

# Android 样式系统 | 常见的主题背景属性

原创 Android 谷歌开发者 7月11日

来自专辑

Android 开发技术



在前一篇 [Android 样式系统](#)文章中，我们介绍了主题背景与样式的区别，以及如何编写灵活的样式与布局代码用于抽离可变化部分。

我们建议使用主题背景属性来间接引用资源，您可以在不同的模式下 (比如在[深色主题背景](#)) 实现灵活地切换。如果您发现在布局或样式代码中直接引用了资源或者是硬编码了具体的值，请考虑使用主题背景属性来替代之前用法。

- 深色主题背景  
<https://developer.android.google.cn/guide/topics/ui/look-and-feel/darktheme>

```
1 <!-- Copyright 2019 Google LLC.
2     SPDX-License-Identifier: Apache-2.0 -->
3 <ConstraintLayout ...
4 -   android:foreground="@drawable/some_ripple"
5 -   android:background="@color/blue" />
6 +   android:foreground="?attr/selectableItemBackground"
7 +   android:background="?attr/colorPrimarySurface" />
```

但是我们还可以使用哪些主题背景属性的功能呢？这篇文章列举了您应该知道的关于主题背景属性的通用功能，它们广泛应用在 Material、AppCompact，或者是平台 (Platform) 中。本文并未完整列举所有属性，只列举了我所使用的，建议您浏览下面的属性文件的定义链接来获取更多信息。

## 颜色

这些颜色大部分来自于 [Material 颜色系统](#) (Material color system)，它们给每个颜色取了语义化的名称可以让您在应用中使用它们 ([体现为主题背景属性](#))。

- [Material 颜色系统](#)  
<https://material.io/design/color/the-color-system.html#color-usage-and-palettes>
- [体现为主题背景属性](#)  
<https://material.io/develop/android/theming/color/>



- `?attr/colorPrimary` 应用的主要颜色；
- `?attr/colorSecondary` 应用的次要颜色，通常作为主要颜色补充；
- `?attr/colorOn[Primary, Secondary, Surface etc]` 对应颜色的相反色；
- `?attr/color[Primary, Secondary]Variant` 给定颜色的另一种阴影；
- `?attr/colorSurface` 部件的表面颜色，如: 卡片、表格、菜单；
- `?android:attr/colorBackground` 屏幕的背景颜色；
- `?attr/colorPrimarySurface` 在浅色主题中的 `colorPrimary` 与深色主题背景中的 `colorSurface` 中做切换；
- `?attr/colorError` 显示错误时的颜色。

其他常用的颜色:

- `?attr/colorControlNormal` 正常状态下设置给 icon/controls 的颜色；
- `?attr/colorControlActivated` 激活模式下设置给 icons/controls 的颜色 (如: 单选框被勾选)；
- `?attr/colorControlHighlight` 设置给高亮控制界面的颜色 (如: ripples，列表选择器)；
- `?android:attr/textColorPrimary` 设置给文本的主要颜色；
- `?android:attr/textColorSecondary` 设置给文本的次要颜色。

大小

- `?attr/listPreferredItemHeight` 列表项的标准高度 (最小值)；
- `?attr/actionBarSize` 工具栏的高度。

Drawables

- `?attr/selectableItemBackground` 可交互条目在 ripple 或者是高亮时的背景颜色 (针对外观)；
- `?attr/selectableItemBackgroundBorderless` 无边界的 ripple；
- `?attr/dividerVertical` 用于垂直分割可视化元素的 drawable；
- `?attr/dividerHorizontal` 用于水平分割可视化元素的 drawable。

TextAppearance

Material 定义了缩放类型，它是在整个应用中使用的一组由文本样式组成的离散集合，集合中的每个值都是一个主题背景属性，可以被设置为 textApperance。请点击 [Material type scale generator](#) 获得更多关于生成不同字体缩放的帮助。

- [Material type scale generator](#)  
<https://material.io/design/typography/the-type-system.html#type-scale>



- ?attr/textAppearanceHeadline1 默认为 96sp light 文本；
- ?attr/textAppearanceHeadline2 默认为 60sp light 文本；
- ?attr/textAppearanceHeadline3 默认为 48sp regular 文本；
- ?attr/textAppearanceHeadline4 默认为 34sp regular 文本；
- ?attr/textAppearanceHeadline5 默认为 24sp regular 文本；
- ?attr/textAppearanceHeadline6 默认为 20sp medium 文本；
- ?attr/textAppearanceSubtitle1 默认为 16sp regular 文本；
- ?attr/textAppearanceSubtitle2 默认为 14sp medium 文本；
- ?attr/textAppearanceBody1 默认为 16sp regular 文本；
- ?attr/textAppearanceBody2 默认为 14sp regular 文本；
- ?attr/textAppearanceCaption 默认为 12sp regular 文本；
- ?attr/textAppearanceButton 默认为 14sp 全大写 medium 文本；
- ?attr/textAppearanceOverline 默认为 10sp 全大写 regular 文本。

## 形状

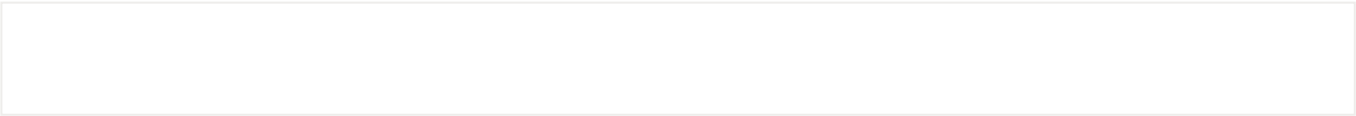
Material 采用了形状系统 ([Shape system](#))，它是由主题背景属性[实现](#)了 small、medium、large 等不同的部件。请注意，如果您想给自定义的部件设置形状外观，您应该使用 MaterialShapeDrawable 作为它的背景，因为它能够理解并能实现具体形状。

- [Shape system](#)  
<https://material.io/design/shape/>
- [实现](#)  
<https://material.io/develop/android/theming/shape/>



- `?attr/shapeAppearanceSmallComponent` 默认圆角为 4dp，用于 Buttons、Chips、TextFields 等；
- `?attr/shapeAppearanceMediumComponent` 默认圆角为 4dp，用于 Cards、Dialogs、Date Pickers 等；
- `?attr/shapeAppearanceLargeComponent` 默认圆角为 0dp (其实是方形)，用于 Bottom Sheets 等。

## 按钮风格



Material 提供了三种不同类型的按钮: [Contained](#)、[Text](#) 以及 [Outlined](#)。MDC 提供了主题背景属性，您可以使用它们给 `MaterialButton` 设置样式:

- `?attr/materialButtonStyle defaults` 默认是 Contained 类型 (或者直接省略样式)；
- `?attr/borderlessButtonStyle` 设置为 Text 样式的按钮；
- `?attr/materialButtonOutlinedStyle` 设置为 Outlined 样式的按钮。

- [Contained](#)  
<https://material.io/components/buttons/#contained-button>
- [Text](#)  
<https://material.io/components/buttons/#text-button>
- [Outlined](#)  
<https://material.io/components/buttons/#outlined-button>

## Floats

- `?android:attr/disabledAlpha` 默认关闭 Widget 的 alpha；
- `?android:attr/primaryContentAlpha` 设置给 foreground 元素的 alpha 值；
- `?android:attr/secondaryContentAlpha` 设置给 secondary 元素的 alpha 值。

## 应用命名空间 vs Android 命名空间

您可能注意到有些属性的引用是通过 `?android:attr/foo` 而有些只是通过 `?attr/bar`。这是因为一些属性是由 Android 平台定义的，所以您需要使用 `android` 命名空间来引用由它们自己定义的属

**性** (类似于布局中使用 View 属性 android:id) 。编译到您的应用但不是来自于静态库的属性 (AppCompact 或者 MDC) ，使用它们时不需要命名空间 (类似于布局中使用 app:baz) 。平台跟库有时候定义了相同的属性，如 colorPrimary。这时候系统优先使用非平台版本的属性，它们可以被所有级别的 API 使用。为了向后兼容，它们会被完整的复制到库中。我在上面列举的都是非平台版本的案例。

优先使用非平台版本的属性，它们可以被所有级别的 API 使用

更多资源

为了获取可以使用的全部主题背景属性，请查阅以下信息：

- [Android platform](https://android.googlesource.com/platform/frameworks/base/+/refs/heads/master/core/res/res/values/attrs.xml)  
https://android.googlesource.com/platform/frameworks/base/+/refs/heads/master/core/res/res/values/attrs.xml
- [AppCompat](https://android.googlesource.com/platform/frameworks/support/+/androidx-master-dev/appcompat/appcompat/src/main/res/values/attrs.xml)  
https://android.googlesource.com/platform/frameworks/support/+/androidx-master-dev/appcompat/appcompat/src/main/res/values/attrs.xml

Material 设计的部件：

- [Color](https://github.com/material-components/material-components-android/blob/master/lib/java/com/google/android/material/color/res/values/attrs.xml)  
https://github.com/material-components/material-components-android/blob/master/lib/java/com/google/android/material/color/res/values/attrs.xml
- [Shape](https://github.com/material-components/material-components-android/blob/master/lib/java/com/google/android/material/shape/res/values/attrs.xml)  
https://github.com/material-components/material-components-android/blob/master/lib/java/com/google/android/material/shape/res/values/attrs.xml
- [Type](https://github.com/material-components/material-components-android/blob/master/lib/java/com/google/android/material/typography/res/values/attrs.xml)  
https://github.com/material-components/material-components-android/blob/master/lib/java/com/google/android/material/typography/res/values/attrs.xml

自己动手

当您想使用主题背景功能抽象某个东西的时候，发现没有现成的主题背景可用时，您可以自定义一个。您可以参考 [Google I/O 应用](#)，它实现了在两个界面中显示主题演讲的列表：



- [Google I/O 应用](https://github.com/google/iosched)  
https://github.com/google/iosched

这两个界面大部分看起来比较相似，除了左边界面有个显示时间的功能而右边是没有的。

将 item 的对齐部分抽象成一个主题背景属性，给不同界面使用的同一个布局中使用主题背景来区分它们的差异：

1. 在 `attrs.xml` 中定义主题背景属性：

- [attrs.xml](#)  
<https://github.com/google/iosched/blob/89df01ebc19d9a46495baac4690c2ebfa74946dc/mobile/src/main/res/values/attrs.xml#L41>

```
1 <!-- Copyright 2019 Google LLC.
2     SPDX-License-Identifier: Apache-2.0 -->
3 <attr name="sessionListKeyline" format="dimension" />
```

2. 在不同的主题背景中使用不同的值：

- [不同的值 1](#)  
<https://github.com/google/iosched/blob/89df01ebc19d9a46495baac4690c2ebfa74946dc/mobile/src/main/res/values/themes.xml#L51>
- [不同的值 2](#)  
<https://github.com/google/iosched/blob/89df01ebc19d9a46495baac4690c2ebfa74946dc/mobile/src/main/res/values/themes.xml#L78>

```
1 <!-- Copyright 2019 Google LLC.
2     SPDX-License-Identifier: Apache-2.0 -->
3 <style name="Theme.IOSched.Schedule">
4     ...
5     <item name="sessionListKeyline">72dp</item>
6 </style>
7
8 <style name="Theme.IOSched.Speaker">
9     ...
10    <item name="sessionListKeyline">16dp</item>
11 </style>
```

3. 给两个界面使用的布局文件中使用主题背景属性：

- [使用主题背景属性](#)  
[https://github.com/google/iosched/blob/89df01ebc19d9a46495baac4690c2ebfa74946dc/mobile/src/main/res/layout/item\\_session.xml#L61](https://github.com/google/iosched/blob/89df01ebc19d9a46495baac4690c2ebfa74946dc/mobile/src/main/res/layout/item_session.xml#L61)

```
1 <!-- Copyright 2019 Google LLC.
2     SPDX-License-Identifier: Apache-2.0 -->
3 <Guideline ...
4     app:layout_constraintGuide_begin="?attr/sessionListKeyline" />
```

保持探索

了解了能够使用的主题背景属性功能后，您可以在编写布局、样式、drawables 时使用它们。

使用主题背景属性功能更容易实现主题功能 (如[深色主题背景](#))，而且让您编写出更灵活，更易于维护的代码。更多关于此部分的内容，敬请关注本系列的下一篇文章。

- [深色主题背景](#)  
<https://developer.android.google.cn/guide/topics/ui/look-and-feel/darktheme>