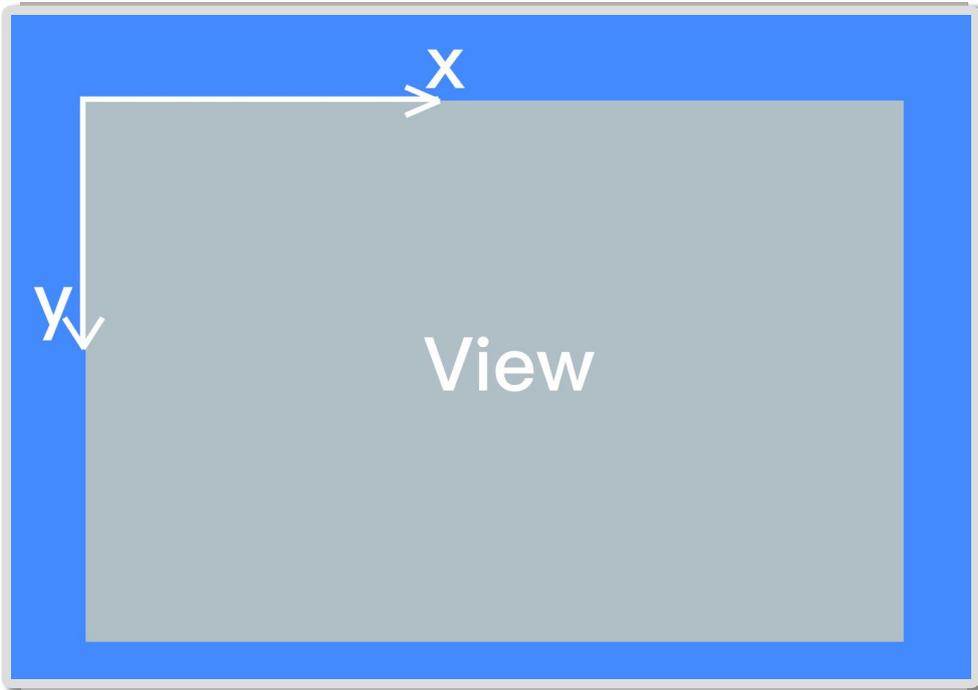


# 图形的位置和尺寸测量

## 绘制的基本要素：

- 重写 `onDraw()`
- 使用 `Canvas` 来绘制
- 使用 `Paint` 来配置
- 坐标系



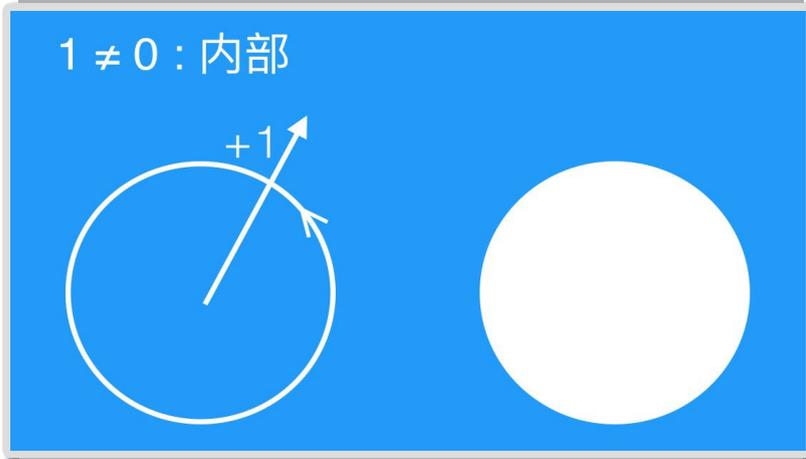
- 尺寸单位是像素，而不是 dp。转换方式：

```
public static float dp2px(float dp) {  
    return  
    TypedValue.applyDimension(TypedValue.COMPLEX_UNIT_DI  
P, dp,  
  
    Resources.getSystem().getDisplayMetrics());  
}
```

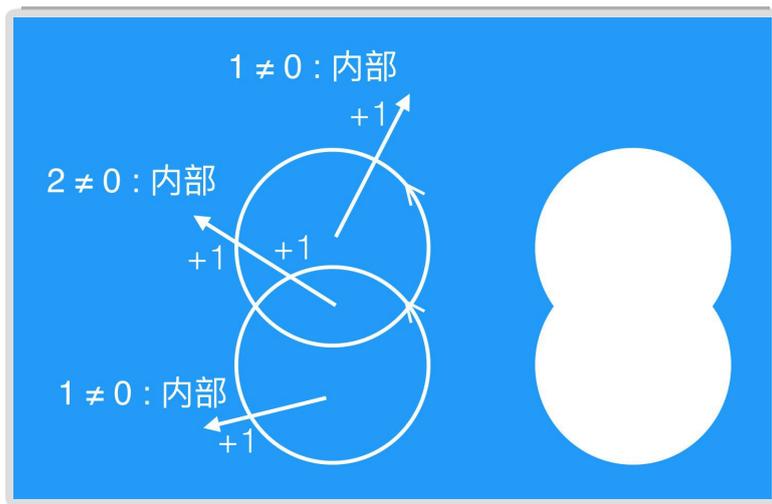
# Path 的方向以及封闭图形的内外判断：

◆ Winding：如果方向相反的穿插次数相等则为外部，不等则为内部：

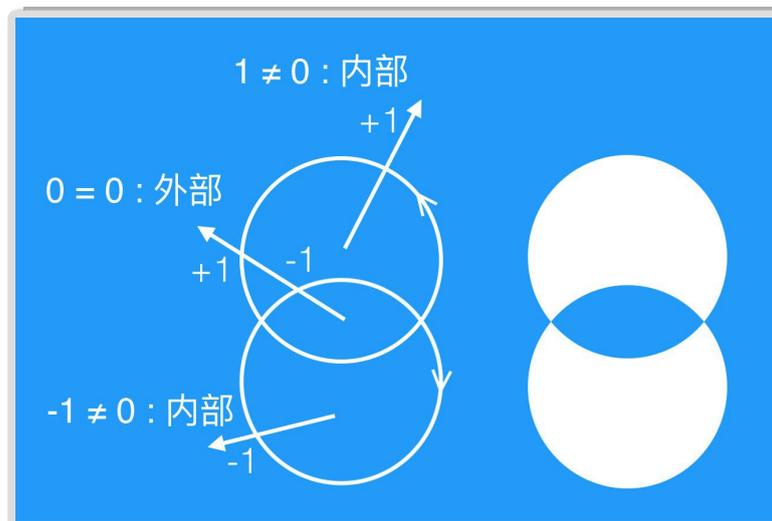
○ 单圆：



○ 双圆同向：

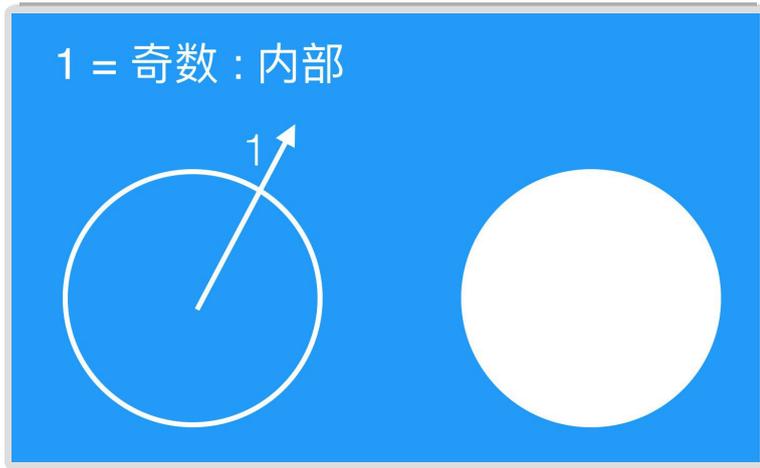


○ 双圆不同向：

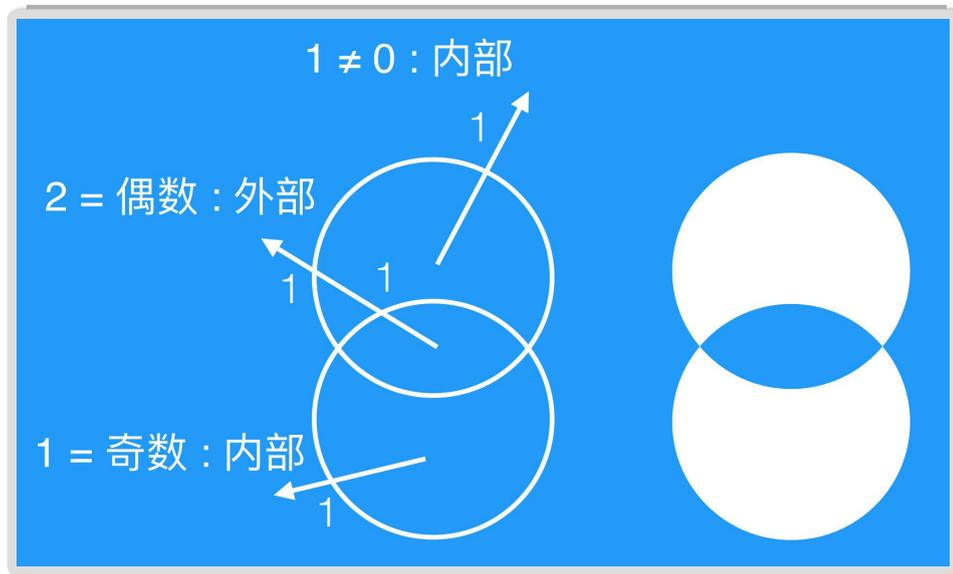


◆ Even Odd : 不考虑方向。穿插奇数次则为内部，偶数次则为外部：

■ 单圆



■ 双圆



## PathMeasure

把 Path 对象填入，用于对 Path 做针对性的计算（例如图形周长）。

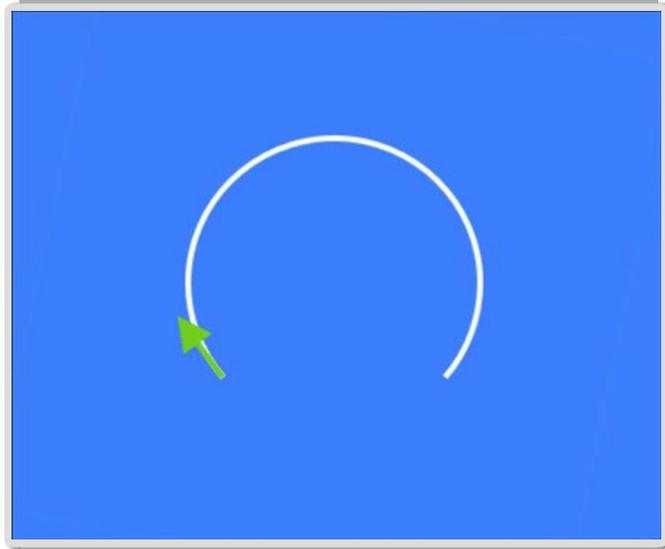
## 仪表盘

- 用 drawArc() 绘制弧形
- 三角函数的计算 横向的位移是 cos，纵向的位移是 sin
- PathDashPathEffect

加上 **PathEffect** 之后，就只绘制 **effect**，而不绘制原图形。所以需要弧线和刻度分别绘制，一共两次。

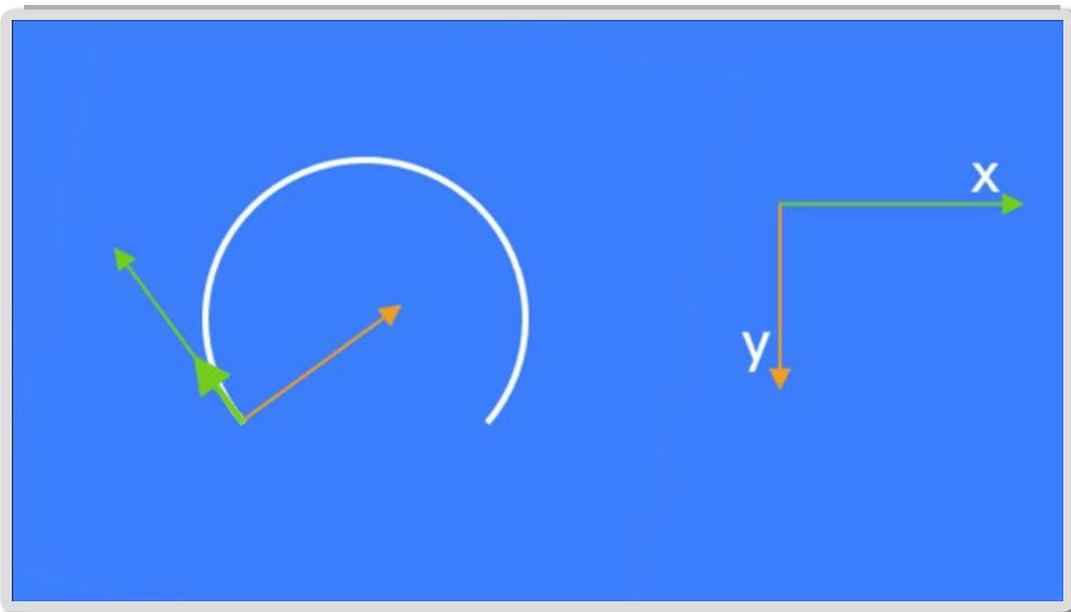
○ **dash** 的方向

■ 绘制 **dash** 的轨迹是这样的：

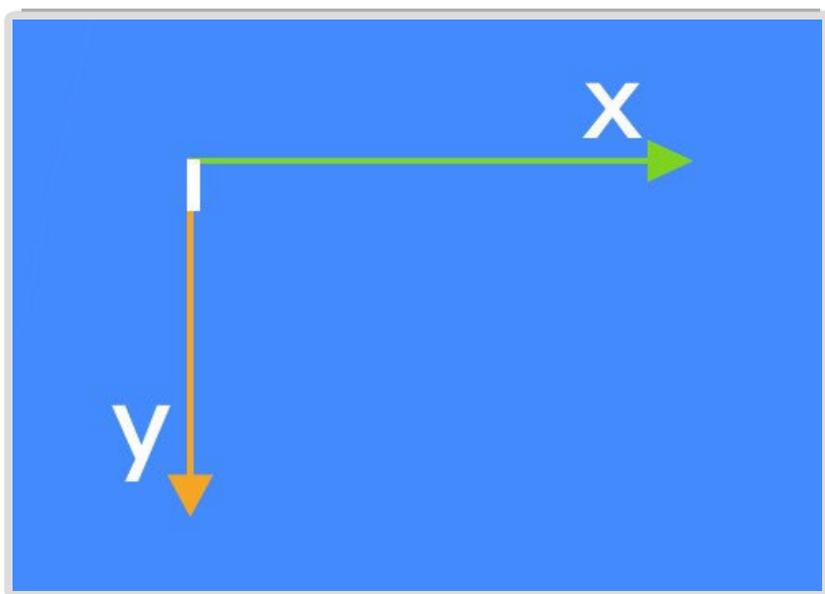


■

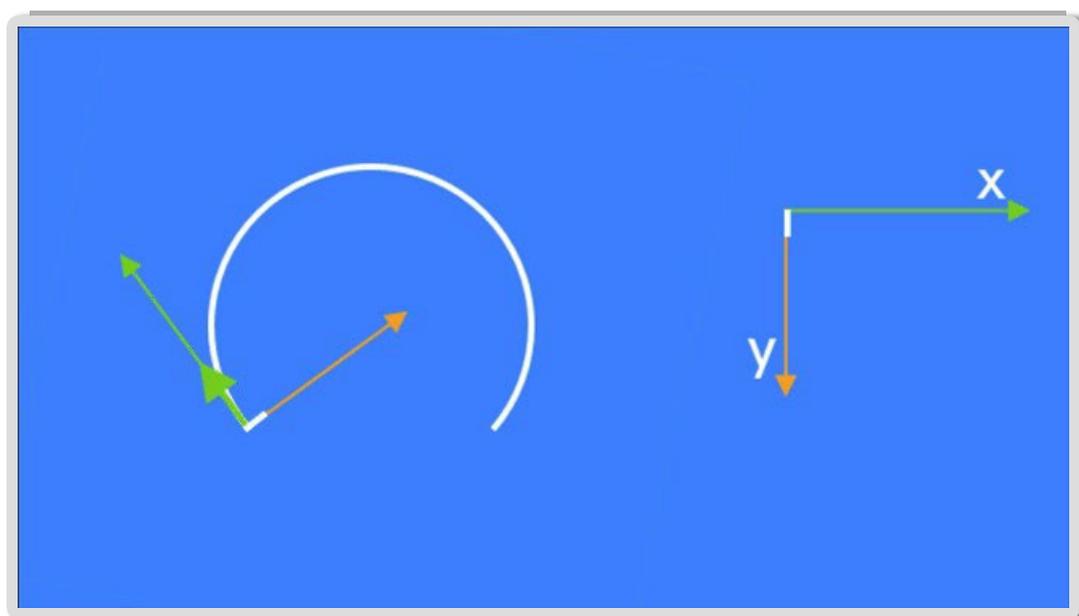
而由于 **x** 轴的正向是轨迹的正向（这是规定），所以实际上的物理模型是这样的：



那么，如果你要加上刻度，就应该这样设置：



然后，你的刻度就会沿着轨迹绘制：



✧ advance 计算

## 饼图

- 用 `drawArc()` 绘制扇形
- 用 `Canvas.translate()` 来移动扇形，并用 `Canvas.save()` 和 `Canvas.restore()` 来保存和恢复位置
- 用三角函数 `cos` 和 `sin` 来计算偏移