



链滴

# Android小知识总结

作者: [idxiu](#)

原文链接: <https://ld246.com/article/1470030070401>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

<h2 style="font-size:22px;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;color:#555555;text-align:justify;background-color:#FFFFFF;">  
 onCreate中计算View的高度  
 </h2>

<p>  
 <span style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;font-size:14px;line-height:28px;background-color:#FFFFFF;">可以借助getViewTreeObserver().addOnGlobalLayoutListener()来获得宽度或者高度，示例代码如下： </span>  
 </p>

<p>  
 <br />  
 </p>

```

<pre class="prettyprint lang-java">mWebView.getViewTreeObserver().addOnGlobalLayoutListener(new ViewTreeObserver.OnGlobalLayoutListener() {
    @Override
    public void onGlobalLayout() {
        mWebView.getWidth();
        mWebView.getHeight();
        //防止调用多次
        mWebView.getViewTreeObserver().removeOnGlobalLayoutListener(this);
    }
});</pre>

```

<h3>  
 <span style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;font-size:22px;line-height:1.5;background-color:#FFFFFF;">PopupWindow</span>  
 <span style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;font-size:22px;line-height:1.5;background-color:#FFFFFF;">w计算高度</span>  
 </h3>

<p>  
 <span style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;font-size:14px;font-weight:normal;line-height:1.5;background-color:#FFFFFF;">PopupWindow在始化创建好后，我想知道它的高度和宽度，可以这样计算： </span>  
 <span style="font-weight:normal;background-color:#FFFFFF;">&nbsp;&nbsp;&nbsp;</span>  
 </p>

```

<pre class="prettyprint lang-java">popupWindow.getContentView().measure(0, 0);
popupWindow.getContentView().getMeasuredWidth();
popupWindow.getContentView().getMeasuredHeight(); </pre>

```

<h3>  
 <span style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;font-size:22px;line-height:1.5;background-color:#FFFFFF;">Theme.NoDisplay</span>  
 </h3>

<p>  
 <span style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;font-size:22px;line-height:1.5;background-color:#FFFFFF;"><span style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;font-size:14px;line-height:21px;background-color:#FFFFFF;">在Android中想进行一些无界面的处理又不适合使用service时，此时可以在项目的androidManifest.xml文件中相应的Activity标签中添加这样一行： </span>  
 </span>  
 </p>

```

<pre class="prettyprint lang-java">android:theme="@android:style/Theme.NoDisplay </pre>

```

</h3>

<h3 style="font-size:22px;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;color:#555555;text-align:justify;background-color:#FFFFFF;">

activity父级activity的重建

</h3>

<p style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;font-size 14px;text-align:justify;background-color:#FFFFFF;">

当我们从其他App跳转到我们应用内部时，可能我们跳转到的不是应用的第一层级页面而是深层次页面，这时候我们需要在关闭应用内部页面时，对高层的页面进行重建。

</p>

<ol style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;font-size:14px;text-align:justify;background-color:#FFFFFF;">

<li>

<p>

为activity指定父级activity

</p>

</li>

</ol>

```
<pre class="prettyprint lang-java">&lt;application&gt;
```

```
&lt;!-- 首层activity --&gt;
```

```
&lt;activity android:name="com.example.Mainactivity"/&gt;
```

```
&lt;!-- 第二层级activity --&gt;
```

```
&lt;activity android:name="com.example.SecondeActivity" android:parentActivityName="om.example.Mainactivity" &gt;
```

```
&lt;!-- 父activity的meta-data, 用来支持4.0以下版本 --&gt;
```

```
&lt;meta-data android:name="android.support.PARENT_ACTIVITY" android:value="com example.Mainactivity" /&gt;
```

```
&lt;/activity&gt;
```

```
&lt;/application&gt; </pre>
```

```
<pre> 二级页面返回时，重建任务栈 </pre>
```

```
<pre class="prettyprint lang-java">@Override
```

```
public void onBackPressed() {
```

```
// 获得指向父级activity的intent, NavUtils在support v4 包中
```

```
Intent upIntent = NavUtils.getParentActivityIntent(this);
```

```
// 判断是否需要重建任务栈
```

```
if (NavUtils.shouldUpRecreateTask(this, upIntent)) {
```

```
// 这个activity不是这个app任务的一部分, 所以当向上导航时创建
```

```
// 用合成后退栈(synthesized back stack)创建一个新任务。
```

```
TaskStackBuilder.create(this)
```

```
// 添加这个activity的所有父activity到后退栈中
```

```
.addNextIntentWithParentStack(upIntent)
```

```
// 向上导航到最近的一个父activity
```

```
.startActivities();
```

```
} else {
```

```
// 这个activity是这个app任务的一部分, 所以
```

```
// 向上导航至逻辑父activity.
```

```
NavUtils.navigateUpTo(this, upIntent);
```

```
}
```

```
super.onBackPressed();
```

```
} </pre>
```

<h3>

兼容低版本修改图片的tintColor

</h3>

</pre>

```
<pre class="prettyprint lang-java">drawable.setColorFilter(Color.RED,PorterDuff.Mode.SRC_I); </pre>
```

<h1 style="font-size:22px;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;color:#

```
55555;text-align:justify;background-color:#FFFFFF;">
```

使用dumpsys命令查看系统信息

```
</h1>
```

```
<p style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;font-size 14px;text-align:justify;background-color:#FFFFFF;">
```

dumpsys命令很强大，可以查看当前系统的信息，用的还比较多，dumpsys常用命令如下：

```
</p>
```

```
<pre class="prettyprint lang-java">dumpsys [options]
```

```
  meminfo 显示内存信息
```

```
  cpuinfo 显示cpu信息
```

```
  account 显示accounts信息
```

```
  activity 显示所有的activities信息
```

```
  window 显示键盘、窗口信息
```

```
  wifi 显示wifi信息 </pre>
```

```
<pre><span style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-seri ;font-size:14px;line-height:28px;background-color:#FFFFFF;">例如：查看某个程序内部内存信息 </span></pre>
```

```
<p>
```

```
<s> </s><span style="font-family:'sans serif', tahoma, verdana, helvetica;line-height:1.5;"> db shell dumpsys meminfo 包名 or pid </span>
```

```
</p>
```

```
<span>再或者查看当前打开app的页面信息，这个很有用 </span>
```

```
<pre class="prettyprint lang-java">adb shell dumpsys window windows | grep -E 'mCurrentF cus' </pre>
```

```
<pre><span>结果类似如下： </span></pre>
```

```
<pre class="prettyprint lang-java">mCurrentFocus=Window{43469d60 u0 com.jikexueyuan.g ekacademy.debug/com.jikexueyuan.geekacademy.ui.activity.ActivityHomeV3} </pre>
```

```
<h3 style="font-size:22px;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;color:# 55555;text-align:justify;background-color:#FFFFFF;">
```

```
<span style="color:#555555;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;fo t-size:22px;line-height:1.5;background-color:#FFFFFF;">Recycleview的LayoutManager设置成St ggeredGridLayoutManager后获取某个字views的</span>spanIndex（也就是在第几列） </h3>
```

```
<pre class="prettyprint lang-java">private static class MyItemDecoration extends RecyclerView.I w.ItemDecoration { @Override public void getItemOffsets(Rect outRect, View view, RecyclerView parent, RecyclerView.Stat e) {
```

```
    super.getItemOffsets(outRect, view, parent, state);
```

```
    int index = ((StaggeredGridLayoutManager.LayoutParams) view.getLayoutParams()).getS anIndex();
```

```
    }
```

```
</pre>
```

```
<h3 style="font-size:22px;font-family:Lato, 'PingFang SC', 'Microsoft YaHei', sans-serif;color:# 55555;text-align:justify;background-color:#FFFFFF;">
```

```
<span style="font-weight:normal;line-height:1.5;">如果获取当前手机为每个app分配的堆大 ，也就是内存大小</span>
```

```
</h3>
```

```
<pre class="prettyprint lang-java">ActivityManager manager = (ActivityManager)getSystem ervice(Context.ACTIVITY_SERVICE);
```

```
int heapSize = manager.getMemoryClass(); </pre>
```

```
<pre>注意：这里获得heapSize是M为单位返回的
```

```
<pre class="prettyprint lang-java">
```

```
  duplicate entry: android/support/annotation/ColorRes.class
```

