1. **Canvas类android.graphics.Canvas**

Canvas类好比手机中的画纸

Canvas类提供了三个构造方法：   
Public Canvas（）；构造一个默认无参的Canvas对象   
Public Canvas（Bitmap bitmap）；根据一个Bitmap构造一个Canvas对象   
Public Canvas（GL gl）；根据一个GL来构造一个对象

**二、Paint类android.graphics.Paint**

Paint类包含有用来画几何图形、文本、位图的类型和颜色等信息，如果把Canvas类看作是画板，那我们可以把Paint类看做是画笔，可以根据需要画出不同颜色和样式的图形、文本等内容。   
Paint类有三个构造方法：   
 Public Paint():构造一个缺省的Paint对象   
 Public Paint(int flags):根据指定的flags来构造一个Paint对象，创建之后可以用 setFlags()方法来更改   
 Public Paint(Paint paint):根据指定的paint对象来构造一个Paint对象   
Paint类提供了很多方法来设置和获取Paint对象的属性，比如：   
 public int getColor ():获得Paint对象的颜色值   
 public ColorFilter getColorFilter ():获得颜色过滤器   
 public float getTextSize ():获得字体大小数值   
 public void setStyle (Paint.Style style):设置paint的类型

**三  Color类android.graphics.Color**

此类 定义了一些方法来创建和转换颜色值。颜色被表示为封装的数值，这个数值由四个字节组成，分别是：alpha、red、green、blue，这些值是非自左乘的，也就是说任何透明性只存储在alpha部分，而不是在颜色组成部分。每一部分按照如下的顺序保存：（alpha<<24）| (red<<16)|<green<<8)|blue.每一部分的取值范围在0-255之间，0意味着这部分不起作 用，255表示100%的起作用。因此不透明的黑色应该是0xFF000000,不透明的白色应该是0xFFFFFFFF 。   
Color类提供了12个常量值来代表不同的颜色值，我们在开发工程中可以直接调用这些常量值来设置我们所要修饰的文本、图形等对象。   
Color类提供了一个无参的构造方法Color()   
Color类提供了一些方法来进行颜色值的创建和转换如下：   
其中三个方法是用来返回一个颜色常量值的红绿蓝分色，数值分别在0-255之间，如下：   
Public static int red(int color);   
Public static int green(int color);   
Public static int blue(int color);   
其中Public static int rgb(int red,int green,int blue);输入红绿蓝三色，返回一个RGB颜色值。   
下面几个方法大家可以参考Android API来了解：   
Public static int HSVToColor(int alpha,float[] hsv);   
Public static int HSVToColor(float[] hsv);   
Public static void RGBToHSV(int red,int green,int blue,float[] hsv);   
Public static int alpha(int color);   
Public static int argb(int alpha,int red,int green,int blue);   
Public static colorToHSV(int color,float[] hsv);   
Public static parseColor(String colorString);

**四．Typeface类android.graphics.Typeface**

Typeface类定义字体和字体内在的类型。这个类被用在画笔Paint设置的时候，比如用textSize,textSkewX和textScale设置来指定text在画的时候如何来显示和测量。

Typeface 提供了一些常量值来表示自身的一些属性，比如BOLD，BOLD\_ITALIC，ITALIC等，开发者可以用 defaultFrOPhone SDNtyle(int style)获得内在的Typeface对象。读者可以参考详细的API文档再这里就不再赘述了。

下面我们来看一下Typeface的主要方法。

Typeface类没有构造方法，通常如果一个类没有构造函数就无法通过构造来生成一个对象实体，此时一般会有特定的静态方法来取得这个类的实体。Typeface就提供了四个静态方法间接来得到Typeface实体分别如下：

Public static Typeface create（Typeface family，int style）；

此方法返回一个与已经存在的Typeface字形体系相匹配且类型是指定类型的Typeface。如果你想得到一个与已经存在的Typeface字形体系 相类似，但是样式不一样的Typeface时可以调用此方法。其中family参数可以为null，如果为空则表示从默认的Typeface字形体系中选 择。

Public static Typefaxe create(String familyname,int style);

此方法通过给定的字形体系的名称familyname和指定的类型返回一个Typeface对象。如果给定字形体系的名称为null，我们可以通过getStyle（）方法来获得返回Typeface对象的真正的属性。

Public static Typeface createFromAsset(AssetManager mgr,String path);

此方法通过规定的字体数据来返回一个Typeface对象。第一个参数为资源管理器，第二个参数是指定字体类型。

Public static Typeface defaultFrOPhone SDNtyle(int style);此方法返回一个指定类型的Typeface对象

Typeface还提供了另外三个方法：

Public int getStyle();此方法返回Typeface内在的类型属性

Public final Boolean isBold();如果getStyle（）有BOLD位组将返回true

Public final Boolean isItalic();如果getStyle（）有ITALIC位组将返回true

**五、Path类android.graphics.Path**

Path类（一组区域）的描画，类囊括多种几何图形比如直线线段、二次曲线、三次曲线等，

调用Canvas.drawPath()方法可以将Path以所定义的paint的方式来画到画板上或者填出图形，也可以用paint所指定方式来画图形。

**六、RectF类android.graphics.RectF和Rect类android.graphics.Rect**

RectF 这个类包含一个矩形的四个单精度浮点坐标。矩形通过上下左右4个边的坐标来表示一个矩形。这些坐标值属性可以被直接访问，用width（）和 height（）方法可以获取矩形的宽和高。注意：大多数方法不会检查这些坐标分类是否错误（也就是left<=right和top& lt;=bottom）.

RectF一共有四个构造方法：

RectF（）构造一个无参的矩形

RectF（float left,float top,float right,float bottom）构造一个指定了4个参数的矩形

RectF(Rect F r)根据指定的RectF对象来构造一个RectF对象(对象的左边坐标不变)

RectF(Rect r)根据给定的Rect对象来构造一个RectF对象

RectF提供了很多方法，下面介绍几个方法：

Public Boolean contain(RectF r);判断一个矩形是否在此矩形内，如果在这个矩形内或者和这个矩形等价则返回true，同样类似的方法还有public Boolean contain(float left,float top,float right,float bottom)和public Boolean contain(float x,float y)。

Public void union(float x,float y)更新这个矩形，使它包含矩形自己和（x，y）这个点。

Android.graphics.Rect类，这个类同android.graphics.RectF很相似，不同的地方是Rect类的坐标是用整形表示的，而RectF的坐标是用单精度浮点型表示的。

**七、Point类android.graphics.Point**

   这个类从字面意思就可以看出它跟点有关系，是点的一个对象类。

这个类有两个属性，分别是：X坐标和y坐标。

构造函数有三个：Point（），Point（int x，int y），Point（Point p）

主要方法有：

Public void set（x，y）；重新设定一下x，y的坐标

Public final void offset(int dx,int dy);给坐标一个补偿值，值可以使正的也可以是负的。

Public final void negate();否定坐标值。

Point类和android.graphics.PointF类似，不同点是前者坐标值的类型是int型，而后者的坐标值是float型。除此之外PointF类多加了几个方法，比如：

Public final float length();返回（0，0）点到该点的距离。

Public static float length(float x,float y);返回（0，0）点到（x，y）点的距离。