

公共控件使用说明

1. 主标题栏: HeyMainTitleBar.....	3
2. 返回标题栏: HeyBackTitleBar.....	4
3. 取消标题栏: HeyCancelTitleBar.....	5
4. 应用标题栏: HeyAppTitleBar.....	6
5. 通知标题栏: HeyAppTitleBar.....	7
6. 加载标题栏: HeyLoadingTitleBar.....	8
7. 单行默认列表: HeySingleDefaultItem.....	9
8. 单行带开关列表: HeySingleItemWithSwitch.....	10
9. 单行带复选框列表: HeySingleItemWithCheckBox.....	11
10. 单行两个图标列表: HeySingleDefaultItem.....	12
11. 单行带摘要列表: HeySingleItemWithSummary.....	13
12. 单行带左边图标列表: HeySingleItemLeftImage.....	14
13. 双行默认列表: HeyMultipleDefaultItem.....	15
14. 双行带开关列表: HeyMultipleItemWithSwitch.....	16
15. 双行带复选框列表: HeyMultipleItemWithCheckBox.....	17
16. 双行两个图标列表: HeyMultipleDefaultItem.....	18
17. 双行带左边图标列表: HeyMultipleItemWithLeftImage.....	19
18. 圆角按钮: HeyShapeButton.....	20
19. 图标按钮: HeyGradientButton.....	22
20. 下载按钮: HeyProgressButton.....	24
21. 单组数字选择器: HeyNumberPicker.....	27
22. 上、下午数字选择器: HeyTimePicker.....	28
23. 双组数字选择器: HeyTimePicker.....	29
24. 时分秒选择器: HeyCounterPicker.....	30
25. 翻页符: HeyPageIndicator.....	31
26. 对话框(一个按钮): HeyDialog.....	34
27. 对话框(两个按钮): HeyDialog.....	35
28. 对话框(权限隐私申明): HeyDialog.....	36
29. 对话框(自定义布局): HeyDialog.....	37
30. 开关: HeySwitch.....	38
31. 复选框: HeyCheckBox.....	40
32. 消息提示: HeyToast.....	41
33. 音量滑块: HeyControllerView.....	42
34. 加载对话框: HeyLoadingDialog.....	44
35. 抗锯齿圆角图片: HeyRoundImageAntiAlias.....	45
36. 通知栏图标: HeyToggleButton.....	46
37. 分组小标题: HeyGroupTitle.....	48
38. 回弹列表 RecyclerView/ListView/ScrollView.....	49
39. 右滑主题 1: HeytapTheme.....	51
40. 右滑主题 2: HeytapCompatTheme.....	52
41. 主题颜色.....	53

42. 字体大小.....	54
43. 常用动画.....	55
44. 常用主题.....	56
45. 常用距离.....	62
46. 抗锯齿优化.....	63
47. 点击态动效: <code>pressed_state_list</code>	64

1.主标题栏：HeyMainTitleBar

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyMainTitleBar
    android:id="@+id/default_titlebar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="@string/hey_default_title_text" />
```

使用说明：

标题文本通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制

2.返回标题栏：HeyBackTitleBar

效果图：



使用方法：

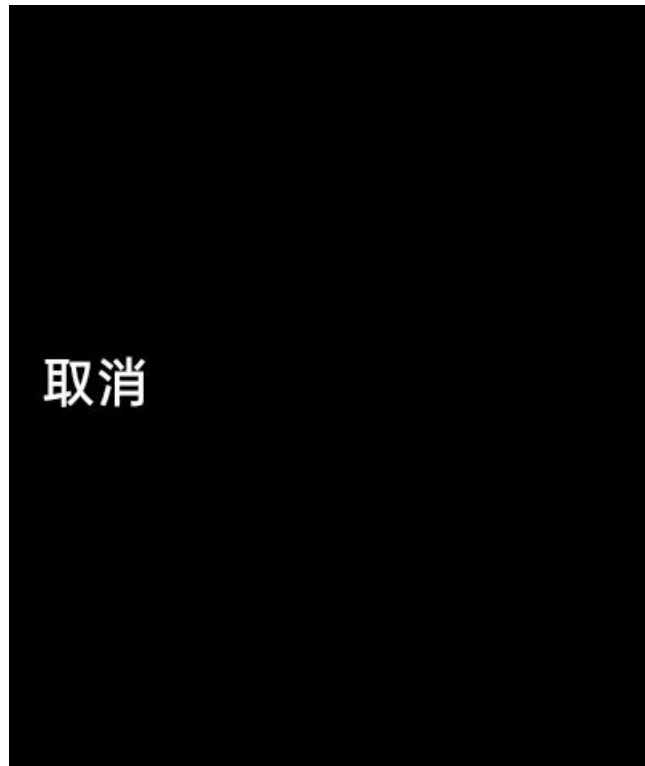
```
<com.heytap.wearable.support.widget.HeyBackTitleBar
    android:id="@+id/back_titlebar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="@string/hey_default_title_text" />
```

使用说明：

- 1.标题文本通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.返回操作可以通过方法 `setBackListener(null, activity)` 控制

3.取消标题栏：HeyCancelTitleBar

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyCancelTitleBar
    android:id="@+id/cancel_titlebar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="取消"/>
```

使用说明：

- 1.标题文本通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.返回操作可以通过方法 `setBackListener(null, activity)` 控制

4.应用标题栏：HeyAppTitleBar

效果图：



使用方法：

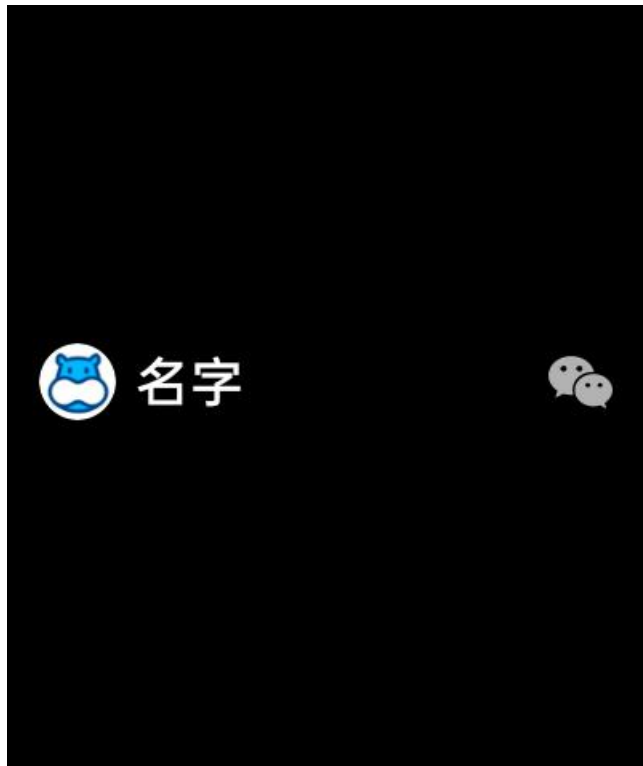
```
<com.heytao.wearable.support.widget.HeyAppTitleBar
    android:id="@+id/app_titlebar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="名字"
    app:heyTitleLeftImage="@drawable/msg" />
```

使用说明：

- 1.标题文本通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.左边图标可以通过属性 `app:heyTitleLeftImage` 控制，也可以通过 `getLeftIcon` 控制

5.通知标题栏：HeyAppTitleBar

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeyAppTitleBar
    android:id="@+id/noti_titlebar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="名字"
    app:heyTitleLeftImage="@drawable/hema"
    app:heyTitleRightImage="@drawable/wechat"
/>
```

使用说明：

- 1.标题文本通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.左边图标可以通过属性 `app:heyTitleLeftImage` 控制，也可以通过 `getLeftIcon` 控制
- 3.右边图标可以通过属性 `app:heyTitleRightImage` 控制，也可以通过 `getRightIcon` 控制

6.加载标题栏：HeyLoadingTitleBar

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyLoadingTitleBar
    android:id="@+id/loading_titlebar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="正在加载"/>
```

使用说明：

1.标题文本通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制

7.单行默认列表：HeySingleDefaultItem

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeySingleDefaultItem
    android:id="@+id/item"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyPrimaryImage="@drawable/hey_arrow"
    app:heyText="列表正文"/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.右边图标可以通过 `app:heyPrimaryImage` 控制，也可以通过 `getPrimaryImage` 控制

8.单行带开关列表：HeySingleItemWithSwitch

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeySingleItemWithSwitch
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.开关可以通过 `getHeySwitch` 获取

9.单行带复选框列表：HeySingleItemWithCheckBox

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeySingleItemWithCheckBox
    android:id="@+id/check_box"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.复选框可以通过 `getCheckBox` 获取

10.单行两个图标列表：HeySingleDefaultItem

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeySingleDefaultItem
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
    android:visibility="visible"
    app:heyShowMinorImage="true"
    app:heyMinorImage="@drawable/hey_lock"
    app:heyPrimaryImage="@drawable/hey_wifi"/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.右边靠右的图标可以通过 `app:heyPrimaryImage` 控制，也可以通过 `getPrimaryImage` 控制
- 3.右边靠左的图标可以通过 `app:heyMinorImage` 控制，也可以通过 `getMinorImage` 控制

11.单行带摘要列表：HeySingleItemWithSummary

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeySingleItemWithSummary
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
    app:heySummaryText="列表信息"
/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.右边摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `getSummaryTextView` 控制

12.单行带左边图标列表：HeySingleItemLeftImage

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeySingleItemLeftImage
    android:id="@+id/left_image"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
/>
```

使用说明：

- 1.列表正文通过属性 `getContentTextView` 控制
- 2.左边图片可以通过 `getLeftImageView` 控制

13.双行默认列表：HeyMultipleDefaultItem

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyMultipleDefaultItem
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
    app:heySummaryText="辅助文字"
    app:heyPrimaryImage="@drawable/hey_arrow"
/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.列表摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `getSummaryTextView` 控制
- 3.右边图标可以通过 `app:heyPrimaryImage` 控制，也可以通过 `getPrimaryImageView` 控制

14.双行带开关列表：HeyMultipleItemWithSwitch

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyMultipleItemWithSwitch
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
    app:heySummaryText="辅助文字"
/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.列表摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `getSummaryTextView` 控制
- 3.开关可以通过 `getHeySwitch` 获取

15.双行带复选框列表：HeyMultipleItemWithCheckBox

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeyMultipleItemWithCheckBox
    android:id="@+id/check_box"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
    app:heySummaryText="辅助文字"
/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.列表摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `getSummaryTextView` 控制
- 3.复选框可以通过 `getCheckBox` 获取

16.双行两个图标列表：HeyMultipleDefaultItem

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyMultipleDefaultItem
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
    app:heySummaryText="辅助文字"
    app:heyShowMinorImage="true"
    app:heyMinorImage="@drawable/hey_lock"
    app:heyPrimaryImage="@drawable/hey_wifi"/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `setTitle(title)` 控制
- 2.列表摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `getSummaryTextView` 控制
- 3.右边靠右的图标可以通过 `app:heyPrimaryImage` 控制，也可以通过 `getPrimaryImage` 控制
- 4.右边靠左的图标可以通过 `app:heyMinorImage` 控制，也可以通过 `getMinorImage` 控制

17.双行带左边图标列表：HeyMultipleItemWithLeftImage

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyMultipleItemWithLeftImage
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:heyText="列表正文"
    app:heySummaryText="辅助文字"
    app:leftImage="@drawable/msg"
    app:heyPrimaryImage="@drawable/hey_arrow"
/>
```

使用说明：

- 1.列表正文通过属性 `app:heyText` 控制，也可以通过方法 `getContentTextView` 控制
- 2.列表摘要文本可以通过 `app:heySummaryText` 控制，也可以通过 `getSummaryTextView` 控制
- 3.左边图标可以通过 `app:leftImage` 控制，也可以通过 `getLeftImageView` 控制
- 4.右边图标可以通过 `app:heyPrimaryImage` 控制，也可以通过 `getPrimaryImageView` 控制

18.圆角按钮：HeyShapeButton

大按钮

效果图：



双按钮

效果图：



使用方法:

```
<com.heytao.wearable.support.widget.HeyShapeButton
    android:id="@+id/sh"
    android:layout_width="185dp"
    android:layout_height="48dp"
    android:gravity="center"
    android:text="按钮"
```

```

        app:heyBackgroundColor="#33FFFFFF"
        app:heyCornerRadius="48dp"
        android:layout_marginBottom="5dp"/>

<com.heytap.wearable.support.widget.HeyShapeButton
    android:id="@+id/sh1"
    android:layout_width="185dp"
    android:layout_height="48dp"
    android:gravity="center"
    android:text="按钮"
    android:textColor="#FFFFFF"
    android:textSize="17sp"
    app:heyCornerRadius="48dp"
    android:layout_marginBottom="5dp"/>

<com.heytap.wearable.support.widget.HeyGradientButton
    android:layout_width="185dp"
    android:layout_height="48dp"
    android:layout_marginBottom="5dp"/>

```

使用说明：

此控件继承自 **Button**，具备 **Button** 的全部特性，同时新增了部分自定义属性；此控件与 **HeyButton** 类似，但是可以在 xml 中自定义背景颜色和圆角半径：

```

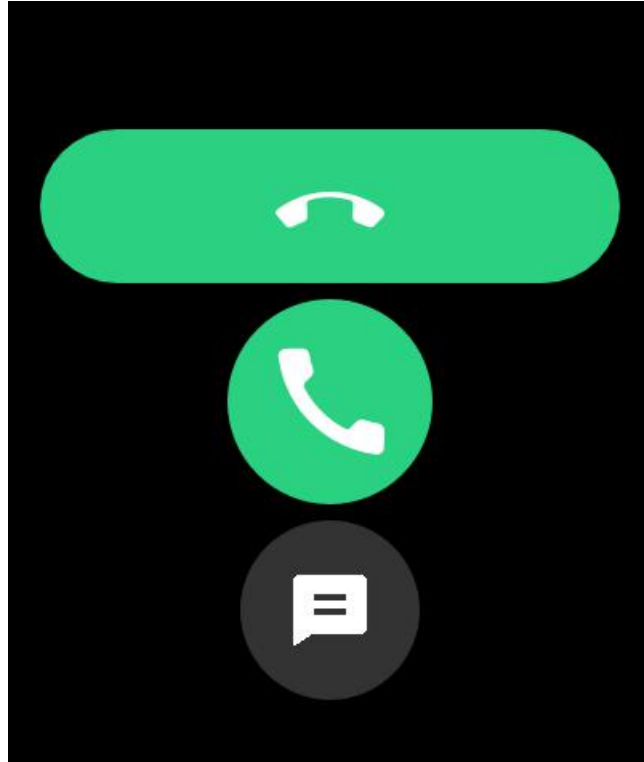
<declare-styleable name="HeyShapeButton">
    <!--背景颜色-->
    <attr name="heyBackgroundColor" format="color|reference"/>
    <!--圆角半径-->
    <attr name="heyCornerRadius" format="dimension|reference"/>
    <!--按钮类型-->
    <attr name="gradientButtonType" format="enum">
        <!--圆角-->
        <enum name="oval" value="1" />
        <!--圆形-->
        <enum name="circle" value="0" />
    </attr>
</declare-styleable>

```

新增属性 gradientButtonType，**oval** 代表圆角按钮；**circle** 代表圆形按钮，默认圆角按钮

19.图标按钮：HeyGradientButton

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeyGradientButton
    android:layout_marginTop="30dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    style="@style/HeyGradientButtonStyle" />

<com.heytao.wearable.support.widget.HeyGradientButton
    android:layout_marginTop="5dp"
    android:layout_width="64dp"
    android:layout_height="64dp"
    app:gradientButtonIconDrawable="@drawable/hey_answer"
    style="@style/HeyGradientButtonStyle"
/>

<com.heytao.wearable.support.widget.HeyGradientButton
    android:layout_marginTop="5dp"
    android:layout_width="56dp"
```

```
android:layout_height="56dp"
android:layout_marginBottom="5dp"
android:padding="10dp"
style="@style/HeyGradientButtonStyle"
app:gradientButtonIconDrawable="@drawable/hey_information"
app:gradientButtonBackground="@color/default_background_color"/>
```

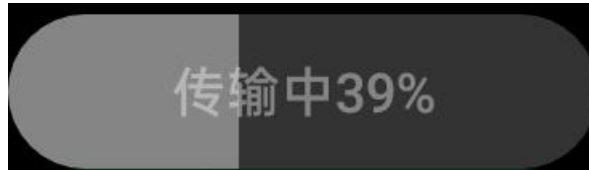
使用说明：

此控件继承自 `ImageButton`，新增了部分自定义属性：

```
<declare-styleable name="HeyGradientButton">
    <!--按钮图标-->
    <attr name="gradientButtonIconDrawable" format="reference" />
    <!--按钮背景颜色-->
    <attr name="gradientButtonBackground" format="color|reference"/>
    <!--按钮圆角半径-->
    <attr name="gradientButtonRadius" format="integer|reference"/>
</declare-styleable>
```

20.下载按钮：HeyProgressBar

效果图：



使用方法：

Xml

```
<com.heytap.wearable.support.widget.HeyProgressBar
    android:id="@+id/download_btn"
    android:layout_width="match_parent"
    android:layout_height="48dp"/>
```

Java

```
mHeyProgressBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        switch (mStateType) {
            case DownloadState.STATE_DEFAULT:
                mProgress = 0;
                mHeyProgressBtn.setProgress(mProgress);
                mStateType = DownloadState.STATE_DOWNLOADING;
                mHeyProgressBtn.setState(mStateType);
                mDownloadHandler.sendMessageDelayed(0, 500);
                break;

            case DownloadState.STATE_DOWNLOADING:
                mStateType = DownloadState.STATE_PAUSE;
                mHeyProgressBtn.setState(mStateType);
                mDownloadHandler.removeMessages(0);
                break;

            case DownloadState.STATE_PAUSE:
                mStateType = DownloadState.STATE_DOWNLOADING;
                mHeyProgressBtn.setState(mStateType);
                mDownloadHandler.sendMessageDelayed(0, 500);
                break;

            case DownloadState.STATE_DOWNLOAD_FINISH:
                mStateType = DownloadState.STATE_DEFAULT;
                mHeyProgressBtn.setState(mStateType);
```



```

        break;
    }
}
});

private static class DownloadHandler extends Handler {

    private WeakReference<Context> reference;

    DownloadHandler(Context context) {
        reference = new WeakReference<>(context);
    }

    @Override
    public void handleMessage(Message msg) {
        TestColorButton downloadActivity = (TestColorButton) reference.get();
        if (downloadActivity != null) {
            switch (msg.what) {
                case 0:
                    if (downloadActivity.mProgress < 100) {
                        downloadActivity.mProgress += 1.0;

downloadActivity.mHeyProgressBtn.setProgress(downloadActivity.mProgress);
                        downloadActivity.mDownloadHandler.sendMessageDelayed(0, 300);
                    } else {
                        downloadActivity.mStateType = DownloadState.STATE_DOWNLOAD_FINISH;

downloadActivity.mHeyProgressBtn.setState(downloadActivity.mStateType);
                    }
                    break;

                    default:
                        break;
            }
        }
    }
}
}

```

使用说明：

此按钮继承自 View，自定义了部分属性：

```

<declare-styleable name="HeyProgressButton">
    <!--按钮背景，layer-list 文件配置按钮圆角和颜色-->
    <attr name="progressButtonBackground" format="dimension"/>

```

```
    <!--字体大小-->
    <attr name="progressbtnTextSize" format="dimension"/>
    <!--默认状态字体颜色-->
    <attr name="progressbtnTextColor" format="color"/>
    <!--下载中字体颜色-->
    <attr name="progressbtnDownloadTextColor" format="color|reference" />
    <!--按钮文字-->
    <attr name="progressbtnText" format="string|reference" />
</declare-styleable>
```

21.单组数字选择器：HeyNumberPicker

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeyNumberPicker
    android:id="@+id/number_picker"
    android:layout_width="match_parent"
    android:layout_height="141dp"
    app:heyTextArray="@array/minute"
/>
```

使用说明：

heyTextArray 是时间选择器需要显示的内容，数组格式

22.上、下午数字选择器：HeyTimePicker

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeyTimePicker
    android:id="@+id/time_picker_2"
    android:layout_gravity="center"
    android:layout_width="match_parent"
    android:layout_height="141dp"/>
```

使用说明：

此控件继承自 `FrameLayout`，本质上是由 3 个 `HeyNumberPicker` 组合而成，并没有自定义任何属性

23.双组数字选择器：HeyTimePicker

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeyTimePicker
    android:id="@+id/time_picker_2"
    android:layout_gravity="center"
    android:layout_width="match_parent"
    android:layout_height="141dp"/>
```

使用说明：

此选择器与上下午选择器一模一样，与上、下午选择器相比，隐藏了上下午，当手表设置成 12 小时制时，上下午出现；当手表设置成 24 小时制时，上下午隐藏

24.时分秒选择器：HeyCounterPicker

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeyCounterPicker
    android:id="@+id/count_picker"
    android:layout_width="wrap_content"
    android:layout_height="141dp"
    android:layout_gravity="center"/>
```

使用说明：

此控件继承自 `FrameLayout`，本质上是由 3 个 `HeyNumberPicker` 和三个 `TextView` 组合而成，并没有自定义任何属性。

25.翻页符：HeyPageIndicator

效果图：



使用方法：

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:background="@drawable/bgg"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.viewpager.widget.ViewPager
        android:id="@+id/viewpager"
        android:background="@drawable/bgg"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />

    <com.heytao.wearable.support.widget.pageindicator.HeyPageIndicator
        android:id="@+id/indicator"
        android:layout_width="wrap_content"
        android:layout_height="24dp"
        android:layout_gravity="bottom|center"
        app:dotColor="@color/black"
        app:dotSize="6dp"
        app:dotSpacing="3dp"
        app:dotStrokeWidth="0.7dp"
        app:dotCornerRadius="3dp"
        app:traceDotColor="@color/white" />
</FrameLayout>
```

```

private void initView() {
    final ViewPager viewPager = findViewById(R.id.viewpager);
    final HeyPageIndicator indicator = findViewById(R.id.indicator);
    mBannerAdapter = new ImagePagerAdapter();
    viewPager.setAdapter(mBannerAdapter);
    viewPager.addOnPageChangeListener(new ViewPager.OnPageChangeListener() {
        @Override
        public void onPageScrolled(int position, float positionOffset, int positionOffsetPixels) {
            indicator.onPageScrolled(position, positionOffset, positionOffsetPixels);
        }

        @Override
        public void onPageSelected(int position) {
            indicator.onPageSelected(position);
        }

        @Override
        public void onPageScrollStateChanged(int state) {
            indicator.onPageScrollStateChanged(state);
        }
    });
    indicator.setDotsCount(mBannerAdapter.getCount());
    indicator.setOnDotClickListener(new HeyPageIndicator.OnIndicatorDotClickListener() {
        @Override
        public void onClick(int position) {
            viewPager.setCurrentItem(position);
        }
    });
}

```

使用说明：

继承自 **FrameLayout**，目前仅支持与 **ViewPager** 配合使用，不支持 **RecyclerView**，新增了部分自定义属性：

```

<declare-styleable name="HeyPageIndicator">
    <!--指示器的颜色，即上图下方后面两个小圆圈的颜色-->
    <attr name="dotColor" format="color"/>
    <!--过渡器颜色，即上图第一个圆圈的填充颜色-->
    <attr name="traceDotColor" format="color"/>
    <!--指示器大小，即上图中底部小圆圈的大小，指示器是由正方形设置圆角而来-->
    <attr name="dotSize" format="dimension"/>
    <!--指示器之间的距离，即上图中底部小圆圈之间的距离-->
    <attr name="dotSpacing" format="dimension"/>
    <!--指示器的描边宽度-->

```



```
<attr name="dotStrokeWidth" format="dimension"/>
<!--指示器圆角半径，默认是 dotSize/2-->
<attr name="dotCornerRadius" format="dimension"/>
<!--指示器是否可以点击，点击指示器具有切换界面的效果-->
<attr name="dotClickable" format="boolean"/>
</declare-styleable>
```

26.对话框(一个按钮): HeyDialog

效果图:



使用方法:

```
HeyDialog.HeyBuilder builder = new HeyDialog.HeyBuilder(this);
builder.setContentStyle(HeyDialog.STYLE_TITLE_CONTENT) //设置对话框内容样式, 含标题
    .setTitle("允许使用位置权限") //设置对话框标题
    .setMessage("轨迹等数据, 允许") //设置对话框内容
    .setPositiveButton("确定", null); //设置对话框按钮文本和点击事件
dialog = builder.create();
dialog.show();
```

使用说明:

HeyBtnDialog 继承自 Dialog, 具备 Dialog 的一切特性; 使用构建者模式创建, 与源生保持一致; `setContentStyle` 用于设置对话框内容样式, 控件包预置了 3 种样式, 本例的样式是对话框包含标题的样式。如果对话框按钮点击事件不需要额外的逻辑, 那么 `setPositiveButton` 方法的第二个参数传入 `null`, 点击后对话框会自动消失。

27.对话框(两个按钮): HeyDialog

效果图:



使用方法:

```
HeyDialog.HeyBuilder builder = new HeyDialog.HeyBuilder(this);
builder.setContentViewStyle(HeyDialog.STYLE_CONTENT)    //设置对话框样式，不含标题
    .setMessage(" 信息权限记录运动轨迹等数据，是否允? ")
    .setButtonOrientation(LinearLayout.HORIZONTAL) //设置按钮方向
    .setNegativeButton("按钮", null)
    .setPositiveButton("按钮", new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Toast.makeText(SecActivity.this, "dialog", Toast.LENGTH_LONG).show();
        }
    });
HeyDialog dialog = builder.create();
```

使用说明:

与一个按钮对话框相比，两个按钮对话框需要调用方法 `setButtonOrientation` 设置按钮方向，类型是 `LinearLayout.HORIZONTAL` 和 `Vertical`。另外，对于水平布局的按钮来说，右边按钮是 `PositiveButton`；对于垂直布局的按钮来说，下面按钮是 `PositiveButton`。

28.对话框(权限隐私申明): HeyDialog

效果图:



使用方法:

```
HeyDialog.HeyBuilder builder = new HeyDialog.HeyBuilder(this);
    builder.setContentVisualStyle(HeyDialog.STYLE_PROTOCOL)
        .setTitle("应用名称声明")
        .setMessage("“应用名称”是用于具体用途的软件。为了实现上述目的，我们在向您提供服务时，可能需要使用您的相关信息。OPPO旨在保护你的个人信息，我们承诺你的个人信息仅会用于我们所声明的目的。详情请查看应用名称用户协议、应用名称隐私政策。")
        .setButtonOrientation(LinearLayout.HORIZONTAL)
        .setSummary("点击“同意”即表示同意上述内容")
        .setNegativeButton("按钮", null)
        .setPositiveButton("按钮", new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Toast.makeText(SecActivity.this, "隐私申明",
                    Toast.LENGTH_LONG).show();
            }
        });
    HeyDialog dialog = builder.create();
```

使用说明: 同上

29.对话框(自定义布局): HeyDialog

效果图:



使用方法:

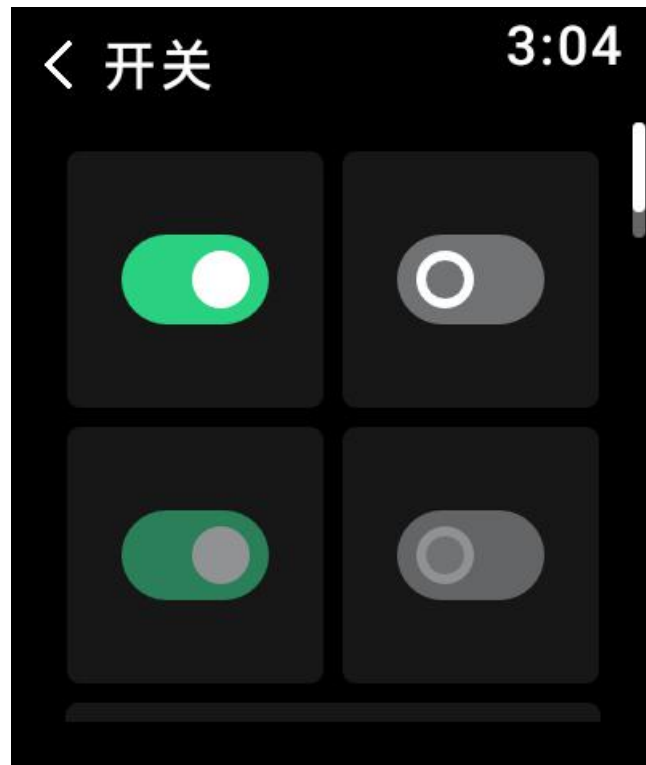
```
HeyDialog.HeyBuilder builder = new HeyDialog.HeyBuilder(this);
builder.setContentView(R.layout.custom_layout) //设置自定义布局
    .setPositiveButton("按钮", null);
HeyDialog dialog = builder.create();
dialog.show();
```

使用说明:

鉴于对话框样式千变万化,所以提供了自定义布局的对话框,使用方法如上所示,只需要传入自己的布局就行,如果所需对话框不在上面列出的样式当中,请通过自定义布局实现,控件包不再单独提供别的对话框

30.开关：HeySwitch

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeySwitch
    android:id="@+id/color_switch1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:clickable="true"
    app:barCheckedColor="#FF11C94C"
    app:barHeight="28dp"
    app:barWidth="46dp"
    app:circlePadding="6dp"
    app:outerCircleStrokeWidth="3dp"
    app:outerCircleWidth="18dp" />
```

使用说明：

此控件继承自 **CompoundButton**，新增了部分自定义属性：

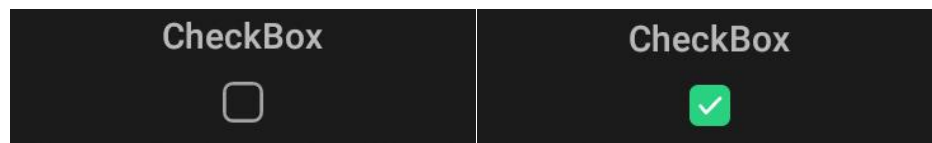
```
<declare-styleable name="HeySwitch">
    <!--开关高度-->
    <attr name="barHeight" format="dimension" />
    <!--开关宽度-->
    <attr name="barWidth" format="dimension" />
```

```
<!--圆点半径-->
<attr name="outerCircleWidth" format="dimension" />
<!--关闭时，中间灰圆与外面白圆之间的距离-->
<attr name="outerCircleStrokeWidth" format="dimension" />
<!--关闭时，中间灰圆外面圆之间的颜色/打开时外圆的颜色-->
<attr name="outerCircleColor" format="dimension" />
<!--关闭时内圆半径-->
<attr name="innerCircleWidth" format="dimension" />
<!--内圆内边距-->
<attr name="circlePadding" format="dimension" />
<attr name="loadingDrawable" format="reference" />
<attr name="barCheckedColor" format="color" />
<attr name="barUncheckedColor" format="color" />
<attr name="innerCircleColor" format="color" />

<!--内圆内边距-->
<attr name="barUncheckedDisabledColor" format="color" />
<attr name="barCheckedDisabledColor" format="color" />
<attr name="outerCircleUncheckedDisabledColor" format="color" />
</declare-styleable>
```

31.复选框：HeyCheckBox

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyCheckBox
    android:id="@+id/oppo_check_box"
    android:layout_gravity="center_horizontal"
    android:layout_width="24dp"
    android:layout_height="24dp"
    android:clickable="true"
    android:focusable="true" />
```

使用说明：

HeyCheckBox,继承自 Button, 新增了部分自定义属性：

```
<declare-styleable name="HeyCheckBox">
    <!--CheckBox 的颜色，包括选中和不选中等-->
    <attr name="HeyCheckBoxStateList" format="reference" />
    <!--CheckBox 是否全选-->
    <attr name="HeyCheckboxStateAllSelect" format="boolean" />
    <!--CheckBox 是否部分选择-->
    <attr name="HeyCheckboxStatePartSelect" format="boolean" />
    <!--CheckBox 大小-->
    <attr name="HeyCheckboxSize" format="dimension" />
    <!--CheckBox 状态(全选，全不选，部分选中)-->
    <attr name="HeyCheckBoxState">
        <enum name="unSelected" value="0" />
        <enum name="partSelected" value="1" />
        <enum name="allSelected" value="2" />
    </attr>
</declare-styleable>
```


32.消息提示：HeyToast

效果图：



使用方法：

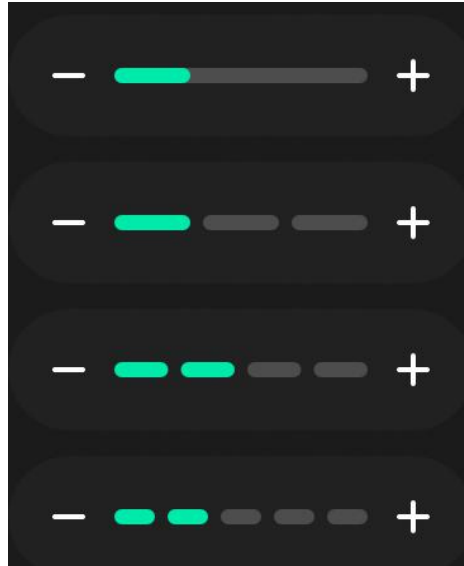
```
HeyToast.showToast(this, "Toast 文字，居中示意超长折行。", Toast.LENGTH_SHORT);
```

使用说明：

HeyToast 继承自 Toast，在 showToast 里面 inflater 了自定义布局，此布局对 Toast 的 TextView 进行了部分属性设置；没有新增自定义属性。

33.音量滑块： HeyControllerView

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeyControllerView
    android:id="@+id/colorControllerView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:progressBarColor="#FF00EAAB"/>

<com.heytao.wearable.support.widget.HeyControllerView
    android:id="@+id/colorControllerView2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    app:curProgress="1"
    app:maxProgress="3"
    app:progressBarType="segment"
    app:progressBarColor="#FF00EAAB"/>

<com.heytao.wearable.support.widget.HeyControllerView
    android:id="@+id/colorControllerView3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    app:curProgress="2"
    app:maxProgress="4"
    app:progressBarType="segment" />
```

```

<com.heytap.wearable.support.widget.HeyControllerView
    android:id="@+id/colorControllerView4"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    app:curProgress="2"
    app:maxProgress="5"
    app:progressBarType="segment" />

```

使用说明：

此控件主要用于设置模块的音量控制，继承自 **FrameLayout**，左右的"-"、"+"可以减少、增加音量，封装了 **HeyHorizontalProgressBar**，具体实现是左右两个 **ImageView** 加上中间的 **HeyHorizontalProgressBar** 构成，此控件自定义了属性：

```

<declare-styleable name="HeyHorizontalProgressBar">
    <!--此属性是指音量控制器中进度条的背景颜色，如上图进度条的背景颜色是灰色-->
    <attr name="backgroundBarColor" format="color|reference"/>
    <!--此属性是指音量控制器中进度条的前景颜色，如上图 2、3、4 中进度条的前景颜色是原谅色-->
    <attr name="progressBarColor" format="color|reference"/>
    <!--此属性当且仅当音量控制器的 progressBarType 为 segment 时才有效；代表音量控制器中有几个小刻度-->
    <attr name="maxProgress" format="integer|reference"/>
    <!--此属性当且仅当音量控制器的 progressBarType 为 segment 时才有效；代表音量控制器中的当前进度-->
    <attr name="curProgress" format="integer|reference"/>
    <!--音量控制器的类型，离散型和连续型，默认类型是连续型-->
    <attr name="progressBarType">
        <enum name="normal" value="0" />
        <enum name="segment" value="1" />
    </attr>
</declare-styleable>

```

34.加载对话框：HeyLoadingDialog

效果图：



使用方法：

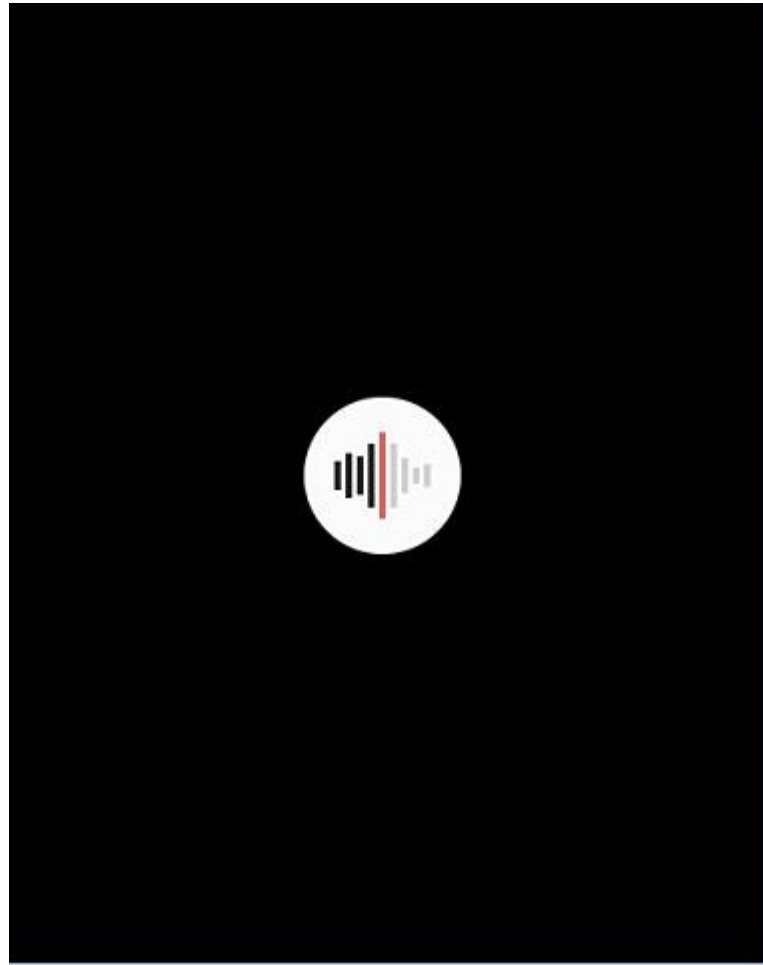
```
public void showLoadingDialog(View view) {  
    final HeyLoadingDialog colorLoadingDialog = new HeyLoadingDialog(this, "加载中，请稍后");  
    colorLoadingDialog.show();  
    new Handler().postDelayed(new Runnable() {  
        @Override  
        public void run() {  
            colorLoadingDialog.dismiss();  
        }  
    }, 15000);  
}
```

使用说明：

此控件继承自 Dialog，没有自定义属性

35.抗锯齿圆角图片：HeyRoundImageAntiAlias

效果图：



使用方法：

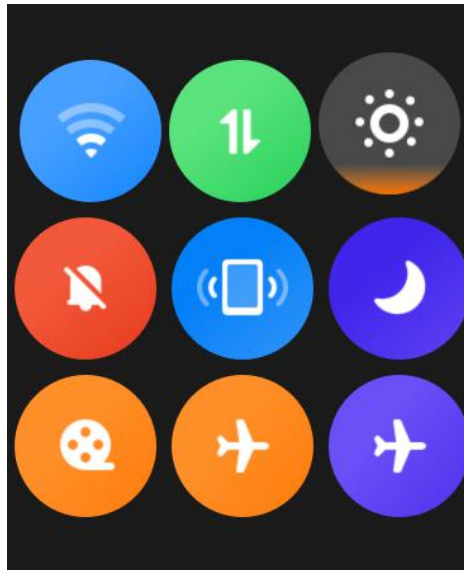
```
<com.heytao.wearable.support.widget.HeyRoundImageAntiAlias
    android:layout_width="50dp"
    android:layout_height="50dp"
    android:gravity="center"
    android:src="@drawable/record" />
```

使用说明：

此控件继承自 `ImageView`，具备 `ImageView` 的全部特性，没有新增自定义属性，直接当做普通 `ImageView` 使用即可。

36.通知栏图标： HeyToggleButton

效果图：



使用方法：

```
<com.heytao.wearable.support.widget.HeyToggleButton
    android:id="@+id/wifi"
    android:layout_width="56dp"
    android:layout_height="56dp"
    android:checked="true"
    android:clickable="true"
    android:enabled="true"
    app:iconDrawable="@drawable/wifi"
    app:onColorAngle="-40"
    app:onColorEndColor="#FF47A0FF"
    app:onColorEndPercent="99"
    app:onColorStartColor="#FF007AFF"
    app:onColorStartPercent="2" />
```

使用说明：

此控件主要用于通知栏快捷开关，具备颜色渐变效果，继承自 CompoundButton，具有循环点击的功能，此控件新增了部分自定义属性：

```
<resources>
    <declare-styleable name="HeyToggleButton">
        <!--开关关闭后按钮的背景色-->
        <attr name="offColor" format="color" />
        <!--开关打开后按钮的背景色，因为目前的需求是开关的背景色要渐变，所有此值无需设置，只需要设置渐变的开始颜色和结束颜色即可。-->
        <attr name="onColor" format="color" />
        <!--按钮上的图片-->
```

```
<attr name="iconDrawable" format="reference" />
<!--开关中图片的内边距-->
<attr name="iconPadding" format="integer" />
<!--渐变角度-->
<attr name="onColorAngle" format="integer" />
<!--渐变开始颜色-->
<attr name="onColorStartColor" format="color" />
<!--渐变开始百分比-->
<attr name="onColorStartPercent" format="integer" />
<!--渐变结束颜色-->
<attr name="onColorEndColor" format="color" />
<!--渐变结束百分比-->
<attr name="onColorEndPercent" format="integer" />
</declare-styleable>
```

37.分组小标题： HeyGroupTitle

效果图：



使用方法：

```
<com.heytap.wearable.support.widget.HeyGroupTitle
    android:id="@+id/title1"
    style="@style/HeyGroupTitleStyle"
    android:layout_width="match_parent"
    app:groupTitleText="分组小标题"
    app:groupTitleShowLoading="true"/>

<com.heytap.wearable.support.widget.HeyGroupTitle
    android:layout_width="match_parent"
    style="@style/HeyGroupTitleStyle"
    app:groupTitleText="分组小标题"/>

<com.heytap.wearable.support.widget.HeyGroupTitle
    android:layout_width="match_parent"
    style="@style/HeyGroupTitleStyle"
    app:groupTitleText="分组小标题"
    app:groupTitlePosition="middle"/>
```


38.回弹列表 RecyclerView/ListView/ScrollView

1.RecyclerView

效果图：



使用方法：

```
<<com.heytao.wearable.support.recyclerview.widget.RecyclerView
    android:id="@+id/recycler"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:fadeScrollbars="false"
    android:scrollbars="vertical" />
```

```
import com.heytao.wearable.support.recyclerview.widget.LinearLayoutManager;
import com.heytao.wearable.support.recyclerview.widget.RecyclerView;
mRecyclerView = findViewById(R.id.recycler);
mRecyclerView.setLayoutManager(new LinearLayoutManager(this));
```

使用说明：

此控件是在原生 RecyclerView 的基础上修改的，除了包名不一样，其他使用方法和原生 RecyclerView 一模一样

ListView/ScrollView:

ListView 和 ScrollView 均新增了是否支持回弹的接口 `setEnableOverScroll(boolean b)`, `false` 表示不支持回弹, `true` 表示支持回弹, 默认为 `false`, 即默认不支持回弹, 如需支持, 可以通过反射调用 `setEnableOverScroll` 设置

39.右滑主题 1：HeytapTheme

效果图：

无

使用方法：

a.继承自此主题

在 styles.xml 中：

```
<style name="AppTheme" parent="@style/HeytapTheme">
    <!-- Customize your theme here. -->
    ....
</style>
```

b.直接使用此主题

在 AndroidManifest.xml 中：

```
<application
    .....
    android:theme="@style/HeytapTheme">
```

使用说明：

新增此主题是为了适配字体，所有用到右滑退出主题的应用，**务必**均接入此主题
界面直接继承自 Activity 的应用请使用此主题

40.右滑主题 2: HeytapCompatTheme

效果图:

无

使用方法:

c.继承自此主题

在 styles.xml 中:

```
<style name="AppTheme" parent="@style/HeytapCompatTheme">
    <!-- Customize your theme here. -->
    ....
</style>
```

d.直接使用此主题

在 AndroidManifest.xml 中:

```
<application
    .....
    android:theme="@style/HeytapCompatTheme">
```

使用说明:

新增此主题是为了适配字体，所有用到右滑退出主题的应用，**务必**均接入此主题

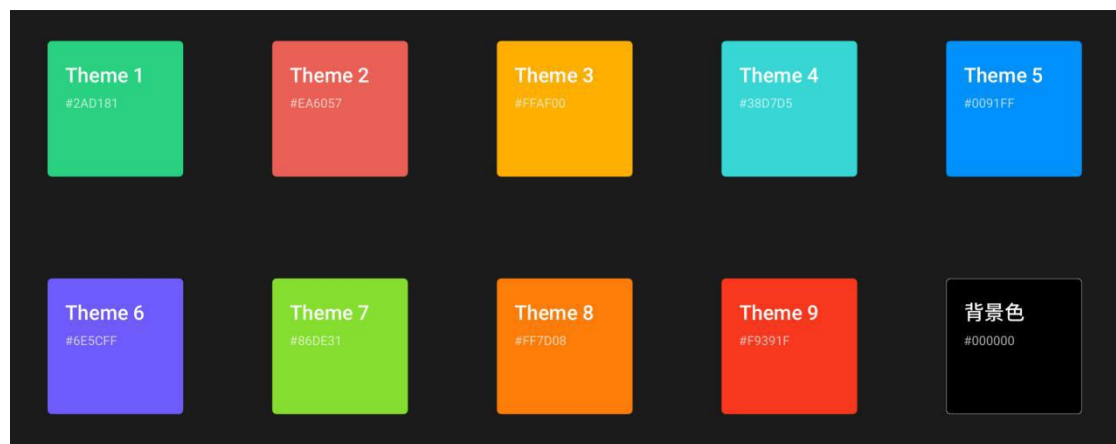
界面继承自 AppCompatActivity 的应用请使用此主题

41.主题颜色

根据设计要求，控件包集成 9 种主题颜色，各个应用的主色调从这些颜色中选取，颜色如下：

```
<color name="theme1">#FF2AD181</color>
<color name="theme1_light">#4D2AD181</color>
<color name="theme2">#FFEA6075</color>
<color name="theme2_light">#4DEA6075</color>
<color name="theme3">#FFFFAF00</color>
<color name="theme3_light">#4DFFAF00</color>
<color name="theme4">#FF38D7D5</color>
<color name="theme4_light">#4D38D7D5</color>
<color name="theme5">#FF0091FF</color>
<color name="theme5_light">#4D0091FF</color>
<color name="theme6">#FF6E5CFF</color>
<color name="theme6_light">#4D6E5CFF</color>
<color name="theme7">#FF86DE31</color>
<color name="theme7_light">#4D86DE31</color>
<color name="theme8">#FFF7D08</color>
<color name="theme8_light">#4DFF7D08</color>
<color name="theme9">#FFF9391F</color>
<color name="theme9_light">#4DF9391F</color>
```

效果如下：



42.字体大小

为适配大小表字体效果，控件包新增以下字体大小，各应用务必引用下面的字体大小，否则小表显示效果会不对：

```
<dimen name="xxxl_body">101sp</dimen>
<dimen name="xxl_body">84sp</dimen>
<dimen name="xl_body">69sp</dimen>
<dimen name="l_body">57sp</dimen>
<dimen name="m_body">46sp</dimen>
<dimen name="s_body">38sp</dimen>
<dimen name="xl_title">31sp</dimen>
<dimen name="l_title">26sp</dimen>
<dimen name="m_title">21sp</dimen>
<dimen name="s_title">17sp</dimen>
<dimen name="l_description">17sp</dimen>
<dimen name="m_description">14sp</dimen>
<dimen name="s_description">12sp</dimen>
<dimen name="caption">9sp</dimen>
```

43.常用动画

为方便应用开发，控件包特意预置部分公用动画：

1.dialog_enter_from_bottom，此动画是从底部弹出动画，常用于对话框

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:zAdjustment="top">
    <translate
        android:fromYDelta="100%p"
        android:toYDelta="0%p"
        android:duration="133"/>
</set>
```

2.dialog_exit_from_top，此动画是从顶部退出动画，常用于对话框

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:zAdjustment="top">
    <translate
        android:fromYDelta="0%p"
        android:toYDelta="100%p"
        android:duration="133"/>
</set>
```

44.常用主题

为统一显示效果，同时减轻各应用工作量，特预置以下常用主题：

1.标题栏默认样式：HeyTitleBarStyle

使用方法：

```
<com.heytap.wearable.support.widget.HeyTitleBar
    style="@style/HeyTitleBarStyle"
    android:id="@+id/back"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:leftText="标题栏"/>
```

使用效果：



2.单行列表默认样式：HeyListSingleItemNormalStyle

使用方法：

```
<com.heytap.wearable.support.widget.HeyListItemNormal
    style="@style/HeyListSingleItemNormalStyle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:itemContentText="单行列表"/>
```

使用效果：



3. 双行列表默认样式: HeyListMultipleItemNormalStyle

使用方法:

```
<com.heytao.wearable.support.widget.HeyListItemNormal  
    style="@style/HeyListMultipleItemNormalStyle"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    app:itemContentText="双行列表"  
    app:itemSummaryText="辅助文字"/>
```

使用效果:

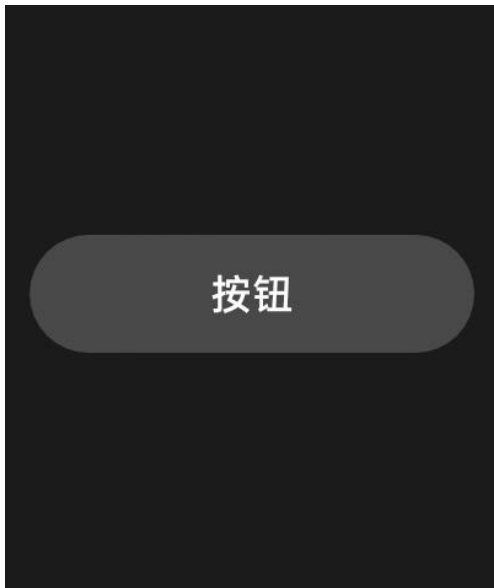


4.大按钮默认样式：HeyBigShapeButtonStyle

使用方法：

```
<com.heytap.wearable.support.widget.HeyShapeButton  
    style="@style/HeyBigShapeButtonStyle"  
    android:layout_width="match_parent"/>
```

使用效果：

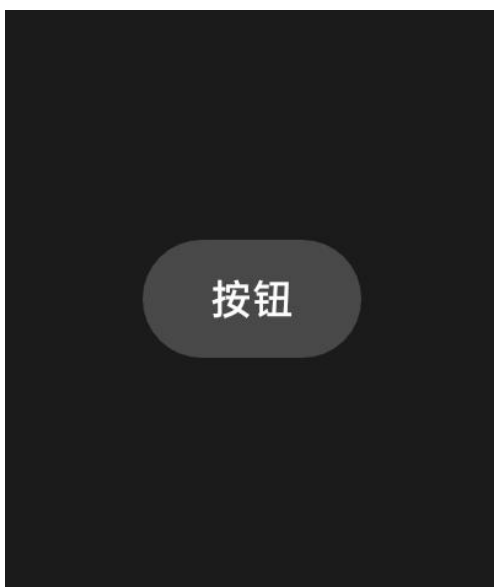


5.小按钮默认样式：HeySmallShapeButtonStyle

使用方法：

```
<com.heytap.wearable.support.widget.HeyShapeButton  
    style="@style/HeySmallShapeButtonStyle"/>
```

使用效果：

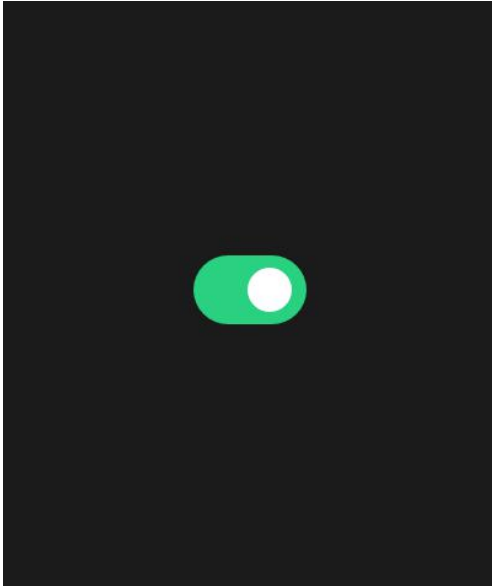


6.开关默认样式: HeySwitchStyle

使用方法:

```
<com.heytap.wearable.support.widget.HeySwitch  
    style="@style/HeySwitchStyle"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

使用效果:



7.渐变默认样式: HeyGradientButtonStyle

使用方法:

```
<com.heytap.wearable.support.widget.HeyGradientButton  
    style="@style/HeyGradientButtonStyle"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

使用效果:

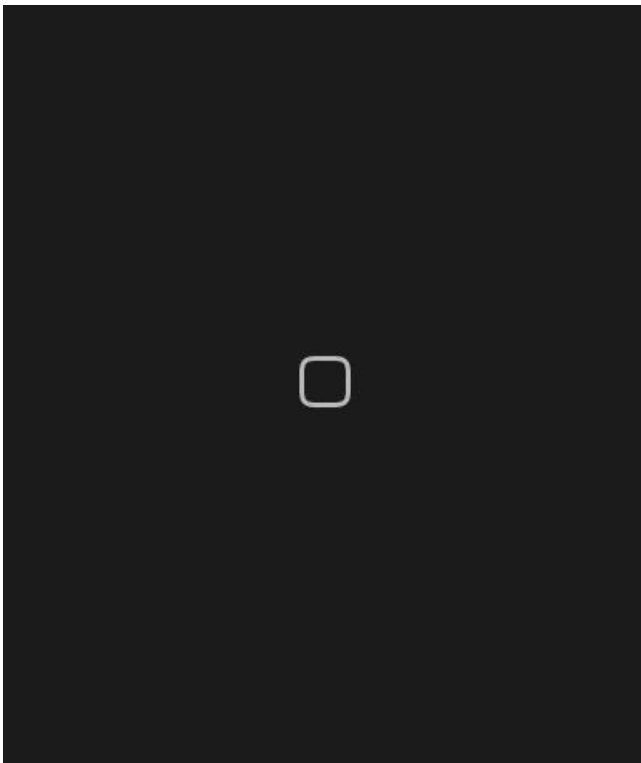


8.单选框默认样式：HeyCheckBoxStyle

使用方法：

```
<com.heytao.wearable.support.widget.HeyCheckBox  
    style="@style/HeyCheckBoxStyle"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"/>
```

使用效果：



9.分组小标题默认样式：HeyGroupTitleStyle

使用方法：

```
<com.heytao.wearable.support.widget.HeyGroupTitle
    android:layout_width="match_parent"
    style="@style/HeyGroupTitleStyle"
    app:groupTitleText="分组小标题"/>
```

使用效果：



45.常用距离

为方便统一控制距离，控件包集成部分常用的宽高及边距，代码如下(有集成需求的请告知我，我集成进去)：

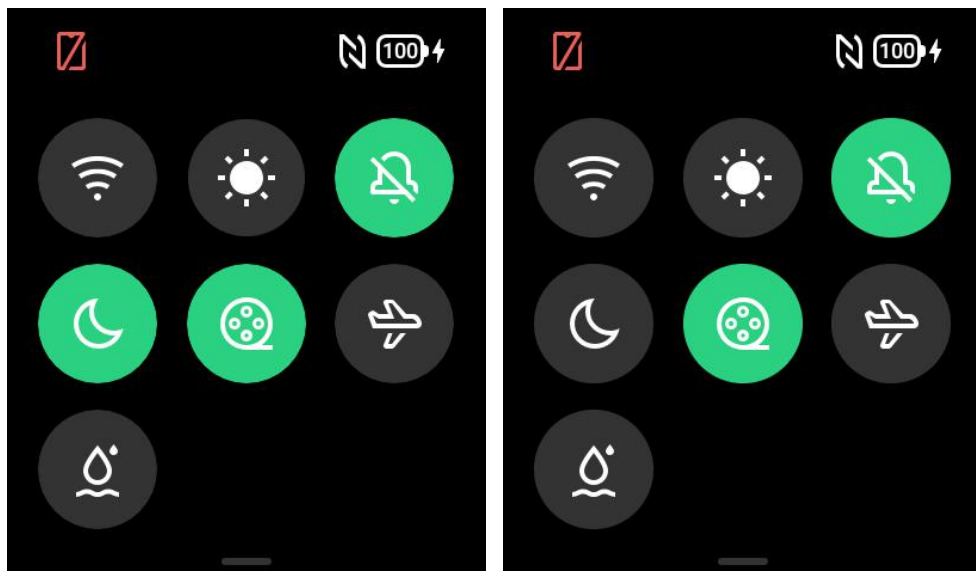
```
<!--正文距离屏幕左右边距-->
<dimen name="hey_content_horizontal_distance">10dp</dimen>
<!--默认正文文字大小-->
<dimen name="hey_default_textSize">17sp</dimen>
<!--默认摘要文字大小-->
<dimen name="group_title_text_size">14sp</dimen>
<!--单行列表默认高度-->
<dimen name="hey_single_item_height">48dp</dimen>
<!--多行列表默认高度-->
<dimen name="hey_multiple_item_height">56dp</dimen>
<!--分组小标题默认高度-->
<dimen name="hey_group_title_height">30dp</dimen>
<!--按钮默认高度-->
<dimen name="hey_button_height">48dp</dimen>
<!--标题栏默认高度-->
<dimen name="hey_title_bar_height">36dp</dimen>
```

46.抗锯齿优化

鉴于当前手表大量使用圆角和圆形图片，这些图片在大小表上表现均有不同程度的锯齿问题，对用户体验影响较为明显，为解决此问题，特整理此方案，便于各应用在以后的开发中避免此问题。

```
myView.setOutlineProvider(new ViewOutlineProvider() {  
    @Override  
    public void getOutline(View view, Outline outline) {  
        outline.setOval(0, 0, getWidth(), getHeight());  
    }  
});  
myView.setClipToOutline(true);
```

此方法是 `View.java` 中定义的，所以任何继承自 `view` 类的控件都可以使用；另外，如果只需要圆角效果，可以调用 `setRoundRect` 方法。如果自定义控件没有继承自 `view` 或者控件内没包含 `view`，那么此方案将无效，效果对比：



47.点击态动效：pressed_state_list

效果图：

无

使用方法：

a.Java 代码中使用：

```
StateListAnimator animator = AnimatorInflater.loadStateListAnimator(context,
R.animator.hey_press_state_list);
yourview.setStateListAnimator(animator);
```

b.xml 中使用：

```
android:stateListAnimator="@animator/pressed_state_list"
```

使用说明：

当前公用的点击态效果是控件大小缩小至 90%，透明度减少 50%，如有特殊动效需求，可以自定义，自定义方法：

首先，在 res/animator 中新建 xml 文件，如 pressed_state_list

其次，在该 xml 中定义动画，例如：

```
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:state_pressed="true">
        <set>
            <objectAnimator android:duration="66" android:propertyName="scaleX"
android:valueTo="0.9" android:valueType="floatType" />
            <objectAnimator android:duration="66" android:propertyName="scaleY"
android:valueTo="0.9" android:valueType="floatType" />
            <objectAnimator android:duration="66" android:propertyName="pivotX"
android:valueTo="24" android:valueType="intType" />
            <objectAnimator android:duration="66" android:propertyName="pivotY"
android:valueTo="24" android:valueType="intType" />
            <objectAnimator android:duration="66" android:propertyName="Alpha"
android:valueTo="0.5" android:valueType="floatType" />
        </set>
    </item>

    <item android:state_pressed="false">
        <set>
            <objectAnimator android:duration="333" android:propertyName="scaleX"
android:valueTo="1.0" android:valueType="floatType" />
            <objectAnimator android:duration="333" android:propertyName="scaleY"
android:valueTo="1.0" android:valueType="floatType" />
            <objectAnimator android:duration="333" android:propertyName="pivotX"
android:valueTo="24" android:valueType="intType" />
        </set>
    </item>
</selector>
```



```
        <objectAnimator android:duration="333" android:propertyName="pivotY"
android:valueTo="24" android:valueType="intType" />
        <objectAnimator android:duration="333" android:propertyName="Alpha"
android:valueTo="1.0" android:valueType="floatType" />
    </set>
</item>
</selector>
```