去掉title和去掉状态栏

**概述：**

**方法一：代码设置**

功能：去掉标题栏和状态栏(电池电量及信号栏)

（1）全局去掉

在manifest中的<application>标签下直接加入

android:theme="@android:style/Theme.NoTitleBar.Fullscreen"

会让应用中的所有activity都全屏

（2）单个activity去掉标题栏和状态栏

2.1去掉标题栏

requestWindowFeature(Window.FEATURE\_NO\_TITLE);

2.1.1当继承自Activity或ActionBarActivity时，在setContentView(...)方法前调用requestWindowFeature(Window.FEATURE\_NO\_TITLE);

2.1.2当继承自AppCompatActivity时，在setContentView(...)方法前调用

supportRequestWindowFeature(Window.FEATURE\_NO\_TITLE);

注意：当继承自AppCompatActivity的如上设置无效时，可以采用修改主题的方法。

2.1.3当标题栏是ActionBar 时，直接隐藏掉即可（有待测试）

//隐藏标题栏，但是上面的状态栏（时间、电池等显示的状态栏）还是存在的。

ActionBar actionBar = getSupportActionBar();  
 //ActionBar actionBar = getActionBar();

//ActionBar actionBar = getActivity().getActionBar();

if(actionBar !=null){

actionBar.hide();

}

2.2去掉状态栏（任选一种）

2.2.1 这种方式会去掉状态栏，但默认也会去掉虚拟按键的背景。

getWindow().setFlags(WindowManager.LayoutParams.TYPE\_STATUS\_BAR, WindowManager.LayoutParams.TYPE\_STATUS\_BAR);

2.2.2 这种方式去掉状态栏，对虚拟按键没有影响。（建议使用）

getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,

WindowManager.LayoutParams.FLAG\_FULLSCREEN);

注意：最上面代码需要放在setContentView()方法前。

**方法二：修改主题风格**

创建项目后，系统会默认在清单文件中指定一个主题AppTheme或者其他主题，设置这个主题会影响整个应用的主题，即就是全局主题，所以我们从这里的AppTheme主题点进去，就可以修改对应的主题风格，包括设置标题栏的有无和颜色，光标的颜色等等信息。另外，如果需要修改某一个页面的主题，可以在这个页面清单文件对应的节点里，添加主题。

原因：在Android 3.0中除了新增了Fragment外，还增加了Action Bar一个非常重要的交互元素，Action Bar取代了传统的titlebar和menu。如果需要去掉Actionbar，一种方式：在你的manifest文件中将Activity的属性中设置主题风格为NoTitleBar，另一种：在运行时调用hide()方法也可以隐藏ActionBar，调用show()方法来显示ActionBar()

注意：如果使用一个主题（theme）来移除Activity上得ActionBar，那么窗口将不再会有ActionBar，因此在运行时也就没有办法来添加ActionBar。



实例：修改AppTheme的风格：

第一步：在manifest中的<application>标签下直接加入主题。（这是全局的修改）

在项目/app/src/main/res/values/styles或者colors文件中，直接重写这个style中的Item色值，就可以改变系统自带的颜色和修改标题栏。

<style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">

        <!-- Customize your theme here. -->

<!-- 设置标题栏Toolbar等和整个应用的颜色基调 -->

        <item name="colorPrimary">@color/colorPrimary</item>

<!-- 设置状态栏颜色 -->

        <item name="colorPrimaryDark">@color/colorPrimaryDark</item>

<!-- 设置Editor光标、输入框颜色. -->

        <item name="colorAccent">@color/colorAccent</item>

        <!-- 去掉ActionBar. -->

        <item name="android:windowActionBar">false</item>

<!-- 去掉标题栏. -->

        <item name="android:windowNoTitle">true</item>

<!-- 设置底部虚拟导航栏的背景为半透明. -->

<item name="android:windowTranslucentNavigation">true</item>

<item name="android:windowTranslucentStatus">true</item>

</style>

注意：这里的主题修改，会影响全局。

第二步：在需要修改主题的<activity>标签下，设置自己的主题风格，不影响全局主题。

下面是在activity中设置没有标题栏的主题

当这个Activity继承AppCompatActivity时，推荐使用Theme.AppCompat主题，如:

android:theme="@style/Theme.AppCompat.NoActionBar"

android:theme="@style/Theme.AppCompat.Light.NoActionBar"

当这个Activity继承其他Activity时，使用如下主题:

android:theme="@android:style/Theme.Black.NoTitleBar" 黑色背景

android:theme="@android:style/Theme.Light.NoTitleBar" 白色背景

android:theme="@android:style/Theme.NoTitleBar"

android:theme="@android:Theme.Holo.Light.NoActionBar" 白底（包含alertDialog）

**方法三：自定义标题布局**

|  |
| --- |
| protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  //预先设置允许改变的窗口状态，需在 setContentView 之前调用，否则设置标题时抛运行时错误。  requestWindowFeature(Window.FEATURE\_CUSTOM\_TITLE);  setContentView(R.layout.custom\_title);  //标题区域可设置为 layout ，如此可以有丰富的展现方式  getWindow().setFeatureInt(Window.FEATURE\_CUSTOM\_TITLE,R.layout.custom\_title\_1);  } |

res\layout\custom\_title\_1.xml 包含一个TextView 用于显示标题。

|  |
| --- |
| <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" android:id="@+id/screen"  android:layout\_width="fill\_parent"  android:layout\_height="fill\_parent"  android:orientation="vertical">  <TextView android:id="@+id/left\_text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:layout\_alignParentLeft="true"  android:text="@string/custom\_title\_left" />  </RelativeLayout> |

**方法四：自定义标题内容**

|  |
| --- |
| <activity android:name=".activity.MainActivity"  android:screenOrientation="portrait"  android:label="@string/titlebar\_text"  </actibity>    requestWindowFeature(Window.FEATURE\_NO\_TITLE);  //设置窗口无标题栏  setContentView(R.layout.main);  //动态设置标题的值，getTitle()的值是该activity的声明中android:label的值  ((TextView) findViewById(R.id.titlebar\_text)).setText(getTitle());  其中，getTitle()取得的值就是上述 android:label="@string/titlebar\_text" 的值 |

**方法五：透明状态栏和导航栏**

方式一：可以在styles.xml中设置

<!--设置导航栏/状态栏窗口color为透明-->

<item name="android:statusBarColor">@android:color/transparent</item>

<item name="android:navigationBarColor">@android:color/transparent</item>

<!--设置导航栏/状态栏窗口color为半透明-->

<item name="android:windowTranslucentStatus" tools:targetApi="kitkat">true</item>

<item name="android:windowTranslucentNavigation" tools:targetApi="kitkat">true</item>

方式二：

this.getWindow().addFlags(WindowManager.LayoutParams.FLAG\_TRANSLUCENT\_STATUS); this.getWindow().addFlags(WindowManager.LayoutParams.FLAG\_TRANSLUCENT\_NAVIGATION); this.getWindow().setStatusBarColor(blackColor);

this.getWindow().setNavigationBarColor(blackColor);

<item name="windowBackground"> 窗体的背景 </item>  
<item name="windowFrame">Dialog 是否有边框 </item>  
<item name="windowNoTitle">窗体是否有标题</item>  
<item name="windowFullscreen">false</item>是否为全屏  
<item name="windowOverscan">false</item>是否要求窗体铺满整屏幕                 
<item name="windowIsFloating">false</item>        窗体是否浮在下层之上  
<item name="windowContentOverlay">@null</item>    设置窗体内容背景  
<item name="windowShowWallpaper">false</item>     是否显示壁纸  
<item name="windowTitleStyle">[@android](http://my.oschina.net/asia" \t "https://blog.csdn.net/feitian145623/article/details/_blank) :style/WindowTitle</item> 窗体的标题栏Style  
<item name="windowTitleSize">25dip</item>    窗体文字大小

窗体标题栏背景  
<item name="windowTitleBackgroundStyle">[@android](http://my.oschina.net/asia" \t "https://blog.csdn.net/feitian145623/article/details/_blank):style/WindowTitleBackground</item>            窗体切换时的动画样式  
<item name="android:windowAnimationStyle">[@android](http://my.oschina.net/asia" \t "https://blog.csdn.net/feitian145623/article/details/_blank) :style/Animation.Activity</item>

在使用输入法时窗体的适配  
<item name="android:windowSoftInputMode">stateUnspecified|adjustUnspecified</item>  
<item name="windowActionBar">false</item>是否打开ActionBar  
<item name="windowActionModeOverlay">false</item>这个暂时不知道，希望大神能回答一下。  
<item name="windowCloseOnTouchOutside">false</item> 是否再点击外部可关闭

<item name="android:windowIsTranslucent">false</item>    窗体是否半透明  
<item name="android:backgroundDimEnabled">false</item>: 背景是否模糊显示

<item name="android:windowContentOverlay">@null</item>  自定义Titlebar时去掉多余的阴影。