

ViewGroup

子 View

measure()

ViewGroup.onMeasure()

子View1.measure()

子View2.measure()

## HenCoder Android UI 部分 布局基础

layout()

保存 layout() 传入的坐标

ViewGroup.onLayout()

子View1.layout()

子View2.layout()



HenCoder 的第二季：自定义 View 第二部分——布局过程的自定义，从这期正式开始了。好像有点小激动。

废话少说，直入正题。

## 简介

之前我说过，自定义 View 最关键的有三个点：绘制、布局和触摸反馈。上一季讲过的绘制，内容虽然多（好像讲了 8 期？），但难度上其实是最简单的。而布局就和它相反，布局过程的技术知识，内容比较少（应该 3 期就能讲完），但你先要理解它的内部工作原理才能正确理解它的使用，而它的工作原理却是有点绕的。所以如果你跟着 HenCoder 学习布局过程的自定义，大致会是这样一种体验：看完视频或者读完文章，感觉吸收了一大波知识，信息量好大难以吞咽的样子，但当你真的把它们吞咽下去，然后下载了我的练习项目去做练习的时候，却又发现：卧槽，就这么点东西？布局过程的自定义竟然这么简单？

会者不难这个词在很多地方都讲得通，但在布局过程的自定义上，尤为适用。

## 总结

有人说：什么？简介完了就是总结了？

嚯嚯，对的。所有的内容都在视频里讲完了，所以在这里就把视频里讲到的关键点总结一下：

## 布局过程的含义

布局过程，就是程序在运行时利用布局文件的代码来计算出实际尺寸的过程。

## 布局过程的工作内容

两个阶段：测量阶段和布局阶段。

**测量阶段：**从上到下递归地调用每个 View 或者 ViewGroup 的 `measure()` 方法，测量他们的尺寸并计算它们的位置； **布局阶段：**从上到下递归地调用每个 View 或者 ViewGroup 的 `layout()` 方法，把测得的它们的尺寸和位置赋值给它们。

## View 或 ViewGroup 的布局过程

1. 测量阶段，`measure()` 方法被父 View 调用，在 `measure()` 中做一些准备和优化工作后，调用 `onMeasure()` 来进行实际的自我测量。`onMeasure()` 做的事，View 和 ViewGroup 不一样：

1. **View：**View 在 `onMeasure()` 中会计算出自己的尺寸然后保存；
2. **ViewGroup：**ViewGroup 在 `onMeasure()` 中会调用所有子 View 的 `measure()` 让它们进行自我测量，并根据子 View 计算出的期望尺寸来计算出它们的实际尺寸和位置（实际上 99.99% 的父 View 都会使用子 View 给出的期望尺寸来作为实际尺寸，原因在下期或下下期会讲到）然后保存。同时，它也会根据子 View 的尺寸和位置来计算出自己的尺寸然后保存；

2. 布局阶段，`layout()` 方法被父 View 调用，在 `layout()` 中它会保存父 View 传进来的自己的位置和尺寸，并且调用 `onLayout()` 来进行实际的内部布局。`onLayout()` 做的事，View 和 ViewGroup 也不一样：

1. **View：**由于没有子 View，所以 View 的 `onLayout()` 什么也不做。
2. **ViewGroup：**ViewGroup 在 `onLayout()` 中会调用自己的所有子 View 的 `layout()` 方法，把它们的尺寸和位置传给它们，让它们完成自我的内部布局。

## 布局过程自定义的方式

三类：

1. 重写 `onMeasure()` 来修改已有的 View 的尺寸；
2. 重写 `onMeasure()` 来全新定制自定义 View 的尺寸；
3. 重写 `onMeasure()` 和 `onLayout()` 来全新定制自定义 ViewGroup 的内部布局。

# 第一类自定义的具体做法

也就是重写 `onMeasure()` 来修改已有的 view 的尺寸的具体做法：

1. 重写 `onMeasure()` 方法，并在里面调用 `super.onMeasure()`，触发原有的自我测量；
2. 在 `super.onMeasure()` 的下面用 `getMeasuredWidth()` 和 `getMeasuredHeight()` 来获取到之前的测量结果，并使用自己的算法，根据测量结果计算出新的结果；
3. 调用 `setMeasuredDimension()` 来保存新的结果。

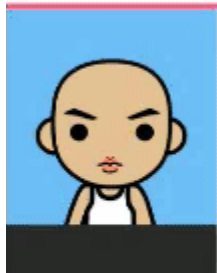
## 练习项目

为了避免转头就忘，强烈建议你趁热打铁，做一下这个练习项目：

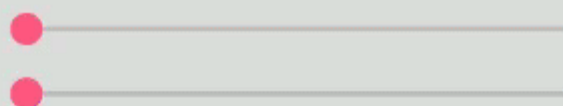
[HenCoderPracticeLayout1](#)（微信用户可以点底部的「阅读原文」）

# HenCoder 布局 1

SquareImageView

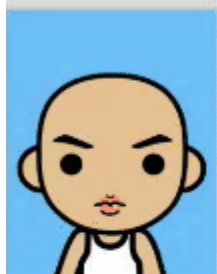


外框调节

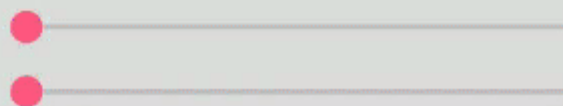


宽度

高度



外框调节



宽度

高度