**ConstraintLayout约束布局最起码要有横竖两个方向的约束条件：**

**对齐：**

**layout\_constraintStart\_toEndOf="@id/tv"**

**表示自身的开始（左边）于对应控件的后面（右边）对齐，constraintStart表示自身的开始（左边）toEndOf 表示对应控件的后面（右边）,想和右边对齐就是\_toStartOf ；**

**layout\_constraintTop\_toTopOf="@id/tv"**

**表示自身的上面于对应控件的上面对齐，constraintStart表示自身的上面 \_toEndOf 表示对应控件的上面，如果想和下面对齐 \_toBottomOf ；**

**Start和left区别，left表示相对的左边，而start表示布局开始计算的一边，正常环境下是一样的，少数地区（中东）start表示的是右边，因为他们是从右边开始阅读的；**

**居中：**

**双向对齐就会居中**

**app:layout\_constraintTop\_toTopOf="@id/tv"**

**app:layout\_constraintBottom\_toBottomOf="@id/tv"**

**布局相对于TV，上面和tv上面对齐，下面也和tv下面对齐,那自身相对于tv就是上下居中的，左右居中类似；如果想自身上面和TV下面对齐，自身下面也和TV下面对齐，那就是相对TV的下面对齐；**

**如果左右同时对齐，width=0dp,控件会被拉伸至对应控件的宽度，**

**weight比重：（1：1）**

**几个控件相互约束，左右均分的话，控件width = 0dp,**

**tv1:**

**android:layout\_width="0dp"**

**app:layout\_constraintStart\_toStartOf="parent"**

**app:layout\_constraintEnd\_toStartOf="@+id/tv2"**

**tv2:**

**android:layout\_width="0dp"**

**app:layout\_constraintStart\_toEndOf="@id/tv1"**

**app:layout\_constraintEnd\_toEndOf="parent"**

**两个控件相互拉扯就会均分布局；**

**weight比重：**

**android:layout\_width="0dp"**

**app:layout\_constraintHorizontal\_weight="1"**

**几个控件需要相互约束，这个属性才有效果；**

**圆周角度定位：**

**圆周角度定位，确定相对控件作为圆心，角度，半径就可以确定自身位置**

**v2:**

**app:layout\_constraintCircle="@id/v1"**

**app:layout\_constraintCircleRadius="40dp"**

**app:layout\_constraintCircleAngle="45"**

**v2在v1为圆心45度角上，距离40dp;**

**文字基准线对齐：**

**两个控件Text内容基准线对齐**

**tv2:**

**layout\_constraintBaseline\_toBaselineO="@id/tv1"**

**tv2的文字和tv1的文字内容基准线对齐**

**注意点：**

**app:layout\_constrainedWidth="true"**

**让控件宽度不会超过限制条件；**

**app:layout\_constraintHorizontal\_bias="0.0"**

**控件水平位置左右约束的百分比，在左边是0.0，在右边是1.0，（控件左右约束力量的百分比），这个效果只有在左右约束都是固定的时候才有效果；**

**app:layout\_constraintVertical\_chainStyle=""**

**两个控件竖向约束力松紧效果，packed表示拉紧spread\_inside表示分开，spread默认均分**

**app:layout\_constraintDimensionRatio="1:2"**

**控件宽高比，当控件宽或者高确定，可以通过这个属性控制高宽；**

**app:layout\_constraintWidth\_percent="0.5"**

**控制控件的宽度占父容器的宽度的百分比；**

**Guideline 百分比的线，方便内部控件调整位置；**

**例子：**

<**android.support.constraint.Guideline**

**android:layout\_width="0dp"**

**android:layout\_height="match\_parent"**

**android:orientation="vertical"**

**app:layout\_constraintGuide\_percent="0.13"**/>

**orientation 线的方向；**

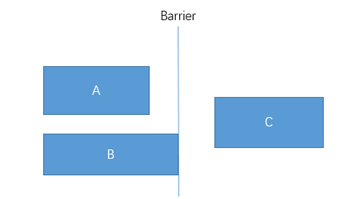
**layout\_constraintGuide\_percent 线的位置（左右上下百分比的位置）**

**layout\_constraintGuide\_begin="200dp" 线的位置（左右上下dp的位置从左边开始计算）**

**layout\_constraintGuide\_end="200dp" 线的位置（左右上下dp的位置从右边开始计算）**

**Barrier**

假设有3个控件ABC，C在AB的右边，但是AB的宽是不固定的，这个时候C无论约束在A的右边或者B的右边都不对。当出现这种情况可以用Barrier来解决。Barrier可以在多个控件的一侧建立一个屏障，如下所示：



这个时候C只要约束在Barrier的右边就可以了，代码如下：

**<TextView
android:id="@+id/A"/>
<TextView
android:id="@+id/B"/>
<android.support.constraint.Barrier
android:id="@+id/barrier"
app:barrierDirection="right"
app:constraint\_referenced\_ids="A,B" />
<TextView
android:id="@+id/C"
app:layout\_constraintLeft\_toRightOf="@+id/barrier" />**

**Group**

Group可以把多个控件归为一组，方便隐藏或显示一组控件，举个例子：

<android.support.constraint.Group
android:visibility="invisible"
app:constraint\_referenced\_ids="TextView1,TextView2,TextView3" />

<TextView
android:id="@+id/TextView1"/>
<TextView
android:id="@+id/TextView2"/>
<TextView
android:id="@+id/TextView3"/>

用Group把TextView1,TextView1,TextView3归为一组，可以一起控制这组控件的可见性。

Group继承ConstraintHelper；可以自定义ConstraintHelper实现自己想要的功能。

**自定义ConstraintHelper**

CircularRevealHelper帮助实现几组控件实现相同的动画效果。

**public class** CircularRevealHelper **extends** ConstraintHelper {

@Override

**public void** updatePostLayout(ConstraintLayout container) {

**super**.updatePostLayout(container);

*//版本判断*

**if** (Build.VERSION.***SDK\_INT*** < Build.VERSION\_CODES.***LOLLIPOP***){ **return**;}

*//获取所有的View*

**for** (**int** i= 0;i<**mCount**;i++) {

View view = container.getViewById(**mIds**[i]);

Double radius = Math.*hypot*(view.getWidth(), view.getHeight());

Float f = Float.*parseFloat*(String.*valueOf*(radius));

*//实现动画*

ViewAnimationUtils.*createCircularReveal*(view, 0, 0, 0f, f)

.setDuration(2000L)

.start();

}

}

}

xml布局：

<**View**

**android:background="#00ff00"**

**android:id="@+id/v1"**/>

<**View**

**android:background="#00ffff"**

**android:id="@+id/v2"**/>

<**com.shinhoandroid.constraintlayout.CircularRevealHelper**

**app:constraint\_referenced\_ids="v1,v2"** />