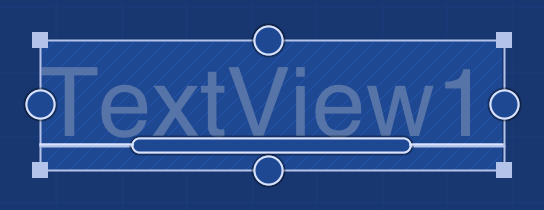
ConstraintLayout高级使用

一 控键类型



调整尺寸控键：该控键允许你调整widget尺寸。

IMG_256

侧约束控键：该控键让你指定widget的位置。例如，你可以使用widget的左侧控键到其他widget的右侧控键相隔24dp

IMG_256

基线约束控键：该控键帮助你对齐任意两个widget的文字部分，与widget的大小无关。例如你有两个不同尺寸的widget但是你想要他们的文字部分对齐。

IMG_256

二 属性介绍

1 位置控制：8个边界控制属性

注：最左边表示可移动的最左边，左边表示View的左边边界.

app:layout\_constraintLeft\_toLeftOf

app:layout\_constraintLeft\_toRightOf 我最左边的位置 在别人的右边

app:layout\_constraintRight\_toRightOf

app:layout\_constraintRight\_toLeftOf

app:layout\_constraintTop\_toTopOf

app:layout\_constraintTop\_toBottomOf

app:layout\_constraintBottom\_toBottomOf

app:layout\_constraintBottom\_toTopOf



如图：



<!--如图，左边一个A，右边一个C，我如果想新建一个B在A C之间，如下-->

<Button app:layout\_constraintLeft\_toRightOf="@+id/bt\_a"

app:layout\_constraintRight\_toLeftOf="@+id/bt\_c"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="B"/>

<!--字面理解：1.我最左边的位置，在button A的右边-->

<!--字面理解：1.我最右边的位置，在button C的左边-->

如上图中，最左边和最右边的位置已经确定，B出现在A和C中间，但是如果我不想在中间怎么办（比如说我想靠右一点）

偏移属性:2个

layout\_constraintHorizontal\_bias（水平方向偏移）（范围0-1）

layout\_constraintVertical\_bias（垂直方向偏移）（范围0-1）

如图：B的水平偏移为0   


app:layout\_constraintHorizontal\_bias="0"

如图：B的水平偏移为0.5   

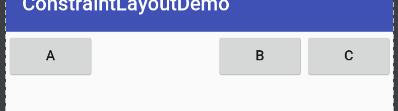

app:layout\_constraintHorizontal\_bias="0.5"

如图：B的水平偏移为0.5   


app:layout\_constraintHorizontal\_bias="0.5"

如图：B的水平偏移为0.7   


app:layout\_constraintHorizontal\_bias="0.7"

如图：B的水平偏移为1   


app:layout\_constraintHorizontal\_bias="1"

总结：   
1.通过8个边界约束属性可以固定View的最左边、最右边、最上面、最下面的位置   
2.通过设置偏移属性，可以控制View在边界范围移动，最左边是0，最右边是1，中间是0.5   
3.当设置了边界约束属性后，View会自动出现在中间，也就是说，默认的偏移属性是0.5

大小控制

layout\_constraintHorizontal\_weight //水平方向上比重，类似线性布局

layout\_constraintVertical\_weight //垂直方向上比重，类似线性布局

<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:ignore="MissingConstraints">

<!--A边界控制属性 有 左 和 右-->

<Button android:id="@+id/bt\_a"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:text="A"

app:layout\_constraintHorizontal\_weight="1"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toLeftOf="@id/bt\_b"/>

<!--B边界控制属性 也有 左 和 右-->

<Button android:id="@+id/bt\_b"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:text="B"

app:layout\_constraintHorizontal\_weight="1"

app:layout\_constraintLeft\_toRightOf="@id/bt\_a"

app:layout\_constraintRight\_toLeftOf="@id/bt\_c"/>

<!--C边界控制属性 只有右-->

<Button android:id="@+id/bt\_c"

android:layout\_width="0dp"

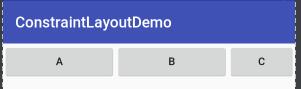
android:layout\_height="wrap\_content"

android:text="C"

app:layout\_constraintHorizontal\_weight="1"

app:layout\_constraintRight\_toRightOf="parent"/>

</android.support.constraint.ConstraintLayout>



<?xml version="1.0" encoding="utf-8"?>

<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:ignore="MissingConstraints">

<TextView android:background="#0f0"

android:id="@+id/bt\_a"

android:layout\_width="0dp"

android:layout\_height="0dp"

android:text="A"

app:layout\_constraintBottom\_toTopOf="@id/bt\_b"

app:layout\_constraintHorizontal\_weight="1"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toLeftOf="@id/bt\_b"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintVertical\_weight="1"/>

<TextView android:background="#0f0"

android:id="@+id/bt\_b"

android:layout\_width="0dp"

android:layout\_height="0dp"

android:text="B"

app:layout\_constraintBottom\_toTopOf="@id/bt\_c"

app:layout\_constraintHorizontal\_weight="1"

app:layout\_constraintLeft\_toRightOf="@id/bt\_a"

app:layout\_constraintRight\_toLeftOf="@id/bt\_c"

app:layout\_constraintTop\_toBottomOf="@id/bt\_a"

app:layout\_constraintVertical\_weight="1"/>

<TextView android:background="#0f0"

android:id="@+id/bt\_c"

android:layout\_width="0dp"

android:layout\_height="0dp"

android:text="C"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_weight="1"

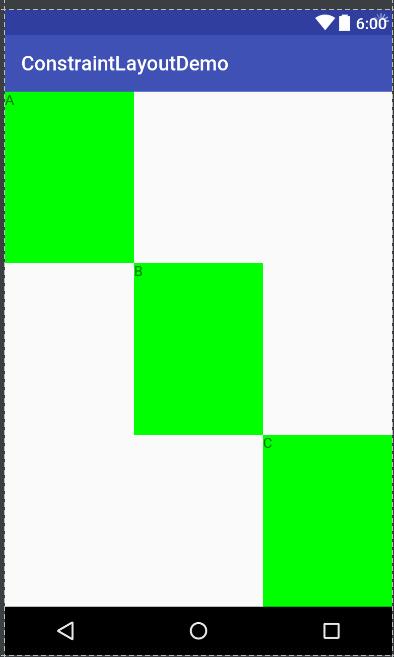
app:layout\_constraintLeft\_toRightOf="@id/bt\_b"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/bt\_b"

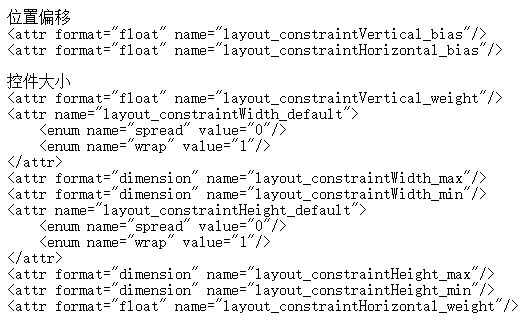
app:layout\_constraintVertical\_weight="1"/>

</android.support.constraint.ConstraintLayout>



结论：   
1.实现水平方向线性布局，所有的View都必须设置左右边界控制属性，而且相互控制   
2.实现比重大小控制，必须设置layout\_width=”0dp”





ConstraintLayout可用的约束 （constraint-layout:1.0.2）

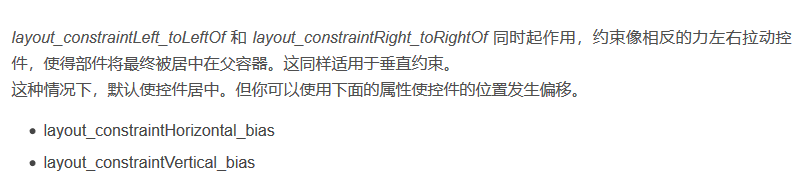
相对位置定位



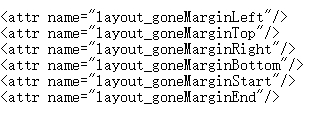
边距（Margins）



中心定位

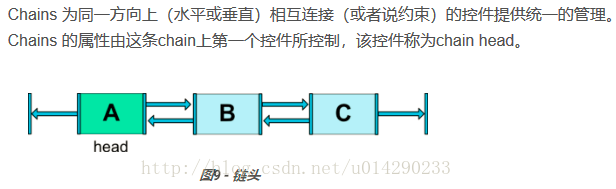


可见性行为（View.GONE）



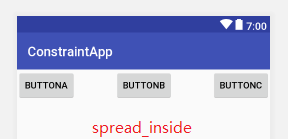
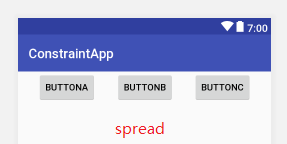
链（Chains）

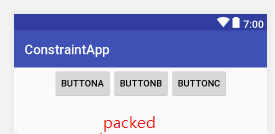
1522650467(1)



该属性有三种值：

* spread 默认值。均匀分布（控件宽为wrap\_content或确定值）
* spread\_inside 与spread类型，只是chain两端的控件不占父容器的空间，多余的空间由chain里的控件平分。
* Packed ，chain 上的所有控件居中紧挨在一起

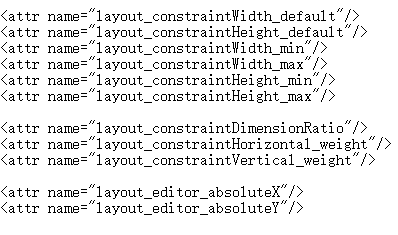




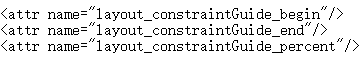
当设置spread时，可以分别为每个子控件设置权重，对应的属性为layout\_constraintHorizontal\_weight和layout\_constraintVertical\_weight，与LinearLayout的weight类似，需要注意的是，此时子控件的layout\_width需要设成0dp。

当设置packed时，可以设置chain在水平或垂直方向上的偏移量，对应的属性为layout\_constraintHorizontal\_bias和layout\_constraintVertical\_bias，取值在0到1之间。

控件大小-宽高比例，权重分配



虚拟助手对象（Guideline）----相当布局的框边界



ConstraintLayout还可以通过创建对象guideline来辅助布局。   
Guideline有水平和垂直两张情况：

* 水平（horizontal）的Guideline高度为0，宽度是ConstraintLayout的宽度
* 垂直（vertical）的Guideline宽度为0，高度是ConstraintLayout的高度

Guideline 有三种属性：

- layout\_constraintGuide\_begin 距离左侧或顶部的距离   
- layout\_constraintGuide\_end 距离右侧或底部的距离   
- layout\_constraintGuide\_percent float类型。距离left或top的比例。

辅助布局属性



<https://blog.csdn.net/qq_33689414/article/details/75103731>

<https://blog.csdn.net/stevenhenry/article/details/78954981>

<https://blog.csdn.net/u014290233/article/details/77744093>

<https://blog.csdn.net/h8800830/article/details/53014661>