shape>
```

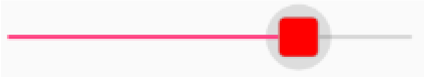


drawable/rectangle.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
    <solid android:color="#FFFF0000" />
    <stroke
        android:width="1dp"
        android:color="#20000000" />
    <corners android:radius="3dp"/>
    <size
        android:width="20dp"
        android:height="14dp" />
</shape>
```

虚线

```
android:dashWidth="5dp"
android:dashGap="3dp"
```

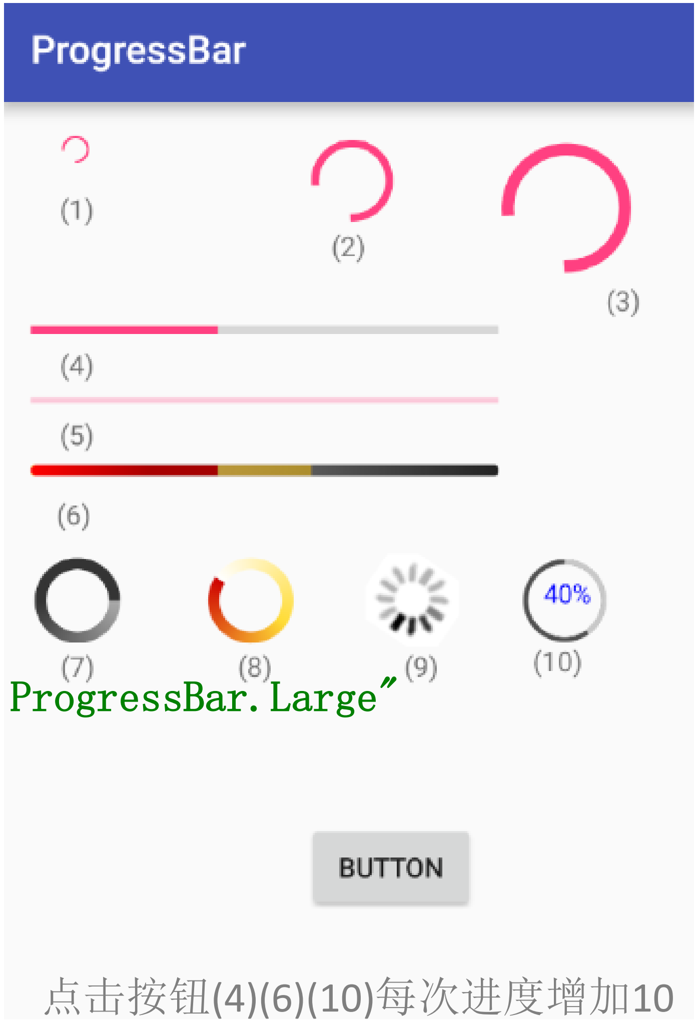


* shape的详细内容请见附录

进度条(ProgressBar)

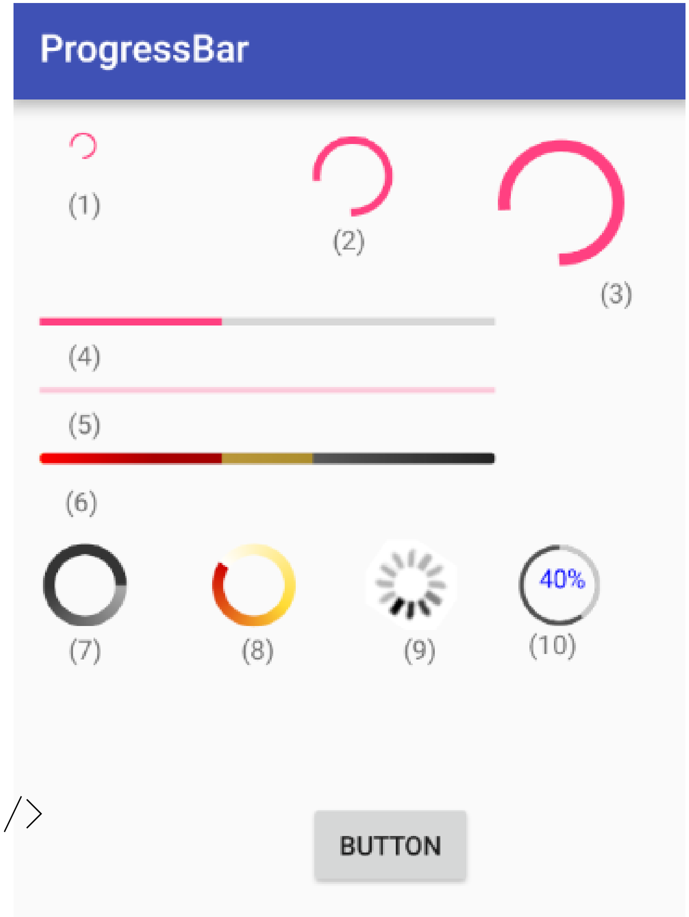
参考

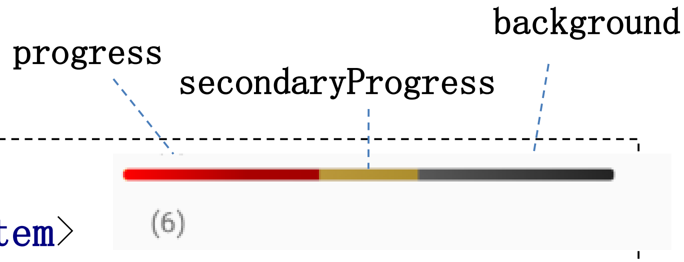
```
<ProgressBar
    android:id=”@+id/progressBar1”
    style=”?android:attr/progressBarStyleSmall”
    android:max=”100”
    android:progress=”80” />
<ProgressBar
    android:id=”@+id/progressBar2”
    android:indeterminate=”false”
    android:max=”100”
    android:progress=”80”
    style=”@style/Widget.AppCompat.ProgressBar” />
<ProgressBar
    android:id=”@+id/progressBar3”
    style=”@android:style/Widget.DeviceDefault.Light.ProgressBar.Large”
    android:max=”100”
    android:progress=”80” />
<ProgressBar
    android:id=”@+id/progressBar4”
    style=”?android:attr/progressBarStyleHorizontal”
    android:max=”100”
    android:indeterminate=”false”
    android:progress=”40” />
```



点击按钮(4)(6)(10)每次进度增加10
(1)(2)(3)(5)(8)是不确定进度的
(7)固定显示一个环形

```
<ProgressBar
    android:id="@+id/progressBar5"
    style="?android:attr/progressBarStyleHorizontal"
    android:max="100"
    android:progress="40"
    android:indeterminate="true"/>
<ProgressBar
    android:id="@+id/progressBar6"
    style="@style/ProgressHorizontal"
    android:max="100"
    android:progress="40"
    android:secondaryProgress="60" />
<ProgressBar
    android:id="@+id/progressBar7"
    android:indeterminate="false"
    android:indeterminateDrawable="@drawable/ring"/>
<ProgressBar
    android:id="@+id/progressBar8"
    android:indeterminateDrawable="@drawable/rotate"/>
<ProgressBar
    android:id="@+id/progressBar9"
    android:indeterminateDrawable="@drawable/loadingpng"/>
<com.example.isszym.progressbar.CirclePgBar
    android:id="@+id/progressBar10" />
<Button android:id="@+id/button"
        android:text="Button"/>
```



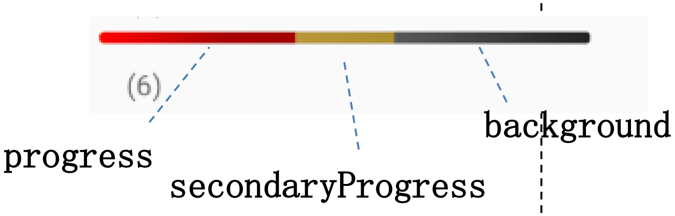


styles.xml

```
<style name="ProgressHorizontal">
    <item name="android:indeterminateOnly">false</item>
    <!-- 进度条的背景，progress ,secondaryProgress 的颜色-->
    <item name="android:progressDrawable">@drawable/pbar_light</item>
    <!--高度-->
    <item name="android:minHeight">10dip</item>
    <item name="android:maxHeight">10dip</item>
</style>
```

drawable\pbar_light.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@android:id/background">
        <shape>
            <corners android:radius="2dip" />
            <gradient
                android:angle="180"
                android:startColor="#222"
                android:centerColor="#888"
                android:centerX="0.75"
                android:endColor="#EEE" />
        </shape>
    </item>
    <item android:id=″@android:id/secondaryProgress″>
        <clip>
            <shape>
                <corners android:radius=″2dip″ />
                <gradient
                    android:angle=″180″
                    android:startColor=″#80ffd300″
                    android:centerColor=″#80ffb600″
                    android:centerX=″0.75″
                    android:endColor=″#a0ffcb00″ />
            </shape>
        </clip>
    </item>
    <item android:id=″@android:id/progress″>
        <clip>
            <shape>
                <corners android:radius=″5dip″ />
                <gradient
                    android:angle=″180″
                    android:startColor=″#800″
                    android:centerColor=″#A00″
                    android:centerX=″0.75″
                    android:endColor=″#F00″/>
            </shape>
        </clip>
    </item>
</layer-list>
```



progress

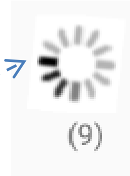
secondaryProgress

background

drawable\loadingpng.xml

```
<?xml version=”1.0” encoding=”utf-8”?>
<rotate xmlns:android=”http://schemas.android.com/apk/res/android”
        android:drawable=”@drawable/loading”
        android:fromDegrees=”0”
        android:pivotX=”50%”
        android:pivotY=”50%”
        android:toDegrees=”360”>
</rotate>
```

drawable/loading.png

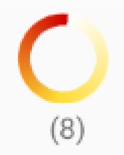


drawable\ring.xml

```
<?xml version=”1.0” encoding=”utf-8”?>
<shape  xmlns:android=”http://schemas.android.com/apk/res/android”
    android:innerRadiusRatio=”3”
    android:shape=”ring”
    android:thicknessRatio=”8”
    android:useLevel=”false”>
    <gradient
        android:centerColor=”#FF333333”
        android:centerX=”0.50”
        android:centerY=”0.50”
        android:endColor=”#FF333333”
        android:startColor=”#FFAAAAAA”
        android:type=”sweep”
        android:useLevel=”false” />
</shape>
```

drawable\rotate.xml

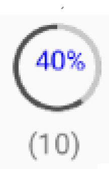
```
<?xml version="1.0" encoding="utf-8"?>
<rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromDegrees="0"
    android:pivotX="50%"
    android:pivotY="50%"
    android:toDegrees="360">
    <shape
        android:innerRadiusRatio="3"
        android:shape="ring"
        android:thicknessRatio="8"
        android:useLevel="false">
        <gradient
            android:centerColor="#FFDC35"
            android:centerY="0.50"
            android:endColor="#CE0000"
            android:startColor="#FFFFFF"
            android:type="sweep"
            android:useLevel="false" />
    </shape>
</rotate>
```

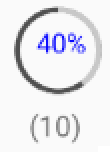


android:innerRadiusRatio 内环半径相对于环的宽度的比例
android:thicknessRatio　环的宽度相对于环的厚度的比例

CirclePgBar.java

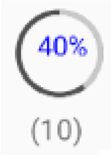
```
import android.content.Context; /** Created by yang_zzheng on 2016/7/12
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.graphics.RectF;
import android.util.AttributeSet;
import android.view.View;
public class CirclePgBar extends View {
    private Paint mBackPaint;    private Paint mFrontPaint;
    private Paint mTextPaint;    private float mStrokeWidth = 8;
    private float mHalfStrokeWidth = mStrokeWidth / 2;
    private float mRadius = 60;
    private RectF mRect;
    private int mProgress = 40;
    private int mTargetProgress = 40;
    private int mMax = 100;
    private int mWidth;  private int mHeight;
    public CirclePgBar(Context context) {
        super(context);
        init();
    }
    public CirclePgBar(Context context, AttributeSet attrs) {
        super(context, attrs);
        init();
    }
```

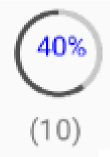




```
public CirclePgBar(Context context, AttributeSet attrs, int defStyleAttr) {
    super(context, attrs, defStyleAttr);
    init();
}
public void setProgress(int progress) {
    mProgress=progress;
    invalidate();
}
public int getProgress() {
    return mProgress;
}
public void setMax(int max) { mMax=max; }
public int getMax() { return mMax;  }
@Override
protected void onDraw(Canvas canvas) {
    initRect();
    float angle = mProgress / (float) mMax * 360;
    canvas.drawCircle(mWidth / 2, mHeight / 2, mRadius, mBackPaint);
    canvas.drawArc(mRect, -90, angle, false, mFrontPaint);
    canvas.drawText(mProgress + "%", mWidth / 2 + mHalfStrokeWidth,
                    mHeight / 2 + mHalfStrokeWidth, mTextPaint);
    /* if(mProgress < mTargetProgress){ mProgress += 1; invalidate(); }*/
}
```

```
@Override
protected void onMeasure(int widthMeasureSpec, int heightMeasureSpe
    super.onMeasure(widthMeasureSpec, heightMeasureSpec);
    mWidth = getRealSize(widthMeasureSpec);
    mHeight = getRealSize(heightMeasureSpec);
    setMeasuredDimension(mWidth, mHeight);
}
private void init() {
    mBackPaint = new Paint();
    mBackPaint.setColor(Color.rgb(80,80,80));
    mBackPaint.setAntiAlias(true);
    mBackPaint.setStyle(Paint.Style.STROKE);
    mBackPaint.setStrokeWidth(mStrokeWidth);
    mFrontPaint = new Paint();
    mFrontPaint.setColor(Color.rgb(200,200,200));
    mFrontPaint.setAntiAlias(true);
    mFrontPaint.setStyle(Paint.Style.STROKE);
    mFrontPaint.setStrokeWidth(mStrokeWidth);
    mTextPaint = new Paint();
    mTextPaint.setColor(Color.BLUE);
    mTextPaint.setAntiAlias(true);
    mTextPaint.setTextSize(40);
    mTextPaint.setTextAlign(Paint.Align.CENTER);
}
```

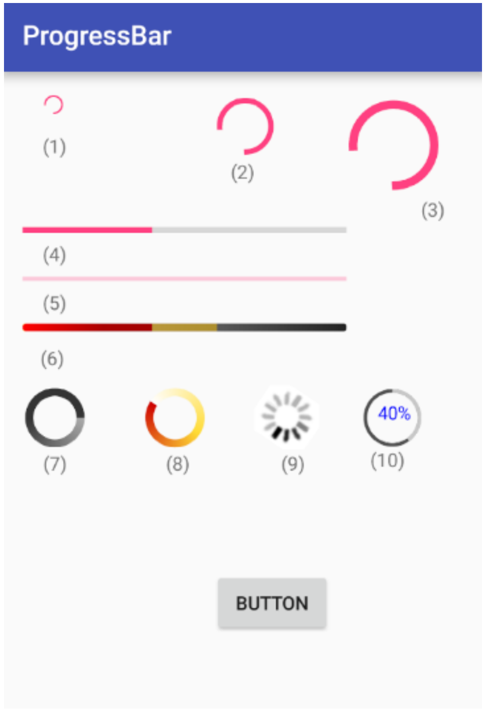




```
    public int getRealSize(int measureSpec) {
        int result = 1;
        int mode = MeasureSpec.getMode(measureSpec);
        int size = MeasureSpec.getSize(measureSpec);
        if (mode == MeasureSpec.AT_MOST || mode == MeasureSpec.UNSPECIFIED) {
            result = (int) (mRadius * 2 + mStrokeWidth);
        } else {
            result = size;
        }
        return result;
    }
    private void initRect() {
        if (mRect == null) {
            mRect = new RectF();
            int viewSize = (int) (mRadius * 2);
            int left = (mWidth - viewSize) / 2;
            int top = (mHeight - viewSize) / 2;
            int right = left + viewSize;
            int bottom = top + viewSize;
            mRect.set(left, top, right, bottom);
        }
    }
}
```

MainActivity.java

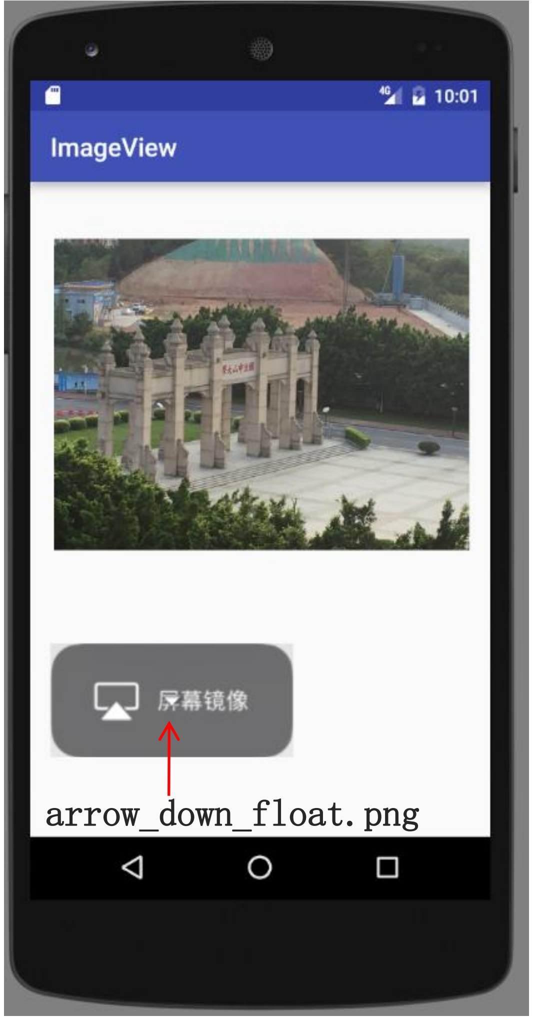
```java
public class MainActivity extends AppCompatActivity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button btn=(Button)findViewById(R.id.button);
        btn.setOnClickListener(new View.OnClickListener(){
            public void onClick(View v){
                ProgressBar pb1= (ProgressBar)findViewById(R.id.progressBar4);
                ProgressBar pb2= (ProgressBar)findViewById(R.id.progressBar6);
                CirclePgBar pb3= (CirclePgBar)findViewById(R.id.progressBar10);
                if(pb1.getProgress()>=pb1.getMax())
                    pb1.setProgress(0);
                else
                    pb1.setProgress(pb1.getProgress()+10);
                if(pb2.getProgress()>=pb2.getMax())
                    pb2.setProgress(0);
                else
                    pb2.setProgress(pb2.getProgress()+10);
                if(pb3.getProgress()>=pb3.getMax())
                    pb3.setProgress(0);
                else
                    pb3.setProgress(pb3.getProgress()+10);
            }
        });
    }
}
```



图像框(ImageView)

参考

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android
    =http://schemas.android.com/apk/res/android
    android:id="@+id/activity_main">
    <ImageView
        app:srcCompat="@drawable/sysu"
        android:id="@+id/imageView"
        android:layout_width="wrap_content"
        android:scaleType="centerInside"
        android:layout_height="300dp" />
    <ImageButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src ="@android:drawable/arrow_down_float"
        android:background ="@drawable/btn"
        android:layout_below="@+id/imageView"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="45dp"
        android:id="@+id/imageButton" />
</RelativeLayout>
```



* scaleType见附录

android:scaleType是控制图片如何伸缩和摆放在ImageView中：

center	按图片的原来size居中显示，当图片长/宽超过View的长/宽，则截取图片的居中部分显示。
centerCrop	按比例扩大图片的size居中显示，使得图片长(宽)等于或大于View的长(宽)。
centerInside	将图片的内容完整居中显示，通过按比例缩小或原来的size使得图片长/宽等于或小于View的长/宽。
fitCenter	把图片按比例扩大/缩小到View的宽度，居中显示。
fitEnd	把图片按比例扩大/缩小到View的宽度，显示在View的下部分位置。
fitStart	把图片按比例扩大/缩小到View的宽度，显示在View的上部分位置。
fitXY	把图片 不按比例扩大/缩小到View的大小显示。
matrix	用矩阵来绘制，动态缩小放大图片来显示。

```
imgView.setImageBitmap(dstBitmap);
```

编辑框(EditText)

参考

- EditText用于输入文本，可以设置输入类型（ inputType ）包括number、date、phone、textUri、textEmailAddress、textPassword、textMultiLine，使获得焦点时自动**显示合适的键盘**。
- textMultiLine 用于多行文本输入。输入类型还有textAutoCorrect可以自动校正错误，textAutoComplete可以自动选择输入。

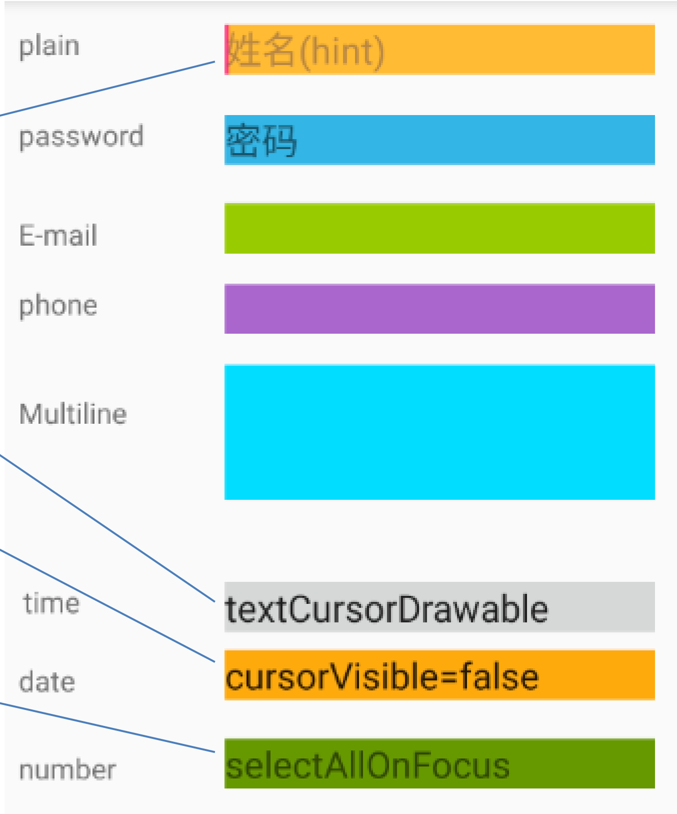
当文本框为空时属性hint显示输入提示，用textColorHint设置提示文字的颜色

可以用属性textCursorDrawable指定一幅图作为光标

属性cursorVisible为false时聚焦控件没有光标

如果属性selectAllOnFocus为true，聚焦控件时会选择所有已经输入内容

enabled为false时，不能获得焦点和输入



plain 姓名(hint)

password 密码

E-mail

phone

Multiline

time textCursorDrawable

date cursorVisible=false

number selectAllOnFocus

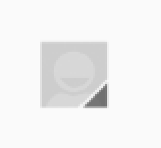
联系人控件 (QuickContactBadge)

参考

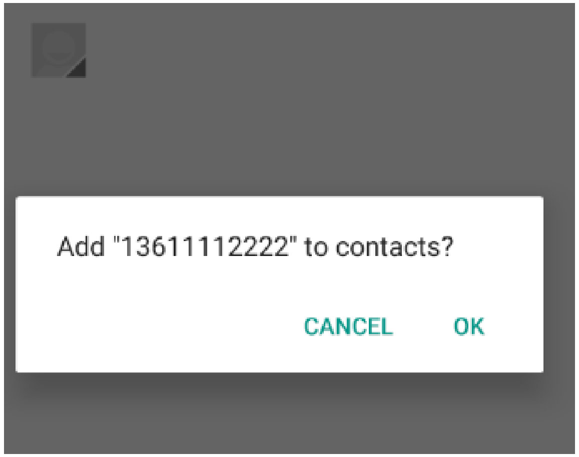
QuickContactBadge可以关联到手机中指定联系人，当用户单击它时，系统将打开相应的联系人的联系方式界面，没有该联系人时提示增加联系人。

```
<QuickContactBadge
    android:id="@+id/quickContactBadge"
    android:background="#ffffff"
    style="?android:attr/quickContactBadgeStyleWindowSmall"
/>
```

```
QuickContactBadge quickContactBadge;
quickContactBadge = (QuickContactBadge) findViewById(R.id.quickContactBadge);
quickContactBadge.assignContactFromPhone("13611112222", true);
quickContactBadge.setMode(ContactsContract.QuickContact.MODE_SMALL);
```



点击后

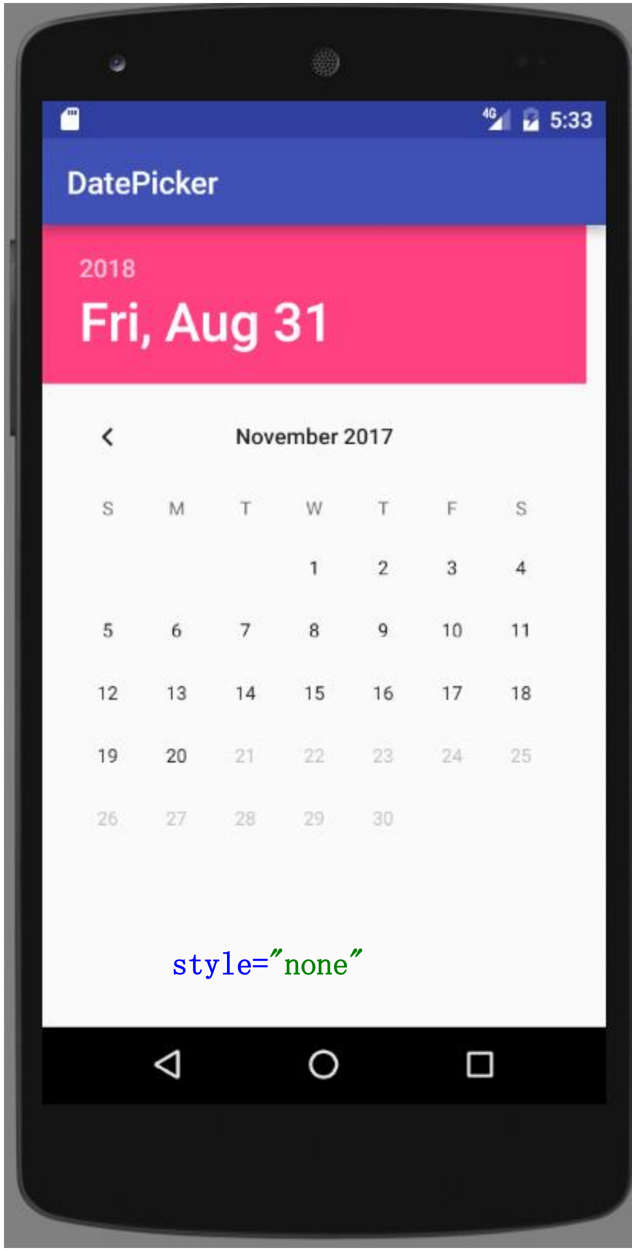


Add "13611112222" to contacts?

日期选择器 (DatePicker)

参考

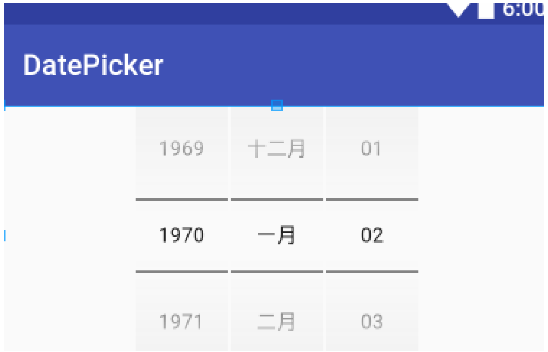
```
public class MainActivity extends AppCompatActivity {
    private DatePicker datePicker;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        datePicker
          =(DatePicker)findViewById(R.id.datePicker);

        Calendar cal = Calendar.getInstance();
        cal.set(Calendar.YEAR, 2016);
        cal.set(Calendar.MONTH, 10);
        cal.set(Calendar.DAY_OF_MONTH, 20);

        datePicker.setMinDate(cal.getTimeInMillis());
        cal.add(Calendar.MONTH, 12);
        datePicker.setMaxDate(cal.getTimeInMillis());
```



style="none"

```
 datePicker.init(2017, 20, 0, new DatePicker.OnDateChangedListener() {
            @Override
            public void onDateChanged(DatePicker view, int year,
                                      int monthOfYear, int dayOfMonth) {
                Calendar calendar = Calendar.getInstance();
                calendar.set(year, monthOfYear, dayOfMonth);
                SimpleDateFormat format = new SimpleDateFormat(
                        "yyyy年MM月dd日  HH:mm");
                Toast.makeText(MainActivity.this,
                        format.format(calendar.getTime()), Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

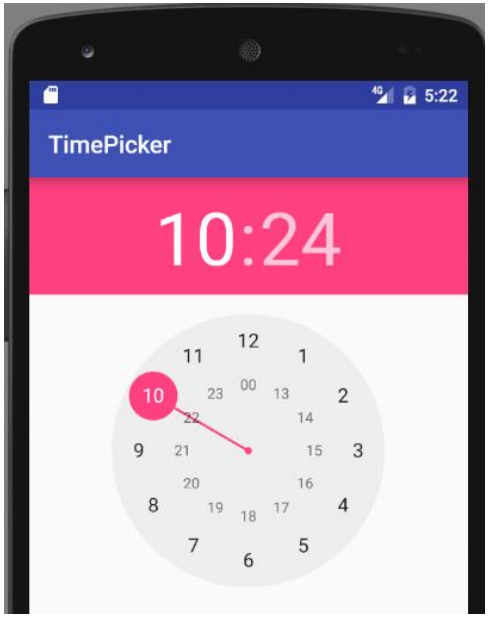
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <DatePicker
        android:id="@+id/datePicker"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" />
</LinearLayout>
```



```
style="@android:style/Widget.DatePicker"
```

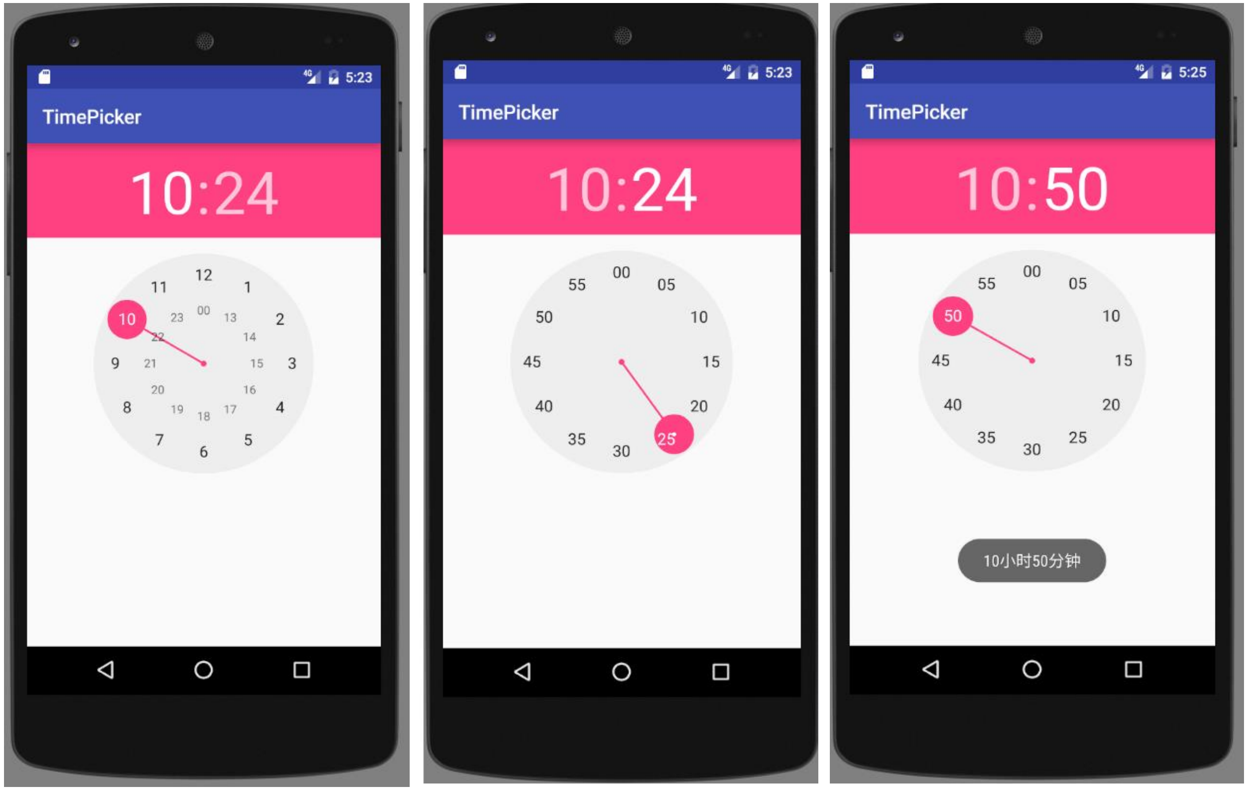
时间选择器 (TimePicker)

参考



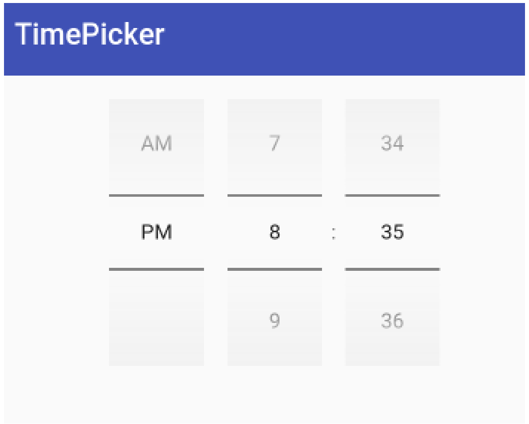
```
public class MainActivity extends AppCompatActivity {
    private TimePicker timePicker;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        timePicker = (TimePicker) findViewById(R.id.timePicker);
        timePicker.setIs24HourView(true);
        timePicker.setCurrentHour(10);  // setHour(10) -- 新版
        timePicker.setCurrentMinute(24);// setMinute(10) -- 新版
        timePicker.setOnTimeChangedListener(new TimePicker.OnTimeChangedListener() {
            @Override
            public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {
                Toast.makeText(MainActivity.this,
                        hourOfDay + "小时" + minute + "分钟",
                        Toast.LENGTH_SHORT).show();
            }
        });
    }
}
```

```
<TimePicker
    android:id="@+id/timePicker"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```



10小时50分钟

style="none"



style=
"@android:style/Widget.Holo.TimePicker"

附录

附录1、安卓项目版本修改

附录2、系统命名颜色

附录3、系统主题Theme列表

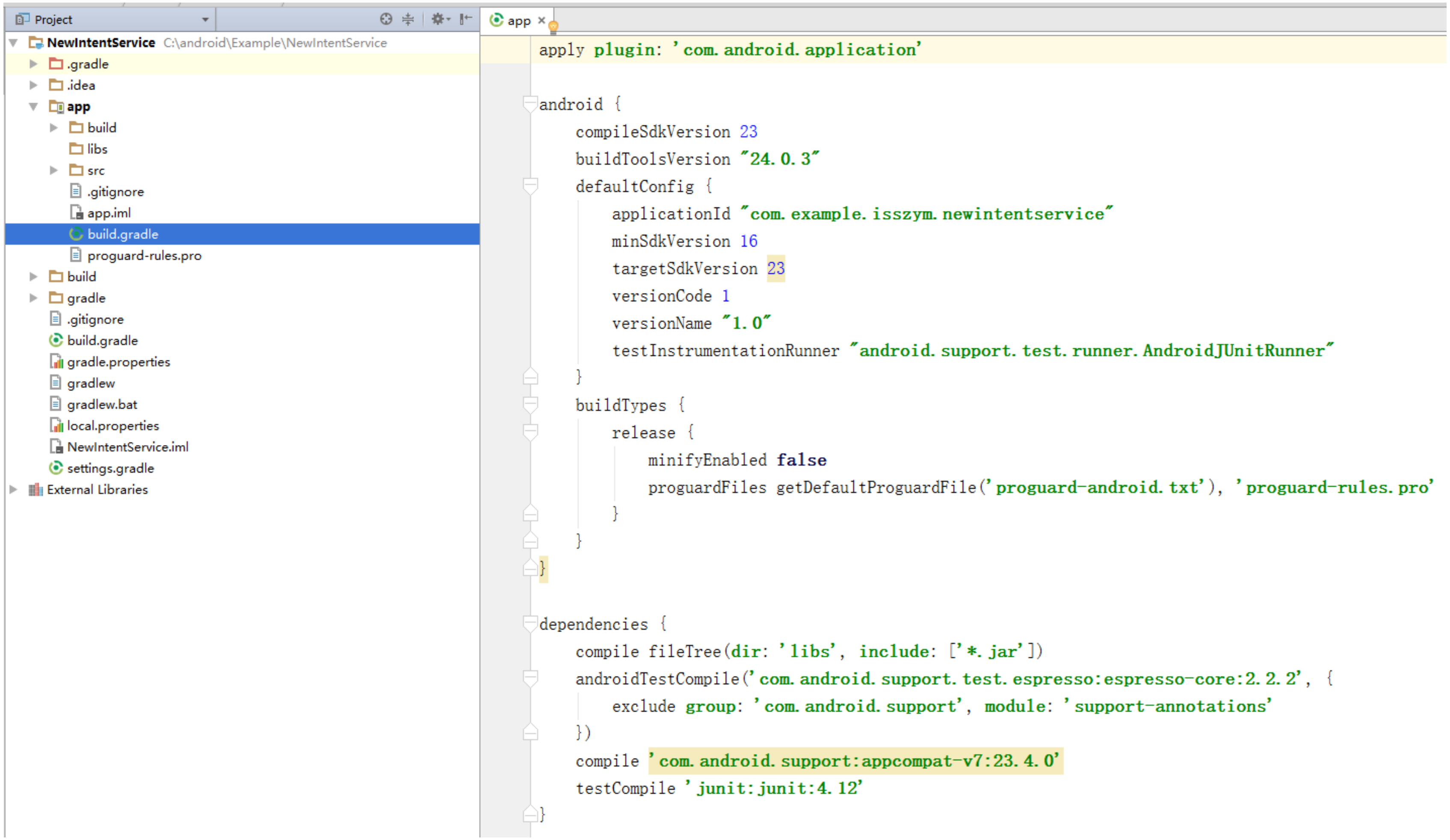
附录4、控件大全

附录5、View的变换

附录6、ToggleButton类

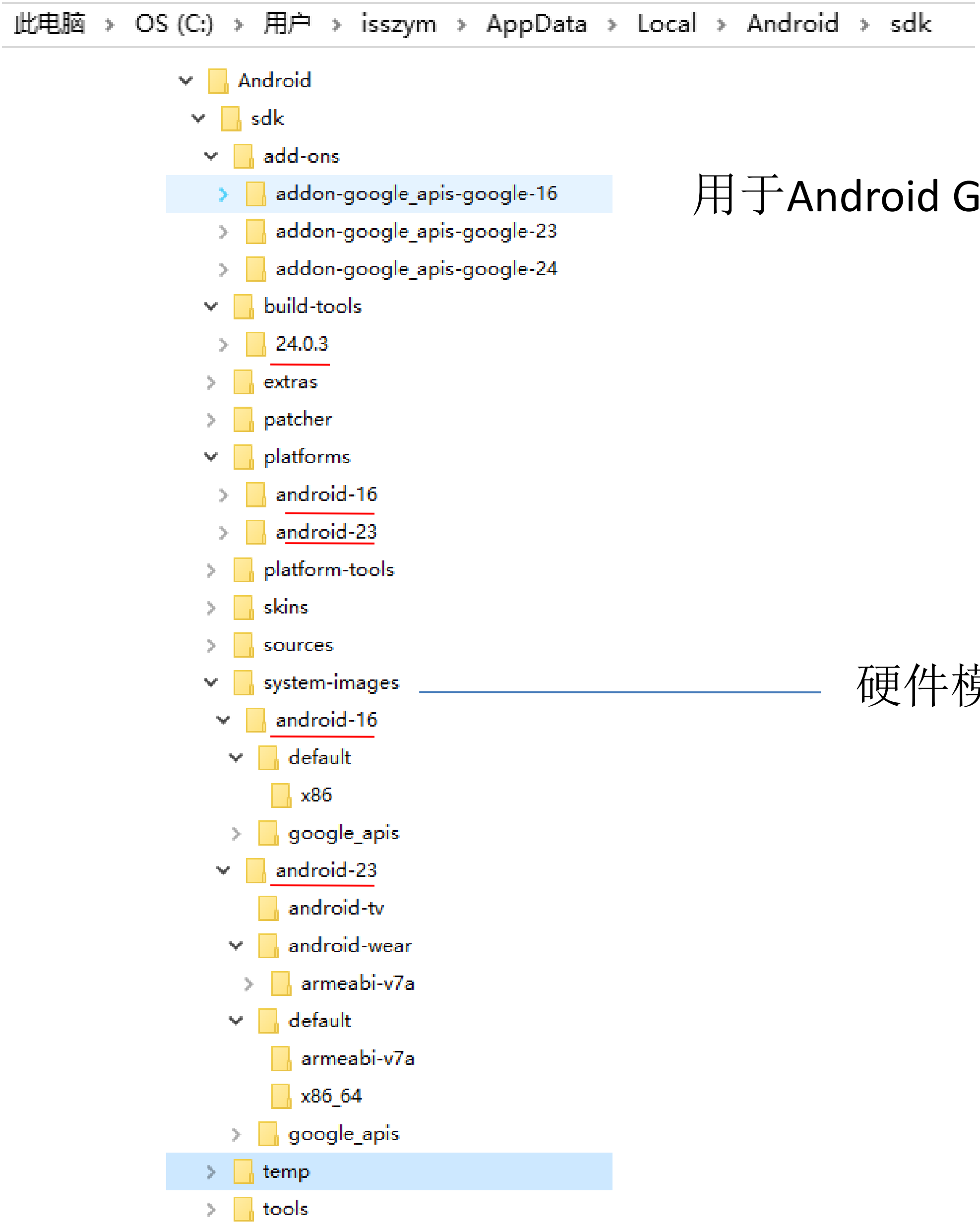
附录7、课件所学的控件

附录1、项目版本修改



```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 23
    buildToolsVersion "24.0.3"
    defaultConfig {
        applicationId "com.example.isszym.newintentservice"
        minSdkVersion 16
        targetSdkVersion 23
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
        exclude group: 'com.android.support', module: 'support-annotations'
    })
    compile 'com.android.support:appcompat-v7:23.4.0'
    testCompile 'junit:junit:4.12'
}
```

- buildTools和compile都可以用最高版本，因为它们也支持低版本编译。
- version 16-安卓4.1 version 23-安卓6.0
- 可以通过新建一个空白app查看该配置文件的当前参数



此电脑 › OS (C:) › 用户 › isszym › AppData › Local › Android › sdk

```
Android
  sdk
    add-ons
      addon-google_apis-google-16
      addon-google_apis-google-23
      addon-google_apis-google-24
    build-tools
      24.0.3
    extras
    patcher
    platforms
      android-16
      android-23
    platform-tools
    skins
    sources
    system-images
      android-16
        default
          x86
        google_apis
      android-23
        android-tv
        android-wear
          armeabi-v7a
        default
          armeabi-v7a
          x86_64
        google_apis
    temp
    tools
```

用于Android Game

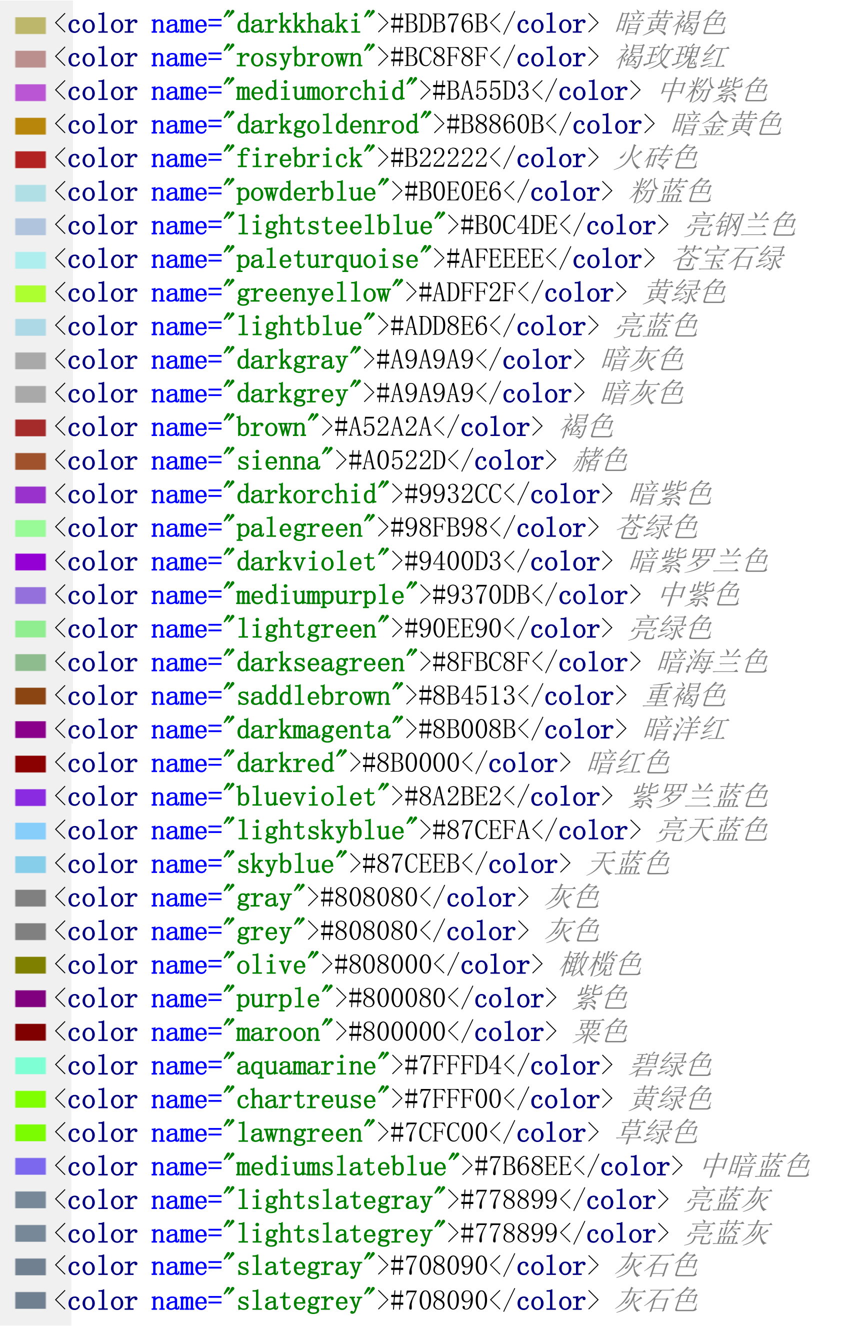
硬件模拟

附录2、系统命名颜色

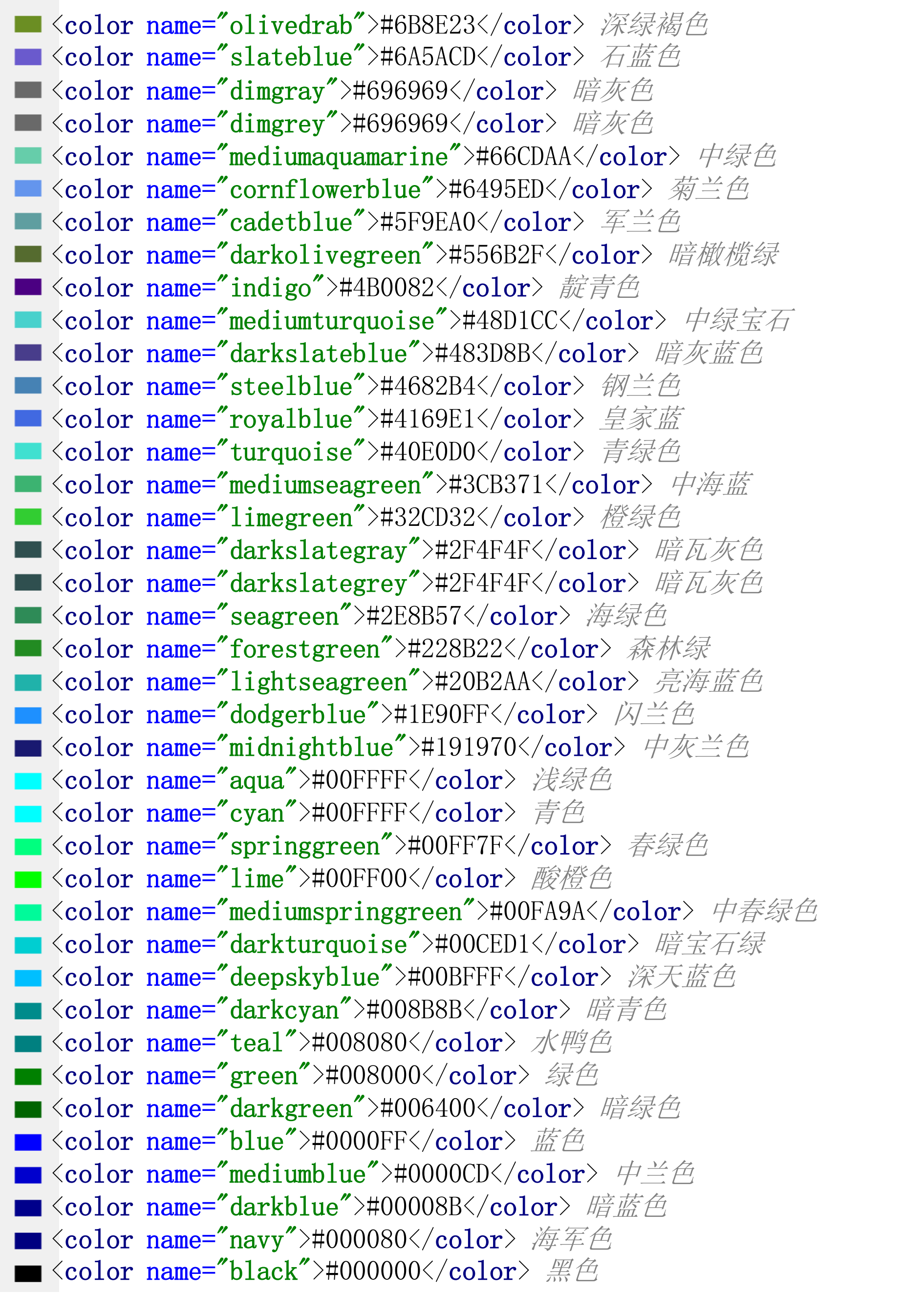
参考

```
<color name=”white”>#FFFFFF</color> 白色
<color name=”ivory”>#FFFFF0</color> 象牙色
<color name=”lightyellow”>#FFFFE0</color> 亮黄色
<color name=”yellow”>#FFFF00</color> 黄色
<color name=”snow”>#FFFAFA</color> 雪白色
<color name=”floralwhite”>#FFFAF0</color> 花白色
<color name=”lemonchiffon”>#FFFACD</color> 柠檬绸色
<color name=”cornsilk”>#FFF8DC</color> 米绸色
<color name=”seashell”>#FFF5EE</color> 海贝色
<color name=”lavenderblush”>#FFF0F5</color> 淡紫红
<color name=”papayawhip”>#FFEFD5</color> 番木色
<color name=”blanchedalmond”>#FFEBCD</color> 白杏色
<color name=”mistyrose”>#FFE4E1</color> 浅玫瑰色
<color name=”bisque”>#FFE4C4</color> 桔黄色
<color name=”moccasin”>#FFE4B5</color> 鹿皮色
<color name=”navajowhite”>#FFDEAD</color> 纳瓦白
<color name=”peachpuff”>#FFDAB9</color> 桃色
<color name=”gold”>#FFD700</color> 金色
<color name=”pink”>#FFC0CB</color> 粉红色
<color name=”lightpink”>#FFB6C1</color> 亮粉红色
<color name=”orange”>#FFA500</color> 橙色
<color name=”lightsalmon”>#FFA07A</color> 亮肉色
<color name=”darkorange”>#FF8C00</color> 暗桔黄色
<color name=”coral”>#FF7F50</color> 珊瑚色
<color name=”hotpink”>#FF69B4</color> 热粉红色
<color name=”tomato”>#FF6347</color> 西红柿色
<color name=”orangered”>#FF4500</color> 红橙色
<color name=”deeppink”>#FF1493</color> 深粉红色
<color name=”fuchsia”>#FF00FF</color> 紫红色
<color name=”magenta”>#FF00FF</color> 红紫色
<color name=”red”>#FF0000</color> 红色
<color name=”oldlace”>#FDF5E6</color> 老花色
<color name=”lightgoldenrodyellow”>#FAFAD2</color>亮金黄色
<color name=”linen”>#FAF0E6</color> 亚麻色
```

```
<color name=”antiquewhite”>#FAEBD7</color> 古董白
<color name=”salmon”>#FA8072</color> 鲜肉色
<color name=”ghostwhite”>#F8F8FF</color> 幽灵白
<color name=”mintcream”>#F5FFFA</color> 薄荷色
<color name=”whitesmoke”>#F5F5F5</color> 烟白色
<color name=”beige”>#F5F5DC</color> 米色
<color name=”wheat”>#F5DEB3</color> 浅黄色
<color name=”sandybrown”>#F4A460</color> 沙褐色
<color name=”azure”>#F0FFFF</color> 天蓝色
<color name=”honeydew”>#F0FFF0</color> 蜜色
<color name=”aliceblue”>#F0F8FF</color> 艾利斯兰
<color name=”khaki”>#F0E68C</color> 黄褐色
<color name=”lightcoral”>#F08080</color> 亮珊瑚色
<color name=”palegoldenrod”>#EEE8AA</color> 苍麒麟色
<color name=”violet”>#EE82EE</color> 紫罗兰色
<color name=”darksalmon”>#E9967A</color> 暗肉色
<color name=”lavender”>#E6E6FA</color> 淡紫色
<color name=”lightcyan”>#E0FFFF</color> 亮青色
<color name=”burlywood”>#DEB887</color> 实木色
<color name=”plum”>#DDA0DD</color> 洋李色
<color name=”gainsboro”>#DCDCDC</color> 淡灰色
<color name=”crimson”>#DC143C</color> 暗深红色
<color name=”palevioletred”>#DB7093</color> 苍紫罗兰色
<color name=”goldenrod”>#DAA520</color> 金麒麟色
<color name=”orchid”>#DA70D6</color> 淡紫色
<color name=”thistle”>#D8BFD8</color> 蓟色
<color name=”lightgray”>#D3D3D3</color> 亮灰色
<color name=”lightgrey”>#D3D3D3</color> 亮灰色
<color name=”tan”>#D2B48C</color> 茶色
<color name=”chocolate”>#D2691E</color> 巧可力色
<color name=”peru”>#CD853F</color> 秘鲁色
<color name=”indianred”>#CD5C5C</color> 印第安红
<color name=”mediumvioletred”>#C71585</color> 中紫罗兰色
<color name=”silver”>#C0C0C0</color> 银色
```



```
<color name="darkkhaki">#BDB76B</color> 暗黄褐色
<color name="rosybrown">#BC8F8F</color> 褐玫瑰红
<color name="mediumorchid">#BA55D3</color> 中粉紫色
<color name="darkgoldenrod">#B8860B</color> 暗金黄色
<color name="firebrick">#B22222</color> 火砖色
<color name="powderblue">#B0E0E6</color> 粉蓝色
<color name="lightsteelblue">#B0C4DE</color> 亮钢兰色
<color name="paleturquoise">#AFEEEE</color> 苍宝石绿
<color name="greenyellow">#ADFF2F</color> 黄绿色
<color name="lightblue">#ADD8E6</color> 亮蓝色
<color name="darkgray">#A9A9A9</color> 暗灰色
<color name="darkgrey">#A9A9A9</color> 暗灰色
<color name="brown">#A52A2A</color> 褐色
<color name="sienna">#A0522D</color> 赭色
<color name="darkorchid">#9932CC</color> 暗紫色
<color name="palegreen">#98FB98</color> 苍绿色
<color name="darkviolet">#9400D3</color> 暗紫罗兰色
<color name="mediumpurple">#9370DB</color> 中紫色
<color name="lightgreen">#90EE90</color> 亮绿色
<color name="darkseagreen">#8FBC8F</color> 暗海兰色
<color name="saddlebrown">#8B4513</color> 重褐色
<color name="darkmagenta">#8B008B</color> 暗洋红
<color name="darkred">#8B0000</color> 暗红色
<color name="blueviolet">#8A2BE2</color> 紫罗兰蓝色
<color name="lightskyblue">#87CEFA</color> 亮天蓝色
<color name="skyblue">#87CEEB</color> 天蓝色
<color name="gray">#808080</color> 灰色
<color name="grey">#808080</color> 灰色
<color name="olive">#808000</color> 橄榄色
<color name="purple">#800080</color> 紫色
<color name="maroon">#800000</color> 粟色
<color name="aquamarine">#7FFFD4</color> 碧绿色
<color name="chartreuse">#7FFF00</color> 黄绿色
<color name="lawngreen">#7CFC00</color> 草绿色
<color name="mediumslateblue">#7B68EE</color> 中暗蓝色
<color name="lightslategray">#778899</color> 亮蓝灰
<color name="lightslategrey">#778899</color> 亮蓝灰
<color name="slategray">#708090</color> 灰石色
<color name="slategrey">#708090</color> 灰石色
```



```
<color name="olivedrab">#6B8E23</color> 深绿褐色
<color name="slateblue">#6A5ACD</color> 石蓝色
<color name="dimgray">#696969</color> 暗灰色
<color name="dimgrey">#696969</color> 暗灰色
<color name="mediumaquamarine">#66CDAA</color> 中绿色
<color name="cornflowerblue">#6495ED</color> 菊兰色
<color name="cadetblue">#5F9EA0</color> 军兰色
<color name="darkolivegreen">#556B2F</color> 暗橄榄绿
<color name="indigo">#4B0082</color> 靛青色
<color name="mediumturquoise">#48D1CC</color> 中绿宝石
<color name="darkslateblue">#483D8B</color> 暗灰蓝色
<color name="steelblue">#4682B4</color> 钢兰色
<color name="royalblue">#4169E1</color> 皇家蓝
<color name="turquoise">#40E0D0</color> 青绿色
<color name="mediumseagreen">#3CB371</color> 中海蓝
<color name="limegreen">#32CD32</color> 橙绿色
<color name="darkslategray">#2F4F4F</color> 暗瓦灰色
<color name="darkslategrey">#2F4F4F</color> 暗瓦灰色
<color name="seagreen">#2E8B57</color> 海绿色
<color name="forestgreen">#228B22</color> 森林绿
<color name="lightseagreen">#20B2AA</color> 亮海蓝色
<color name="dodgerblue">#1E90FF</color> 闪兰色
<color name="midnightblue">#191970</color> 中灰兰色
<color name="aqua">#00FFFF</color> 浅绿色
<color name="cyan">#00FFFF</color> 青色
<color name="springgreen">#00FF7F</color> 春绿色
<color name="lime">#00FF00</color> 酸橙色
<color name="mediumspringgreen">#00FA9A</color> 中春绿色
<color name="darkturquoise">#00CED1</color> 暗宝石绿
<color name="deepskyblue">#00BFFF</color> 深天蓝色
<color name="darkcyan">#008B8B</color> 暗青色
<color name="teal">#008080</color> 水鸭色
<color name="green">#008000</color> 绿色
<color name="darkgreen">#006400</color> 暗绿色
<color name="blue">#0000FF</color> 蓝色
<color name="mediumblue">#0000CD</color> 中兰色
<color name="darkblue">#00008B</color> 暗蓝色
<color name="navy">#000080</color> 海军色
<color name="black">#000000</color> 黑色
```

附录3、系统主题Theme列表

系统默认的主题有三种：Theme,Theme.Holo,Theme.DeviceDefault， 但是实际上在此基础系统还定义了大量的派生主题，最典型的是对应的Light主题。

API 1:	
android:Theme	根主题
android:Theme.Black	背景黑色
android:Theme.Light	背景白色
android:Theme.Wallpaper	以桌面墙纸为背景
android:Theme.Translucent	透明背景
android:Theme.Panel	平板风格
android:Theme.Dialog	对话框风格
API 11:	
android:Theme.Holo Holo	根主题
android:Theme.Holo.Black Holo	黑主题
android:Theme.Holo.Light Holo	白主题
API 14:	
Theme.DeviceDefault	设备默认根主题
Theme.DeviceDefault.Black	设备默认黑主题
Theme.DeviceDefault.Light	设备默认白主题
API 21: (Android Material Design主题)	
Theme.Material Material	根主题
Theme.Material.Light Material	白主题
兼容包v7中带的主题：	
Theme.AppCompat	兼容主题的根主题
Theme.AppCompat.Black	兼容主题的黑色主题
Theme.AppCompat.Light	兼容主题的白色主题
Theme.AppCompat.Light.DarkActionBar	兼容主题的白色主题(暗色ActionBar)

参考-3725466.htm

一个完整的主题应该定义哪些内容呢，以Theme为例，如下：

1）颜色

```
<item name="colorForeground">@android:color/bright_foreground_dark</item>
<item name="colorForegroundInverse">@android:color/bright_foreground_dark_inverse</item>
...
<item name="colorFocusedHighlight">@color/legacy_selected_highlight</item>
<item name="colorMultiSelectHighlight">@color/legacy_selected_highlight</item>
<item name="colorActivatedHighlight">@color/legacy_selected_highlight</item>
```

2）字体

```
<!-- Text styles -->
<item name="textAppearance">@android:style/TextAppearance</item>
<item name="textColorPrimary">@android:color/primary_text_dark</item>
<item name="textColorSecondary">@android:color/secondary_text_dark</item>
...
<item name="textAppearanceLargePopupMenu">@android:style/TextAppearance.Widget.PopupMenu.Large</item>
<item name="textAppearanceSmallPopupMenu">@android:style/TextAppearance.Widget.PopupMenu.Small</item>
```

3）按钮

```
<!-- Button styles -->
<item name="buttonStyle">@android:style/Widget.Button</item>
...
<item name="selectableItemBackground">@android:drawable/item_background</item>
<item name="borderlessButtonStyle">?android:attr/buttonStyle</item>
<item name="homeAsUpIndicator">@android:drawable/ic_ab_back_holo_dark</item>
```

4）List

```
<!-- List attributes -->
<item name="listPreferredItemHeight">64dip</item>
<item name="listPreferredItemHeightSmall">?android:attr/listPreferredItemHeight</item>
...
<item name="listPreferredItemPaddingRight">6dip</item>
<item name="listPreferredItemPaddingStart">6dip</item>
<item name="listPreferredItemPaddingEnd">6dip</item>
```

5）Window

```
<!-- Window attributes -->
<item name="windowBackground">@android:drawable/screen_background_selector_dark</item>
<item name="windowFrame">@null</item>
<item name="windowNoTitle">false</item>
...
<item name="windowCloseOnTouchOutside">false</item>
<item name="windowTranslucentStatus">false</item>
<item name="windowTranslucentNavigation">false</item>
```

6）Dialog

```
<!-- Dialog attributes -->
<item name="dialogTheme">@android:style/Theme.Dialog</item>
<item name="dialogTitleIconsDecorLayout">@layout/dialog_title_icons</item>
<item name="dialogCustomTitleDecorLayout">@layout/dialog_custom_title</item>
<item name="dialogTitleDecorLayout">@layout/dialog_title</item>
```

7）AlertDialog

```
<!-- AlertDialog attributes -->
<item name="alertDialogTheme">@android:style/Theme.Dialog.Alert</item>
<item name="alertDialogStyle">@android:style/AlertDialog</item>
<item name="alertDialogCenterButtons">true</item>
<item name="alertDialogIcon">@android:drawable/ic_dialog_alert</item>
```

8）Panel

```
<!-- Panel attributes -->
<item name="panelBackground">@android:drawable/menu_background</item>
<item name="panelFullBackground">@android:drawable/menu_background_fill_parent_width</item>
<!-- These three attributes do not seems to be used by the framework. Declared public though -->
<item name="panelColorBackground">#000</item>
<item name="panelColorForeground">?android:attr/textColorPrimary</item>
<item name="panelTextAppearance">?android:attr/textAppearance</item>

<item name="panelMenuIsCompact">false</item>
<item name="panelMenuListWidth">296dip</item>
```

9）滚动条（Scrollbar）

```
<!-- Scrollbar attributes -->
<item name="scrollbarFadeDuration">250</item>
<item name="scrollbarDefaultDelayBeforeFade">300</item>
<item name="scrollbarSize">10dip</item>
...
<item name="scrollbarTrackVertical">@null</item>
```

10）文字选中（Text selection）

```
<!-- Text selection handle attributes -->
<item name="textSelectHandleLeft">@android:drawable/text_select_handle_left</item>
<item name="textSelectHandleRight">@android:drawable/text_select_handle_right</item>
...
<item name="textEditSuggestionItemLayout">@android:layout/text_edit_suggestion_item</item>
<item name="textCursorDrawable">@null</item>
```

AndroidManifest.xml

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    </activity>
</application>
```

values/styles.xml

```
<resources>
    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
    <!-- Customize your theme here. -->
    <item name="colorPrimary">@color/colorPrimary</item>
    <item name="colorPrimaryDark">@color/colorPrimaryDark</item>
    <item name="colorAccent">@color/colorAccent</item>
    </style>
</resources>
```

values/colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
</resources>
```

附录4、控件大全 参考

AbsListView
AbsListView.LayoutParams
AbsoluteLayout
AbsoluteLayout.LayoutParams
AbsSeekBar
AbsSpinner
ActionMenuView
ActionMenuView.LayoutParams
AdapterView
AdapterView.AdapterContextMenuInfo
AdapterViewAnimator
AdapterViewFlipper
AlphabetIndexer
AnalogClock
ArrayAdapter
AutoCompleteTextView
BaseAdapter
BaseExpandableListAdapter
Button
CalendarView
CheckBox
CheckedTextView
Chronometer
CompoundButton
CursorAdapter
CursorTreeAdapter
DatePicker
DialerFilter
DigitalClock
EdgeEffect
EditText
ExpandableListView
ExpandableListView.ExpandableListContextMenuInfo
Filter
Filter.FilterResults
FrameLayout
FrameLayout.LayoutParams
Gallery
Gallery.LayoutParams
GridLayout
GridLayout.Alignment
GridLayout.LayoutParams
GridLayout.Spec
GridView
HeaderViewListAdapter
HorizontalScrollView
ImageButton
ImageSwitcher
ImageView
LinearLayout
LinearLayout.LayoutParams
ListPopupWindow
ListView
ListView.FixedViewInfo
MediaController
MultiAutoCompleteTextView
MultiAutoCompleteTextView.CommaTokenizer
NumberPicker
OverScroller
PopupMenu
PopupWindow
ProgressBar
QuickContactBadge
RadioButton
RadioGroup
RadioGroup.LayoutParams
RatingBar
RelativeLayout
RelativeLayout.LayoutParams
RemoteViews
RemoteViewsService
ResourceCursorAdapter
ResourceCursorTreeAdapter
Scroller
ScrollView
SearchView
SeekBar
ShareActionProvider
SimpleAdapter
SimpleCursorAdapter
SimpleCursorTreeAdapter
SimpleExpandableListAdapter
SlidingDrawer
Space
Spinner
StackView
Switch
TabHost
TabHost.TabSpec
TableLayout
TableLayout.LayoutParams
TableRow
TableRow.LayoutParams
TabWidget
TextClock
TextSwitcher
TextView
TextView.SavedState
TimePicker
Toast
ToggleButton
Toolbar
Toolbar.LayoutParams
TwoLineListItem
VideoView
ViewAnimator
ViewFlipper
ViewSwitcher
ZoomButton
ZoomButtonsController
ZoomControls

附录5、View的变换

参考

View是所有控件的基类。View除了可以平移、绕轴心(pivot)旋转和缩放，还可以绕X轴和Y轴旋转。

```
public void setPivotX(float pivotX)
public void setPivotY(float pivotY)
public void setTranslationX(float translationX)      // 相对于自己在X方向平移
public void setTranslationY(float translationY)
public void setTranslationZ(float translationZ)
public void setScaleX(float scaleX)                   // 绕PivotX
public void setScaleY(float scaleY)                   // 绕PivotY
public void setRotate(float rotation)                 // 绕PivotX和PivotY
public void setRotateX(float rotationX)               // 绕X轴
public void setRotateY(float rotationY)               // 绕Y轴
```

附录6、ToggleButton类

ToggleButton类是系统类，这个是该类的源码

```java
public class ToggleButton extends CompoundButton {
    private CharSequence mTextOn;
    private CharSequence mTextOff;
    private Drawable mIndicatorDrawable;
    private static final int NO_ALPHA = 0xFF;
    private float mDisabledAlpha;
    public ToggleButton(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);
        TypedArray a =
                context.obtainStyledAttributes(
                        attrs, com.android.internal.R.styleable.ToggleButton, defStyle, 0);
        mTextOn = a.getText(com.android.internal.R.styleable.ToggleButton_textOn);
        mTextOff = a.getText(com.android.internal.R.styleable.ToggleButton_textOff);
        mDisabledAlpha = a.getFloat(com.android.internal.R.styleable.ToggleButton_disabledAlpha, 0.5f);
        syncTextState();
        a.recycle();
    }
    public ToggleButton(Context context, AttributeSet attrs) {
        this(context, attrs, com.android.internal.R.attr.buttonStyleToggle);
    }
    public ToggleButton(Context context) {
        this(context, null);
    }
    private void syncTextState() {
        boolean checked = isChecked();
        if (checked && mTextOn != null) {
            setText(mTextOn);
        } else if (!checked && mTextOff != null) {
            setText(mTextOff);
        }
    }
    @Override
    public void setChecked(boolean checked) {
        super.setChecked(checked);
        syncTextState();
    }
    public CharSequence getTextOn() {
        return mTextOn;
    }
    public void setTextOn(CharSequence textOn) {
        mTextOn = textOn;
    }
    public CharSequence getTextOff() {
        return mTextOff;
    }
    protected void onFinishInflate() {
        super.onFinishInflate();
        updateReferenceToIndicatorDrawable(getBackground());
    }
    @Override
    public void setBackgroundDrawable(Drawable d) {
        super.setBackgroundDrawable(d);
        updateReferenceToIndicatorDrawable(d);
    }
    private void updateReferenceToIndicatorDrawable(Drawable backgroundDrawable) {
        if (backgroundDrawable instanceof LayerDrawable) {
            LayerDrawable layerDrawable = (LayerDrawable) backgroundDrawable;
            mIndicatorDrawable =
                    layerDrawable.findDrawableByLayerId(com.android.internal.R.id.toggle);
        }
    }
    @Override
    protected void drawableStateChanged() {
        super.drawableStateChanged();
        if (mIndicatorDrawable != null) {
            mIndicatorDrawable.setAlpha(isEnabled() ? NO_ALPHA : (int) (NO_ALPHA * mDisabledAlpha));
        }
    }
}
```

```
public abstract class CompoundButton extends Button implements Checkable {
    private boolean mChecked;
    private int mButtonResource;
    private boolean mBroadcasting;
    private Drawable mButtonDrawable;
    private OnCheckedChangeListener mOnCheckedChangeListener;
    private OnCheckedChangeListener mOnCheckedChangeWidgetListener;

    private static final int[] CHECKED_STATE_SET = {
            R.attr.state_checked
    };
    public CompoundButton(Context context) {
        this(context, null);
    }
    public CompoundButton(Context context, AttributeSet attrs) {
        this(context, attrs, 0);
    }
    public CompoundButton(Context context, AttributeSet attrs, int defStyle) {
        super(context, attrs, defStyle);
        TypedArray a =
                context.obtainStyledAttributes(
                        attrs, com.android.internal.R.styleable.CompoundButton, defStyle, 0);

        Drawable d = a.getDrawable(com.android.internal.R.styleable.CompoundButton_button);
        if (d != null) {
            setButtonDrawable(d);
        }
        boolean checked = a
                .getBoolean(com.android.internal.R.styleable.CompoundButton_checked, false);
        setChecked(checked);
        a.recycle();
   }
      public void toggle() {
        setChecked(!mChecked);
      }
```

```java
@Override
public boolean performClick() {
    /*        * XXX: These are tiny, need some surrounding 'expanded touch area',
     * which will need to be implemented in Button if we only override
     * performClick()        */
    /* When clicked, toggle the state */
    toggle();
    return super.performClick();
}
@ViewDebug.ExportedProperty
public boolean isChecked() {
    return mChecked;
}
/**      * <p>Changes the checked state of this button.</p>
 * @param checked true to check the button, false to uncheck it      */
public void setChecked(boolean checked) {
    if (mChecked != checked) {
        mChecked = checked;
        refreshDrawableState();
        notifyViewAccessibilityStateChangedIfNeeded(
                AccessibilityEvent.CONTENT_CHANGE_TYPE_UNDEFINED);
        // Avoid infinite recursions if setChecked() is called from a listener
        if (mBroadcasting) {
            return;
        }
        mBroadcasting = true;
        if (mOnCheckedChangeListener != null) {
            mOnCheckedChangeListener.onCheckedChanged(this, mChecked);
        }
        if (mOnCheckedChangeWidgetListener != null) {
            mOnCheckedChangeWidgetListener.onCheckedChanged(this, mChecked);
        }
        mBroadcasting = false;
    }
}
```

```
/**      * Register a callback to be invoked when the checked state of this button
   * changes. This callback is used for internal purpose only.     *
   * @param listener the callback to call on checked state change
   * @hide      */
  void setOnCheckedChangeWidgetListener(OnCheckedChangeListener listener) {
     mOnCheckedChangeWidgetListener = listener;
  }
  /**      * Interface definition for a callback to be invoked when the checked state
   * of a compound button changed.      */
  public static interface OnCheckedChangeListener {
     /**         * Called when the checked state of a compound button has changed.          *
      * @param buttonView The compound button view whose state has changed.
      * @param isChecked  The new checked state of buttonView.          */
     void onCheckedChanged(CompoundButton buttonView, boolean isChecked);
  }
  /**     * Set the background to a given Drawable, identified by its resource id.
   *     * @param resid the resource id of the drawable to use as the background      */
  public void setButtonDrawable(int resid) {
     if (resid != 0 && resid == mButtonResource) {         return;       }
     mButtonResource = resid;
     Drawable d = null;
     if (mButtonResource != 0) {    d = getResources().getDrawable(mButtonResource);     }
     setButtonDrawable(d);
  }
  /**      * Set the background to a given Drawable     *
   * @param d The Drawable to use as the background      */
  public void setButtonDrawable(Drawable d) {
     if (d != null) {
        if (mButtonDrawable != null) {
           mButtonDrawable.setCallback(null);
           unscheduleDrawable(mButtonDrawable);
        }
        d.setCallback(this);
        d.setVisible(getVisibility() == VISIBLE, false);
        mButtonDrawable = d;
        setMinHeight(mButtonDrawable.getIntrinsicHeight());
     }
     refreshDrawableState();
  }
```

```
@Override
public void onInitializeAccessibilityEvent(AccessibilityEvent event) {
    super.onInitializeAccessibilityEvent(event);
    event.setClassName(CompoundButton.class.getName());
    event.setChecked(mChecked);
}
@Override
public void onInitializeAccessibilityNodeInfo(AccessibilityNodeInfo info) {
    super.onInitializeAccessibilityNodeInfo(info);
    info.setClassName(CompoundButton.class.getName());
    info.setCheckable(true);
    info.setChecked(mChecked);
}
@Override
public int getCompoundPaddingLeft() {
    int padding = super.getCompoundPaddingLeft();
    if (!isLayoutRtl()) {
        final Drawable buttonDrawable = mButtonDrawable;
        if (buttonDrawable != null) {
            padding += buttonDrawable.getIntrinsicWidth();
        }
    }
    return padding;
}
@Override
public int getCompoundPaddingRight() {
    int padding = super.getCompoundPaddingRight();
    if (isLayoutRtl()) {
        final Drawable buttonDrawable = mButtonDrawable;
        if (buttonDrawable != null) {
            padding += buttonDrawable.getIntrinsicWidth();
        }
    }
    return padding;
}
```

```
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);

    final Drawable buttonDrawable = mButtonDrawable;
    if (buttonDrawable != null) {
        final int verticalGravity = getGravity() & Gravity.VERTICAL_GRAVITY_MASK;
        final int drawableHeight = buttonDrawable.getIntrinsicHeight();
        final int drawableWidth = buttonDrawable.getIntrinsicWidth();

        int top = 0;
        switch (verticalGravity) {
            case Gravity.BOTTOM:
                top = getHeight() - drawableHeight;
                break;
            case Gravity.CENTER_VERTICAL:
                top = (getHeight() - drawableHeight) / 2;
                break;
        }
        int bottom = top + drawableHeight;
        int left = isLayoutRtl() ? getWidth() - drawableWidth : 0;
        int right = isLayoutRtl() ? getWidth() : drawableWidth;

        buttonDrawable.setBounds(left, top, right, bottom);
        buttonDrawable.draw(canvas);
    }
}

@Override
protected int[] onCreateDrawableState(int extraSpace) {
    final int[] drawableState = super.onCreateDrawableState(extraSpace + 1);
    if (isChecked()) {
        mergeDrawableStates(drawableState, CHECKED_STATE_SET);
    }
    return drawableState;
}
```

```
@Override
protected void drawableStateChanged() {
    super.drawableStateChanged();
    if (mButtonDrawable != null) {
        int[] myDrawableState = getDrawableState();
        // Set the state of the Drawable
        mButtonDrawable.setState(myDrawableState);
        invalidate();
    }
}
@Override
protected boolean verifyDrawable(Drawable who) {
    return super.verifyDrawable(who) || who == mButtonDrawable;
}
@Override
public void jumpDrawablesToCurrentState() {
    super.jumpDrawablesToCurrentState();
    if (mButtonDrawable != null) mButtonDrawable.jumpToCurrentState();
}

static class SavedState extends BaseSavedState {
    boolean checked;
    /**         * Constructor called from {@link CompoundButton#onSaveInstanceState()}         */
    SavedState(Parcelable superState) {
        super(superState);
    }
    /**         * Constructor called from {@link #CREATOR}         */
    private SavedState(Parcel in) {
        super(in);
        checked = (Boolean)in.readValue(null);
    }
    @Override
    public void writeToParcel(Parcel out, int flags) {
        super.writeToParcel(out, flags);
        out.writeValue(checked);
    }
```

```
        @Override
        public String toString() {
            return "CompoundButton.SavedState{"
                    + Integer.toHexString(System.identityHashCode(this))
                    + " checked=" + checked + "}";
        }
        public static final Parcelable.Creator<SavedState> CREATOR
                = new Parcelable.Creator<SavedState>() {
            public SavedState createFromParcel(Parcel in) {
                return new SavedState(in);
            }

            public SavedState[] newArray(int size) {
                return new SavedState[size];
            }
        };
    }

    @Override
    public Parcelable onSaveInstanceState() {
        // Force our ancestor class to save its state
        setFreezesText(true);
        Parcelable superState = super.onSaveInstanceState();

        SavedState ss = new SavedState(superState);

        ss.checked = isChecked();
        return ss;
    }

    @Override
    public void onRestoreInstanceState(Parcelable state) {
        SavedState ss = (SavedState) state;

        super.onRestoreInstanceState(ss.getSuperState());
        setChecked(ss.checked);
        requestLayout();
    }
}
```

附录7、课件所学的控件

【安卓系统】 2

概述

安卓系统结构

Dalvik虚拟机

.dex JIT ART模式

app的四大组件

Activity Service

ContentProvider

BroadcastReceiver

(Intent)

app的基本结构

界面程序 界面配置

清单文件 资源文件

第一个Android程序

【基本属性】 13

定义id

android:id="@+id/tv"

定义颜色

Color.argb()

0x80FF0000 #80FF0000

@color/colorPrimary

尺寸单位

px dp sp pt

设置字体

fontFamily textStyle

typeFace

设置宽度和高度

height width

padding layout_margin

lines ems letterSpacing

布局比重

layout_weight

文本设置

text textColor textSize

textStyle

shadowDx shadowDy

shadowColor shadowRadius

autolink linksClickable

textColorLink

textAllCaps textIsSelectable

textColorHighlight

textSacaleX

文本对齐

gravity

(top left

center_vertical center)

textAlignment

（textStart center viewStart gravity）

省略文本

ellipsize

(none marquee start end middle）

背景设置

background

tv.setBackgroundColor()

drawableLeft

drawableTop

bitmap tileMode repeat

shape solid stroke

【基本控件】 34

切换按钮(ToggleButton)

android:textOff=""

android:textOn=""

android:checked="true"

setOnCheckedChangeListener()

开关按钮(Switch)

android:thumb

android:track

android:switchMinWidth

android:checked

setOnCheckedChangeListener()

复选框(CheckBox)

android:checked

setOnCheckedChangeListener()

单选按钮(RadioButton)

RadioGroup

radiogroup.setOnCheckedChangeListener()

DisplayToast()

可选文本框(CheckedTextView)

android:checked

android:clickable

android:checkMark

评价条(RatingBar)

android:stepSize

android:numStars

android:rating

setOnRatingBarChangeListener()

拖动条(SeekBar)

android:max

android:progress

setOnSeekBarChangeListener()

进度条(ProgressBar)

android:max
android:progress
android:secondaryProgress
android:progressDrawable
android:indeterminate
android:indeterminateDrawable
shape
CirclePgBar（自定义控件）

图像框(ImageView)

ImageButton
android:src android:scaleType
app:srcCompat
android:background

编辑框(EditText)

android:inputType
  (textPassword
   textMultiLine
   textAutoComplete)
android:hint
android:textColorHint
android:textCursorDrawable
android:cursorVisible
android:selectAllOnFocus
android:enabled

联系人控件(QuickContactBadge)

assignContactFromPhone()

日期选择器(DatePicker)

setMinDate()
setMaxDate()
init()
OnDateChangedListener()

时间选择器(TimePicker)

setIs24HourView()
setCurrentHour()
setCurrentMinute()
setHour()
setMinute()
setOnTimeChangedListener()

【附录】 85

附录1、安卓项目版本修改
附录2、系统命名颜色
附录3、系统主题Theme列表
附录4、控件大全
附录5、View的变换
附录6、课件所学的控件

view(视图) -- box(框)