1. 在value文件夹下创建attrs.xml文件用来定义自定义控件所需要的属性

<?xml version="1.0" encoding="utf-8"?>  
<resources>  
   
 <attr name="titleText" format="string"/>  
 <attr name="titleTextColor" format="color"/>  
 <attr name="titleTextSize" format="dimension"/>  
   
 <declare-styleable name="TitleView">  
 <attr name="titleText"/>  
 <attr name="titleTextColor"/>  
 <attr name="titleTextSize"/>  
 </declare-styleable>  
  
</resources>

1.上半部分是定义属性的名称为“titleText”等，后面的format是属性的取值（取值从string,color,demension,integer,enum,reference,float,boolean,fraction,flag中选）

2.<declare-styleable>之内的就是自定义控件的属性了，第一个name=“TitleView”的位置表示自定义控件的名字是TitleText就像TextView，Button等一样

二．在布局文件中使用控件

<com.example.as.myview.view.TitleView  
 android:id="@+id/titleView"  
 android:layout\_height="200dp"  
 android:layout\_width="100dp"  
 android:titleTextColor="#ff0000"  
 android:titleTextSize="40sp"  
 android:titleText="测试"/>

通过自定义控件的所在的包名来引用

1. 为自定义控件重写相关方法来
2. public class TitleView extends View {  
    //获取控件的titleText属性的值  
    private String titleText;  
    //获取控件的titleTextColor的值  
    private int titleTextColor;  
    //获取控件的titleTextSize的值  
    private int titleTextSize;  
    //重写构造器  
    public TitleView(Context context, AttributeSet attributeSet)  
    {  
    super(context,attributeSet);  
    TypedArray typedArray=context.obtainStyledAttributes(attributeSet,R.styleable.TitleView);  
    int a=typedArray.getIndexCount();  
    for (int i=0;i<a;i++)  
    {  
    int attr=typedArray.getIndex(i);  
    //浏览各个属性，并获得各个属性的值  
    switch(attr)  
    {  
    case R.styleable.TitleView\_titleText:  
    titleText=typedArray.getString(attr);  
    break;  
    case R.styleable.TitleView\_titleTextColor:  
    titleTextColor=typedArray.getColor(attr, Color.*BLUE*);  
    break;  
    case R.styleable.TitleView\_titleTextSize:  
    titleTextSize=typedArray.getDimensionPixelSize(attr,(int) TypedValue.*applyDimension*(TypedValue.*COMPLEX\_UNIT\_SP*,16,getResources().getDisplayMetrics()));  
    break;  
    }  
    typedArray.recycle();  
    //定义一个画笔在区域内实现功能  
    //如本例就是写出titleText属性的值  
    //设置大小为titleTextSize属性的值  
    //颜色为titleTextColor属性的值  
    Paint mpaint=new Paint();  
    mpaint.setTextSize(titleTextSize);  
    mpaint.setColor(titleTextColor);  
    Rect mbond=new Rect();  
    mpaint.getTextBounds(titleText,0,titleText.length(),mbond);  
    }  
     
    }  
     
    @Override  
    protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {  
    super.onMeasure(widthMeasureSpec, heightMeasureSpec);  
    }  
     
    @Override  
    protected void onDraw(Canvas canvas) {  
    super.onDraw(canvas);  
    }  
   }

此处的onMeasure和onDraw方法都没有重写

此后的讲解具体请参照http://m.blog.csdn.net/article/details?id=50991560

1. TypedArray:此对象代表一个数组，这个数组封装了自定义控件的各种属性值，如titleText，titleTextSize等属性，之后通过索引来为各种属性设置具体的值，如此例中的

int attr=typedArray.getIndex(i);  
 //浏览各个属性，并获得各个属性的值  
 switch(attr)  
 {  
 case R.styleable.TitleView\_titleText:  
 titleText=typedArray.getString(attr);  
 break;  
 case R.styleable.TitleView\_titleTextColor:  
 titleTextColor=typedArray.getColor(attr, Color.*BLUE*);  
 break;  
 case R.styleable.TitleView\_titleTextSize:  
 titleTextSize=typedArray.getDimensionPixelSize(attr,(int) TypedValue.*applyDimension*(TypedValue.*COMPLEX\_UNIT\_SP*,16,getResources().getDisplayMetrics()));  
第一行的getIndex（）就是通过索引来锁定到某个属性然后做出修改，最后别忘了调用TypedArray对象的recycle（）方法来回收资源

2）返回TypedArray数组的方法

public TypedArray obtainStyledAttributes (int[] attrs)；

2.public TypedArray obtainStyledAttributes (int resid, int[] attrs)；

3.public TypedArray obtainAttributes (AttributeSet set, int[] attrs)；

4.public TypedArray obtainStyledAttributes (AttributeSet set, int[] attrs, int defStyleAttr, int defStyleRes)。

1.2.4都是Resources.Theme的函数，3是Resourse的函数，其中3.4较为常用

3方法：注意此方法中有个AttributeSet参数，它对应着attrs中的属性，AttributeSet对象中的数据是从xml布局文件中读取出来的，所以调用此方法的时候一定要为各个属性值设置好，否则会报错。

4方法：返回一个与 attrs 属性相对应的数组。另外，如果在 AttributeSet 中为 attrs 指定了样式属性，那么这个样式属性就会应用在这些属性上。

attribute 最终由下面四个因素决定：

1. 在 AttributeSet 中定义的属性（Any attribute values in the given AttributeSet）；
2. AttributeSet 指定的样式资源文件（The style resource specified in the AttributeSet (named “style”)）；
3. 由 defStyleAttr 和 defStyleRes 指定的样式资源文件（The default style specified by defStyleAttr and defStyleRes）；
4. 主题中的默认值（The base values in this theme）。

上面四种元素的优先级是从上到下排序的，也就是说：如果在 AttributeSet 中定义了某个属性的值，那么无论后面的样式属性如何定义，它的值都不会改变。

接下来我们分别解释下，函数中各参数的含义：

1. AttributeSet set ：XML 中定义的属性值，可能为 null；
2. int[] attrs ：目标属性值；
3. int defStyleAttr ：在当前主题中有一个引用指向样式文件，这个样式文件将 TypedArray 设置默认值。如果此参数为 0 则表示不进行默认值设置。
4. int defStyleRes ：默认的样式资源文件，只有当 defStyleAttr 为 0 或者无法在对应的主题下找到资源文件时才起作用。如果此参数为 0 则表示不进行默认设置。