



链滴

Android 小知识集合

作者: [zjs](#)

原文链接: <https://ld246.com/article/1504101017703>

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 id="V7包中的Theme-AppCompat主题系列中并没有全屏样式">V7 包中的 Theme.AppCompat 主题系列中并没有全屏样式</h2>

<p>解决方法

{% codeblock lang:xml %}</p>

<p>{% endcodeblock %}</p>

<hr>

<h2 id="SharedPreferences-Editor的apply和commit方法异同">SharedPreferences.Editor 的 apply 和 commit 方法异同</h2>

<p>SharedPreferences.Editor 官方文档 apply 方法</p>

<blockquote>

<p>Commit your preferences changes back from this Editor to the SharedPreferences object it is editing. This atomically performs the requested modifications, replacing whatever is currently in the SharedPreferences.</p>

<p>Note that when two editors are modifying preferences at the same time, the last one to call apply wins.</p>

<p>Unlike commit(), which writes its preferences out to persistent storage synchronously, apply() commits its changes to the in-memory SharedPreferences immediately but starts an asynchronous commit to disk and you won't be notified of any failures. If another editor on this SharedPreferences does a regular commit() while a apply() is still outstanding, the commit() will block until all async commits are completed as well as the commit itself.</p>

<p>As SharedPreferences instances are singletons within a process, it's safe to replace any instance of commit() with apply() if you were already ignoring the return value.</p>

<p>You don't need to worry about Android component lifecycles and their interaction with apply() writing to disk. The framework makes sure in-flight disk writes from apply() complete before switching states.</p>

</blockquote>

<hr>

<blockquote>

<p>The SharedPreferences.Editor interface isn't expected to be implemented directly. However, if you previously did implement it and are now getting errors about missing apply(), you can simply call commit() from apply().</p>

</blockquote>

<p>Google 翻译</p>

<blockquote>

<p>将您的首选项从本编辑器提交到正在编辑的 SharedPreferences 对象。这原子地执行请求的修改，替换当前在 SharedPreferences 中的任何内容。</p>

<p>请注意，当两个编辑器同时修改首选项时，最后一个调用应用胜利。</p>

<p>与 commit () 一样，它将其优先级同步写入永久存储器，apply () 将立即将其更改提交给内存中的 SharedPreferences，但是会启动到磁盘的异步提交，并且不会收到任何失败的通知。如果这个 SharedPreferences 上的另一个编辑器在 apply () 仍然未完成时执行常规 commit ()，则 commit () 将阻塞，直到所有异步提交完成以及提交本身。</p>

<p>由于 SharedPreferences 实例是一个进程中的单例，如果您已经忽略了返回值，那么可以使用 apply () 替换任何 commit () 实例。</p>

<p>您不必担心 Android 组件生命周期及其与 apply () 写入磁盘的交互。该框架确保在开关状态前，从 apply () 完成的空中磁盘写入。</p>

</blockquote>

<hr>

<blockquote>

<p>SharedPreferences.Editor 接口不期望直接实现。但是，如果您以前已经实现了它，并且现在遇到有关缺少 apply () 的错误，可以直接从 apply () 调用 commit ()。</p>

</blockquote>

apply 没有返回值而 commit 返回 boolean 表明修改是否提交成功

apply 是将修改数据原子提交到内存, 而后异步真正提交到硬件磁盘, 而 commit 是同步的提交到硬件磁盘, 因此, 在多个并发的提交 commit 的时候, 他们会等待正在处理的 commit 保存到磁盘后操作, 从而降低了效率。而 apply 只是原子的提交到内容, 后面有调用 apply 的函数的将会直接覆前面的内存数据, 这样从一定程度上提高了很多效率。

apply 方法不会提示任何失败的提示。

由于在一个进程中, sharedPreferences 是单实例, 一般不会出现并发冲突, 如果对提交的结果关心的话, 建议使用 apply, 当然需要确保提交成功且有后续操作的话, 还是需要用 commit 的。

<hr>

<h2 id="dp转化px-px转化dp">dp 转化 px, px 转化 dp</h2>

<p>{% codeblock lang:Java %}

import android.content.Context;</p>

<p>public class DensityUtil {</p>

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<pre><code class="highlight-chroma"> * 根据手机的分辨

<p>做图片引导页使用大分辨率图片, 一个 png 图片达到 700kb, 不容小视, app 压缩故从图片压起

例子, 5 张高分辨率图片, 分辨率为 720*1280, 大小

{% asset_img appzip.jpg appzip %}

转 webp 图片 (智图)

{% asset_img appzip3.jpg appzip3 %}</p>