

2.3.7 ProgressBar(进度条)

分类 [Android 基础入门教程](#)

Android 基础入门教程(Q群号 : 153836263)

本节引言：

本节给大家带来的是Android基本UI控件中的ProgressBar(进度条)，ProgressBar的应用场景很多，比如 用户登录时，后台在发请求，以及等待服务器返回信息，这个时候会用到进度条；或者当在进行一些比较耗时的操作，需要等待一段较长的时间，这个时候如果没有提示，用户可能会以为程序Crash或者手机死机了，这样会大大降低用户体验，所以在需要进行耗时操作的地方，添加上进度条，让用户知道当前的程序在执行中，也可以直观的告诉用户当前任务的执行进度等！使用进度条可以给我带来这样的便利！好了，开始讲解本节内容~ 对了，ProgressBar官方API文档：[ProgressBar](#)

1.常用属性讲解与基础实例

从官方文档，我们看到了这样一个类关系图：

```
public class
ProgressBar
extends View

java.lang.Object
├─ android.view.View
│   └─ android.widget.ProgressBar

► Known Direct Subclasses
AbsSeekBar, ContentLoadingProgressBar

► Known Indirect Subclasses
RatingBar, SeekBar, TintRatingBar
```

ProgressBar继承与View类，直接子类有AbsSeekBar和ContentLoadingProgressBar，其中AbsSeekBar的子类有SeekBar和RatingBar，可见这二者也是基于ProgressBar实现的

常用属性详解：

android:max：进度条的最大值

android:progress：进度条已完成进度值

android:progressDrawable：设置轨道对应的Drawable对象

1.0 Android基础入门教程

1.0.1 2015年最新Android基...

1.1 背景相关与系统架构分析

1.2 开发环境搭建

1.2.1 使用Eclipse + ADT + S...

1.2.2 使用Android Studio开...

1.3 SDK更新不了问题解决

1.4 Genymotion模拟器安装

1.5.1 Git使用教程之本地仓...

1.5.2 Git之使用GitHub搭建...

1.6 .9(九妹)图片怎么玩

1.7 界面原型设计

1.8 工程相关解析(各种文件...

1.9 Android程序签名打包

1.11 反编译APK获取代码&...

2.1 View与ViewGroup的概念

2.2.1 LinearLayout(线性布局)

2.2.2 RelativeLayout(相对布...

2.2.3 TableLayout(表格布局)

2.2.4 FrameLayout(帧布局)

2.2.5 GridLayout(网格布局)

2.2.6 AbsoluteLayout(绝对...

2.3.1 TextView(文本框)详解

2.3.2 EditText(输入框)详解

2.3.3 Button(按钮)与ImageB...

2.3.4 ImageView(图像视图)

2.3.5.RadioButton(单选按钮...

2.3.6 开关按钮ToggleButton...

2.3.7 ProgressBar(进度条)

2.3.8 SeekBar(拖动条)

2.3.9 RatingBar(星级评分条)

2.4.1 ScrollView(滚动条)

android:indeterminate : 如果设置成true, 则进度条不精确显示进度

android:indeterminateDrawable : 设置不显示进度的进度条的Drawable对象

android:indeterminateDuration : 设置不精确显示进度的持续时间

android:secondaryProgress : 二级进度条, 类似于视频播放的一条是当前播放进度, 一条是缓冲进度, 前者通过progress属性进行设置!

对应的再Java中我们可调用下述方法:

getMax() : 返回这个进度条的范围的上限

getProgress() : 返回进度

getSecondaryProgress() : 返回次要进度

incrementProgressBy(int diff) : 指定增加的进度

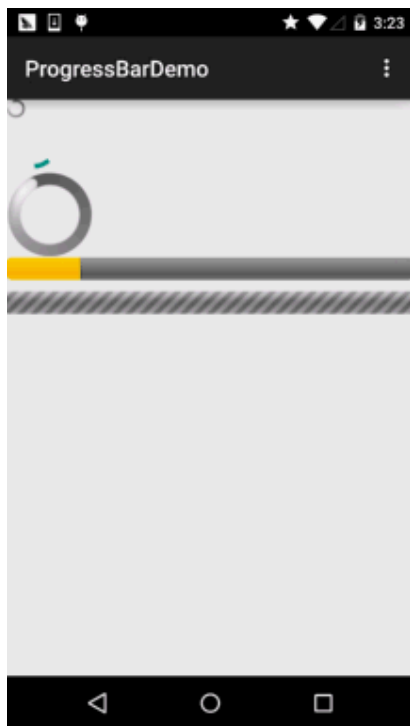
isIndeterminate() : 指示进度条是否在不确定模式下

setIndeterminate(boolean indeterminate) : 设置不确定模式下

接下来来看看系统提供的默认的进度条的例子吧!

系统默认进度条使用实例:

运行效果图:



实现布局代码:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res
/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
```

2.4.2 Date & Time组件(上)

2.4.3 Date & Time组件(下)

2.4.4 Adapter基础讲解

2.4.5 ListView简单实用

2.4.6 BaseAdapter优化

2.4.7 ListView的焦点问题

2.4.8 ListView之checkbox错...

2.4.9 ListView的数据更新问题

2.5.0 构建一个可复用的自定...

2.5.1 ListView Item多布局的...

2.5.2 GridView(网格视图)的...

2.5.3 Spinner(列表选项框)...

2.5.4 AutoCompleteTextVie...

2.5.5 ExpandableListView(...

2.5.6 ViewPager(翻转视图)...

2.5.7 Toast(吐司)的基本使用

2.5.8 Notification(状态栏通...

2.5.9 AlertDialog(对话框)详解

2.6.0 其他几种常用对话框基...

2.6.1 PopupWindow(悬浮框)...

2.6.2 菜单(Menu)

2.6.3 ViewPager的简单使用

2.6.4 DrawerLayout(官方侧...

3.1.1 基于监听的事件处理机制

3.2 基于回调的事件处理机制

3.3 Handler消息传递机制浅析

3.4 TouchListener PK OnTo...

3.5 监听EditText的内容变化

3.6 响应系统设置的事件(Co...

3.7 AsyncTask异步任务

3.8 Gestures(手势)

4.1.1 Activity初学乍练

4.1.2 Activity初窥门径

4.1.3 Activity登堂入室

4.2.1 Service初涉

4.2.2 Service进阶

4.2.3 Service精通

4.3.1 BroadcastReceiver牛...

4.3.2 BroadcastReceiver庖...

4.4.1 ContentProvider初探

<!-- 系统提供的圆形进度条,依次是大中小 -->

```
<ProgressBar
    style="@android:style/Widget.ProgressBar.Small"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
<ProgressBar
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
<ProgressBar
    style="@android:style/Widget.ProgressBar.Large"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

<!--系统提供的水平进度条-->

```
<ProgressBar
    style="@android:style/Widget.ProgressBar.Horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:max="100"
    android:progress="18" />
```

```
<ProgressBar
    style="@android:style/Widget.ProgressBar.Horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:indeterminate="true" />
```

```
</LinearLayout>
```

好吧，除了第二个能看，其他的就算了...系统提供的肯定是满足不了我们的需求的！下面我们就来讲解下实际开发中我们对进度条的处理！

2.使用动画来替代圆形进度条

第一个方案是，使用一套连续图片，形成一个帧动画，当需要进度图的时候，让动画可见，不需要的时候让动画不可见即可！而这个动画，一般是使用AnimationDrawable来实现的！好的，我们来定义一个AnimationDrawable文件：

PS:用到的图片素材：[进度条图片素材打包.zip](#)

运行效果图：

4.4.2 ContentProvider再探...

4.5.1 Intent的基本使用

4.5.2 Intent之复杂数据的传递

5.1 Fragment基本概念

5.2.1 Fragment实例精讲—...

5.2.2 Fragment实例精讲—...

5.2.3 Fragment实例精讲—...

5.2.4 Fragment实例精讲—...

5.2.5 Fragment实例精讲—...

6.1 数据存储与访问之——文...

6.2 数据存储与访问之——S...

6.3.1 数据存储与访问之——...

6.3.2 数据存储与访问之——...

7.1.1 Android网络编程要学...

7.1.2 Android Http请求头与...

7.1.3 Android HTTP请求方...

7.1.4 Android HTTP请求方...

7.2.1 Android XML数据解析

7.2.2 Android JSON数据解析

7.3.1 Android 文件上传

7.3.2 Android 文件下载 (1)

7.3.3 Android 文件下载 (2)

7.4 Android 调用 Webservice

7.5.1 WebView(网页视图)基...

7.5.2 WebView和JavaScrip...

7.5.3 Android 4.4后WebVie...

7.5.4 WebView文件下载

7.5.5 WebView缓存问题

7.5.6 WebView处理网页返...

7.6.1 Socket学习网络基础准备

7.6.2 基于TCP协议的Socket...

7.6.3 基于TCP协议的Socket...

7.6.4 基于UDP协议的Socke...

8.1.1 Android中的13种Draw...

8.1.2 Android中的13种Draw...

8.1.3 Android中的13种Draw...

8.2.1 Bitmap(位图)全解析 P...

8.2.2 Bitmap引起的OOM问题

8.3.1 三个绘图工具类详解

8.3.2 绘图类实战示例



实现步骤：

在res目录下新建一个:anim文件，然后创建amin_pgbar.xml的资源文件：

```
<?xml version="1.0" encoding="utf-8"?>
<animation-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:oneshot="false" >

    <item
        android:drawable="@drawable/loading_01"
        android:duration="200"/>

    <item
        android:drawable="@drawable/loading_02"
        android:duration="200"/>

    <item
        android:drawable="@drawable/loading_03"
        android:duration="200"/>

    <item
        android:drawable="@drawable/loading_04"
        android:duration="200"/>

    <item
        android:drawable="@drawable/loading_05"
        android:duration="200"/>

    <item
        android:drawable="@drawable/loading_06"
        android:duration="200"/>

    <item
        android:drawable="@drawable/loading_07"
        android:duration="200"/>

    <item
        android:drawable="@drawable/loading_08"
        android:duration="200"/>

    <item
        android:drawable="@drawable/loading_09"
        android:duration="200"/>

</animation-list>
```

- 8.3.3 Paint API之—— Mask...
- 8.3.4 Paint API之—— Xferm...
- 8.3.5 Paint API之—— Xferm...
- 8.3.6 Paint API之—— Xferm...
- 8.3.7 Paint API之—— Xferm...
- 8.3.8 Paint API之—— Xferm...
- 8.3.9 Paint API之—— Color...
- 8.3.10 Paint API之—— Colo...
- 8.3.11 Paint API之—— Colo...
- 8.3.12 Paint API之—— Path...
- 8.3.13 Paint API之—— Sha...
- 8.3.14 Paint几个枚举/常量值...
- 8.3.15 Paint API之——Type...
- 8.3.16 Canvas API详解(Part 1)
- 8.3.17 Canvas API详解(Part...
- 8.3.18 Canvas API详解(Part...
- 8.4.1 Android动画合集之帧...
- 8.4.2 Android动画合集之补...
- 8.4.3 Android动画合集之属...
- 8.4.4 Android动画合集之属...
- 9.1 使用SoundPool播放音...
- 9.2 MediaPlayer播放音频与...
- 9.3 使用Camera拍照
- 9.4 使用MediaRecord录音
- 10.1 TelephonyManager(电...
- 10.2 SmsManager(短信管理...
- 10.3 AudioManager(音频管...
- 10.4 Vibrator(振动器)
- 10.5 AlarmManager(闹钟服务)
- 10.6 PowerManager(电源服...
- 10.7 WindowManager(窗口...
- 10.8 LayoutInflater(布局服务)
- 10.9 WallpaperManager(壁...
- 10.10 传感器专题(1)——相...
- 10.11 传感器专题(2)——方...
- 10.12 传感器专题(3)——加...
- 10.12 传感器专题(4)——其...
- 10.14 Android GPS初涉
- 11.0 《2015最新Android基...

```

<item
    android:drawable="@drawable/loading_10"
    android:duration="200"/>

<item
    android:drawable="@drawable/loading_11"
    android:duration="200"/>

<item
    android:drawable="@drawable/loading_12"
    android:duration="200"/>

</animation-list>

```

接着写个布局文件，里面仅仅有一个ImageView即可，用于显示进度条，把src设置为上述drawable资源即可！最后到MainActivity.java

```

public class MainActivity extends AppCompatActivity {

    private ImageView img_pgbar;
    private AnimationDrawable ad;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        img_pgbar = (ImageView) findViewById(R.id.img_pgbar);
        ad = (AnimationDrawable) img_pgbar.getDrawable();
        img_pgbar.postDelayed(new Runnable() {
            @Override
            public void run() {
                ad.start();
            }
        }, 100);
    }

}

```

这里只是写了如何启动动画，剩下的就你自己来了哦~在需要显示进度条的时候，让ImageView可见；在不需要的时候让他隐藏即可！另外其实Progressbar本身有一个indeterminateDrawable，只需把这个参数设置上述的动画资源即可，但是进度条条的图案大小是不能直接修改的，需要Java代码中修改，如果你设置了宽高，而且这个宽高过大的时候，你会看到多个进度条...自己权衡下吧~



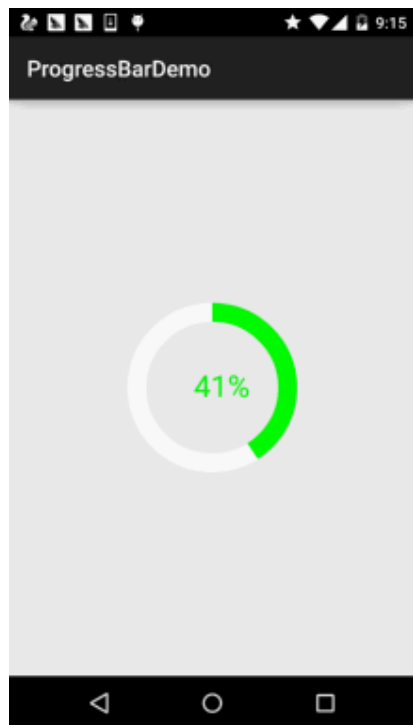
反馈

3.自定义圆形进度条

相信你看完2会吐槽，卧槽，这么坑爹，拿个动画来坑人，哈哈，实际开发中都这样，当然上述这种情况只适用于不用显示进度的场合，如果要显示进度的场合就没用处了，好吧，接下来看下 网上一个简单的自定义圆形进度条！代码还是比较简单，容易理解，又兴趣可以看

看，或者进行相关扩展~

运行效果图：



实现代码：

自定义View类：

```
/**
 * Created by Jay on 2015/8/5 0005.
 */
public class CirclePgBar extends View {

    private Paint mBackPaint;
    private Paint mFrontPaint;
    private Paint mTextPaint;
    private float mStrokeWidth = 50;
    private float mHalfStrokeWidth = mStrokeWidth / 2;
    private float mRadius = 200;
    private RectF mRect;
    private int mProgress = 0;
    //目标值，想改多少就改多少
    private int mTargetProgress = 90;
    private int mMax = 100;
    private int mWidth;
    private int mHeight;

    public CirclePgBar(Context context) {
        super(context);
        init();
    }

    public CirclePgBar(Context context, AttributeSet attrs) {
```

```

        super(context, attrs);
        init();
    }

    public CirclePgBar(Context context, AttributeSet attrs, int
defStyleAttr) {
        super(context, attrs, defStyleAttr);
        init();
    }

    //完成相关参数初始化
    private void init() {
        mBackPaint = new Paint();
        mBackPaint.setColor(Color.WHITE);
        mBackPaint.setAntiAlias(true);
        mBackPaint.setStyle(Paint.Style.STROKE);
        mBackPaint.setStrokeWidth(mStrokeWidth);

        mFrontPaint = new Paint();
        mFrontPaint.setColor(Color.GREEN);
        mFrontPaint.setAntiAlias(true);
        mFrontPaint.setStyle(Paint.Style.STROKE);
        mFrontPaint.setStrokeWidth(mStrokeWidth);

        mTextPaint = new Paint();
        mTextPaint.setColor(Color.GREEN);
        mTextPaint.setAntiAlias(true);
        mTextPaint.setTextSize(80);
        mTextPaint.setTextAlign(Paint.Align.CENTER);
    }

    //重写测量大小的onMeasure方法和绘制View的核心方法onDraw()
    @Override
    protected void onMeasure(int widthMeasureSpec, int heightMe
asureSpec) {
        super.onMeasure(widthMeasureSpec, heightMeasureSpec);
        mWidth = getRealSize(widthMeasureSpec);
        mHeight = getRealSize(heightMeasureSpec);
        setMeasuredDimension(mWidth, mHeight);
    }

    @Override
    protected void onDraw(Canvas canvas) {
        initRect();
        float angle = mProgress / (float) mMax * 360;
        canvas.drawCircle(mWidth / 2, mHeight / 2, mRadius, mBa
ckPaint);
        canvas.drawArc(mRect, -90, angle, false, mFrontPaint);
        canvas.drawText(mProgress + "%", mWidth / 2 + mHalfStro

```

```

        keWidth, mHeight / 2 + mHalfStrokeWidth, mTextPaint);
        if (mProgress < mTargetProgress) {
            mProgress += 1;
            invalidate();
        }
    }

    public int getRealSize(int measureSpec) {
        int result = 1;
        int mode = MeasureSpec.getMode(measureSpec);
        int size = MeasureSpec.getSize(measureSpec);

        if (mode == MeasureSpec.AT_MOST || mode == MeasureSpec.
UNSPECIFIED) {
            //自己计算
            result = (int) (mRadius * 2 + mStrokeWidth);
        } else {
            result = size;
        }

        return result;
    }

    private void initRect() {
        if (mRect == null) {
            mRect = new RectF();
            int viewSize = (int) (mRadius * 2);
            int left = (mWidth - viewSize) / 2;
            int top = (mHeight - viewSize) / 2;
            int right = left + viewSize;
            int bottom = top + viewSize;
            mRect.set(left, top, right, bottom);
        }
    }
}

```

然后在布局文件中加上：

```

<com.jay.progressbardemo.CirclePgBar
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>

```

就是这么简单~

本节小结：

本节给大家介绍了Android中的常用控件：ProgressBar讲解了基本用法，以及实际开发中对于进度条的两种实现方法，第二个自定义进度条可以自行完善，然后用到实际开发中~！好的，本节就到这里，谢谢~



	<div>在线实例</div> <div><div>· HTML 实例</div><div>· CSS 实例</div><div>· JavaScript 实例</div><div>· Ajax 实例</div><div>· jQuery 实例</div><div>· XML 实例</div><div>· Java 实例</div></div>	<div>字符集&工具</div> <div><div>· HTML 字符集设置</div><div>· HTML ASCII 字符集</div><div>· HTML ISO-8859-1</div><div>· HTML 实体符号</div><div>· HTML 拾色器</div><div>· JSON 格式化工具</div></div>	<div>最新更新</div> <div><div>· PHP接收 json , 并...</div><div>· Foundation CSS ...</div><div>· Foundation CSS ...</div><div>· Foundation 图标...</div><div>· Foundation 网格...</div><div>· Foundation 块状...</div><div>· Foundation 网...</div></div>	<div>站点信息</div> <div><div>· 意见反馈</div><div>· 免责声明</div><div>· 关于我们</div><div>· 文章归档</div></div>
				<div>关注微信</div> <div></div> <div>Copyright © 2013-2015 菜鸟教程 runoob.com All Rights Reserved. 备案号：闽ICP备15012807号-1</div>