# 自定义ViewGroup

1. 继承于View类，实现ViewParent, ViewManager接口
2. 实现onMeasure方法，测绘自身容器大小
3. 实现onLayout方法，绘制每一个子view的大小。

# onMeasure

**参数：**

参数是两个int型，分别表示宽的“mode+sizes+大小”，高的“mode+size”。使用一个32位的二进制数，32和31位代表了mode的值，后30位代表size的值。

**int** mode = MeasureSpec.*getMode*(measureSpec);

**int** size = MeasureSpec.*getSize*(measureSpec);

**必须操作事项:**

1. 必须调用setMeasuredDimension方法设置整个容器大小。该大小根据需求设置。
2. 必须调用measureChild测绘子view大小，该方法最终调用子view的onMeasure方法。
3. 设置子View的测绘的LayoutParams，该数据会在子View的**getMeasuredWidth ，getMeasuredHeight**方法中获取到

例子：

|  |
| --- |
| //在布局文件中设定，width：match\_parent，height：wrap\_content  super.onMeasure(widthMeasureSpec, heightMeasureSpec);  //获取父view的最宽大显示限制  int rw = MeasureSpec.*getSize*(widthMeasureSpec);  int rh = MeasureSpec.*getSize*(heightMeasureSpec);  int childCount = this.getChildCount();  for (int i = 0; i < childCount; i++) {  View child = this.getChildAt(i);  this.measureChild(child, widthMeasureSpec, heightMeasureSpec);  LayoutParams lParams =(LayoutParams)child.getLayoutParams();  lParams.height=childheight;  lParams.width=childwith;  }  setMeasuredDimension(rw,rh); |

# onLayout

1. 获取子view数量，一般由addView方法添加，或者在xml布局中添加了view
2. 遍历每个子View，设置layout。

|  |
| --- |
| protected void onLayout(boolean arg0, int arg1, int arg2, int arg3, int arg4) {  int count = getChildCount();  View child;  //根据每个子View的设定大小，设置每一个子View的位置和大小  for(int i = 0 ;i < count;i++) {  child = getChildAt(i);  child.layout(左边距离,上边距离 ,右边距离，底边距离);  }  } |

# 相关API

|  |  |
| --- | --- |
| addView | 添加view，具体实现：  requestLayout();  invalidate(**true**);  addViewInner(child, index, params, **false**); |
| requestLayout |  |
| invalidate |  |
| dispatchDraw |  |
|  |  |
|  |  |
|  |  |
|  |  |