# Day：View的绘制流程

## View的测量

1、View的measure方法，根据父布局传递来的MeasureSpec，来确定自己的测量宽、高的值。

2、measure方法为final方法，不可重写，但measure方法有调用onMeasure方法，我们可以重写onMeasure来手动改变View的测量宽、高的值（可参考ImageView、TextView）。

3、onMeasure方法的默认实现中，当SpecMode为AT\_MOST模式或EXACTLY模式时，测量宽、高就是其SpecSize中的值。

4、View的MeasureSpec是由父布局传递而来，父布局又是如何为每一个子View生成MeasureSpec的呢？

5、ViewGroup提供了measureChildren方法，我们可以在重写ViewGroup的onMeasure方法时，调用measureChildren方法来测量每一个子View（因为ViewGroup首先它也是一个View，可在onMeasure方法中拿到ViewGroup的MeasureSpec）。

6、在measureChildren方法中，会根据ViewGroup自己的MeasureSpec和子View自己设置的宽高，来为每一个子View设置MeasureSpec。最后调用childView的measure方法，并把给childView生成的MeasureSpec传递过去。

## View的布局

1、View的layout方法中，会为调用setFrame方法为View自己指定位置，如果是ViewGroup还会调用onLayout方法为所有的childView指定位置。

2、View调用setFrame方法为mLeft、mTop、mRight、mBottom几个属性赋值，来确定自己相对于父布局的位置。

3、ViewGroup的onLayout方法是一个抽象方法，子类需要实现此方法，并在实现中根据ViewGroup的布局逻辑计算出每个childView的位置，然后调用childView的layout方法进行布局。

## View的绘制

1、View的draw方法会按流程分别绘制背景、content、childView、前景、默认焦点等。其中调用onDraw方法绘制View的内容；如果是ViewGroup还会调用dispatchDraw方法绘制childView。

2、ViewGroup的dispatchDraw方法中，会调用drawChild方法来绘制所有的childView，drawChild方法则直接调用childView的draw方法进行绘制。

3、就这样按视图结构层次依次绘制下去。

4、我们可以通过重写View的onDraw方法，拿到Canvas来绘制我们想要绘制的内容。

## 附

ViewGroup为childView生成MeasureSpec的规则：

* View配置固定宽高时，View的SpecMode总是EXCATLY模式，SpecSize总是LayoutParams中设置的值
* View配置match\_parent时，ViewGroup是什么模式，View就是什么模式，SpecSize是ViewGroup剩余的空间
* View配置wrap\_content时，View的SpecMode总是AT\_MOST模式，SpecSize总是ViewGroup的剩余空间

注意：

当View配置wrap\_content时，View的SpecMode总是为AT\_MOST，SpecSize总是为ViewGroup的剩余空间。View在根据MeasureSpec设置测量宽高时，就会设置成ViewGroup的剩余空间，与期望的wrap\_content不符。

解决方案为自定义View时重写onMeasure方法，在View的SpecMode为AT\_MOST时，为View指定一个宽高。ImageView、TextView等都是如此操作，可参考其源码。