

Android样式和主题

共同点：定义的方式一样

不同点：作用范围不一样

主题给Application和 Activity设置，作用范围比较大

样式给TextView、Button设置，作用范围比较小

如果很多控件有相同的属性特点，比如说TextView的字体颜色，字体大小等属性一样，那么我们可以将这些相同的属性抽取出来变成Style样式。

在res目录下的values目录中的styles.xml文件中自定义，我们自己的样式或主题

应用程序国际化

应用程序国际化，通常简称I18n。国际化的目的就是为了让开发的应用尽可能被更多的用户使用。

在res目录下创建每个国家国际化对应的文件夹，以英文为例。然后为每个string设置不同的语言。

同一应用安装在不同国家或地区的手机上其应用的文字图片会有相应的显示。



帧动画

多张图片组成的动画

1、在\res\drawable目圈下放入图多张图片并写好一个XML文件用来记录图片的信息

```
<?xml version="1.0" encoding="utf-8"?>

<!-- android:oneshot表示是否是播放一次帧动画,true表示只完整播放一次 ; false表示可以完整播放N次-->
<animation-list xmlns:android="http://schemas.android.com/apk/res/android" android:oneshot="false">

    <!-- android:drawable引入图片
         | android:duration图片的播放时长, 单位毫秒
    -->
    <item android:drawable="@drawable/girl_01" android:duration="200" />
    <item android:drawable="@drawable/girl_02" android:duration="200" />
</animation-list>
```

2、布局中创建一个ImageView组件

3、代码中将UI组件及图片信息的XML调入开始

```

//子线程
new Thread() {
    public void run() {
        // 将frame.xml文件设置到ImageView控件中背景中
        iv.setBackgroundResource(R.drawable.frame);
        // 休息
        SystemClock.sleep(3000);
        // 获取ImageView控件的背景，即帧(子线程)
        final AnimationDrawable ad = (AnimationDrawable) iv.getBackground();

        MainActivity.this.runOnUiThread(new Thread() {
            public void run() {
                // 播放帧动画(主线程或UI线程)
                ad.start();
            }
        });
    }
}

```

补间动画代码版

其实际就是一张图片的透明，旋转，缩放，位移，合集

- 1、布局文件中放入一个ImageView组件，并为组件设置一个图片
- 2、放入5个按钮（按钮点击触发动画）
- 3、代码完善动画效果

透明：AlphaAnimation

- 1、创建AlphaAnimation对象 参数一：从什么透明度, 0.0F透明 参数二：到什么透明度, 1.0F不透明
AlphaAnimation aa = new AlphaAnimation(1F, 0F);

- 2、设置画动播放的时长，单位毫秒

aa.setDuration(200);

- 3、在默认情况下，动画只播放1次，让其再播放1次，即共2次

aa.setRepeatCount(1); setRepeatCount(Animation.INFINITE); 无限循环

- 4、反向播放

aa.setRepeatMode(AlphaAnimation.REVERSE);

- 5、动画播放完毕后，就停止当前动画状态

aa.setFillAfter(true);

- 6、让ImageView来播放这个AlphaAnimation动画

iv.startAnimation(aa);

旋转：RotateAnimation

在默认情况下，饶ImageView的原点，是左上角

Animation.RELATIVE_TO_SELF, 0.5F, 表法相对自己(ImageView), 0.5F表示自己宽x度的一半

Animation.RELATIVE_TO_SELF, 0.5F, 表法相对自己(ImageView), 0.5F表示自己长y度的一半

旋转时，以自己中心为基准点，绕三圈

RotateAnimation ra = new RotateAnimation(0F, 360*3F, Animation.RELATIVE_TO_SELF , 0.5F
 , Animation.RELATIVE_TO_SELF , 0.5F);

缩放：ScaleAnimation

1F表示X不变，3F表示是X的3位，相对自己

ScaleAnimation sa = new ScaleAnimation(0.1F, 3F, 0.1F, 3F, Animation.RELATIVE_TO_SELF, 0.5F,
 Animation.RELATIVE_TO_SELF, 0.5F);

位移：TranslateAnimation

获取图片的高或者宽

```
int ivHeight = iv.getHeight();
```

```
int ivWidth = iv.getWidth();
```

以X轴移动自己的宽度这么远，向右平移 Y轴不变

```
TranslateAnimation ta = new TranslateAnimation(0, ivWidth, 0, 0);
```

动画集合：AnimationSet

1、先写任意几个动画，但不要让其播放

2、创建动画集合, false表示旋转时带着缩放

```
AnimationSet set = new AnimationSet(false);
```

3、将旋转动画和缩放动画加入到动画集合中

```
set.addAnimation(ra);
```

```
set.addAnimation(sa);
```

4、让View对象去播放动画集合

```
iv.startAnimation(set);
```

补间动画XML版本：

<!-- 透明动画 在res目录创建一个anim目录 -->

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<alpha
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="200"
    android:fillAfter="true"
    android:fromAlpha="0"
    android:repeatCount="infinite"
    android:repeatMode="restart"
    android:toAlpha="1" >
```

```
</alpha>
```

<!-- rotate>是旋转动 -->

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<rotate
```

```
xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="3000"
    android:fromDegrees="0"
    android:pivotX="50%"
    android:pivotY="50%"
    android:toDegrees="360"
    android:repeatCount="infinite"
    android:repeatMode="reverse" >
```

```
</rotate>
```

<!-- scale是缩放动画 -->

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<scale
```

```

xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="600"
    android:fillAfter="true"
    android:fromXScale="3"
    android:fromYScale="3"
    android:repeatCount="infinite"
    android:repeatMode="reverse"
    android:toXScale="1"
    android:toYScale="1" >
</scale>

<!-- translate是位移动画 -->
<?xml version="1.0" encoding="utf-8"?>
<translate
xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="200"
    android:fillAfter="true"
    android:fromXDelta="30"
    android:fromYDelta="10"
    android:repeatCount="infinite"
    android:repeatMode="reverse"
    android:toXDelta="10"
    android:toYDelta="50" >
</translate>

```

属性动画

1 位移

```

ObjectAnimator animator = ObjectAnimator.ofFloat(mIm, "translationY", 0,
120, 0, 0, 120);
//动画总时间
animator.setDuration(10000);
//开始动画
animator.start();

```

2 透明度

```

ObjectAnimator animator = ObjectAnimator.ofFloat(mIm, "alpha", 1, 0, 0, 1);
animator.setDuration(8000);
无限循环
animator.setRepeatCount(ObjectAnimator.INFINITE);
animator.setRepeatMode(ObjectAnimator.REVERSE);
animator.start();

```

3 缩放

```
ObjectAnimator animator = ObjectAnimator.ofFloat(mIm, "scaleX", 0, 2);
animator.setDuration(2000);
animator.start();
```

4 旋转

```
ObjectAnimator animator = ObjectAnimator.ofFloat(mIm, "rotationX", 0,
90, 270, 360);
animator.setDuration(2000);
animator.setRepeatCount(ObjectAnimator.INFINITE);
animator.setRepeatMode(ObjectAnimator.REVERSE);
animator.start();
```

```
AnimatorSet set=new AnimatorSet();
ObjectAnimator ta = ObjectAnimator.ofFloat(mIm, "translationY",
0, 300, 100, 200, 50);
ObjectAnimator a = ObjectAnimator.ofFloat(mIm, "alpha", 0, 1);
set.setDuration(3000);
一个一个来
set.playSequentially(ta, a);
一起玩耍
set.playTogether(ta, a);
set.start();
```

对话框合集

普通对话框

1、创建Builder对象

```
AlertDialog.Builder builder = new Builder(this);
```

2、设置标题

```
builder.setTitle("上网");
```

3、设置提示内容

```
builder.setMessage("是否打开Tomcat首页");
```

4、设置确定按钮

```
builder.setPositiveButton("确定", new OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Intent intent = new Intent();
        intent.setAction(Intent.ACTION_VIEW);
        intent.setData(Uri.parse("http://192.168.143.1:8080/"));
        MainActivity.this.startActivity(intent);
    }
});
```

5、设置取消按钮，null默认对话框消失

```
builder.setNegativeButton("取消", null);
```

6、对话框和Toast一样，一定要show出来

```
builder.show();
```

单选对话框

参一：对话框的内容，一般为一个数组

参二：对话框出现后内容的默认选项，不要默认选中为小于0的任意数即可

参三：选中点击某项后的回调事件

```
builder.setSingleChoiceItems(subjects , - 1 , new OnClickListener()
```

重写onClick(DialogInterface dialog, int which)方法

参数which表示选中的内容的索引号，可将索引赋值给某个变量

2、设置按钮并设定相关点击事件

```
builder.setPositiveButton("确定",new OnClickListener() {
```

3、将对话框显示出来

```
builder.show();
```

多选对话框

参数一：内容 一般为一个数组

参数二：是否选中，false表示未选中，true表示选中，通过下标一一对应 为一个布尔数组数组长度与表示内容的数组长度相同

参数三：回调

```
builder.setMultiChoiceItems(subjects , checkedSubjects , new OnMultiChoiceClickListener() {
```

重写 onClick(DialogInterface dialog, int which,boolean isChecked)方法

which:索引 isChecked:表示选中的状态标记为true

将这个值赋值给布尔数组

2、设置按钮并设定相关点击事件

```
builder.setPositiveButton("确定",new OnClickListener() {
```

3、将对话框显示出来

```
builder.show();
```

进度条对话框 更新进度要在子线程中完成。

1、创建进度条对话框

```
ProgressDialog pd = new ProgressDialog(this);
```

2、默认是圆形进度条，我们设置成条形进度条

```
pd.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);
```

3、将对话框显示出来

```
pd.show();
```

4、设定进度条任务

进度条的最大值及任务量

5、任务完成让进度条消失

```
pd.dismiss();
```