Google开源的FlexboxLayout

原创 2016-05-15 stormzhang [AndroidDeveloper](javascript:void(0);)

阅读本文大概需要8分钟。

最近Google开源了一个项目叫「FlexboxLayout」，但是貌似没多少资料对这个项目介绍的，那么今天来讲讲这个项目的作用以及简单的使用介绍。

**1什么是 Flexbox**

简单来说 Flexbox 是属于web前端领域CSS的一种布局方案，是2009年W3C提出了一种新的布局方案，可以简便、完整、响应式地实现各种页面布局，并且 React Native 也是使用的 Flex 布局。

你可以简单的理解为 Flexbox 是CSS领域类似 Linearlayout 的一种布局，但是要比 Linearlayout 要强大的多。

**2什么是 FlexboxLayout？**

刚才说了 Flexbox 是CSS领域的比较强大的一个布局，我们在 Android 开发中使用 Linearlayout + RelativeLayout 基本可以实现大部分复杂的布局，但是Google就想了，有没有类似 Flexbox 的一个布局呢？这使用起来一个布局就可以搞定各种复杂的情况了，于是 FlexboxLayout 就应运而生了。

所以 FlexboxLayout 是针对 Android 平台的，实现类似 Flexbox 布局方案的一个开源项目，开源地址：

https://github.com/google/flexbox-layout

**3使用方式**

使用方式很简单，只需要添加以下依赖：

compile 'com.google.android:flexbox:0.1.2'

xml中这样使用：

或者代码中这样使用：

使用起来是不是很像Linearlayout的用法，只不过有很多属性你们比较陌生，这些属性都是Flexbox布局本身具备的，别着急，下面跟你们介绍下FlexboxLayout的一些具体属性的用法与意义。

**4支持的属性**

**flexDirection**

flexDirection 属性决定主轴的方向（即项目的排列方向）。类似 LinearLayout 的 vertical 和 horizontal。

有四个值可以选择：

* row（默认值）：主轴为水平方向，起点在左端。
* row-reverse：主轴为水平方向，起点在右端。
* column：主轴为垂直方向，起点在上沿。
* column-reverse：主轴为垂直方向，起点在下沿。

**flexWrap**

默认情况下 Flex 跟 LinearLayout 一样，都是不带换行排列的，但是flexWrap属性可以支持换行排列。这个也比 LinearLayout 吊啊有三个值：

* nowrap ：不换行
* wrap：按正常方向换行
* wrap-reverse：按反方向换行

**justifyContent**

justifyContent属性定义了项目在主轴上的对齐方式。

* flex-start（默认值）：左对齐
* flex-end：右对齐
* center： 居中
* space-between：两端对齐，项目之间的间隔都相等。
* space-around：每个项目两侧的间隔相等。所以，项目之间的间隔比项目与边框的间隔大一倍。

**alignItems**

alignItems属性定义项目在副轴轴上如何对齐。

* flex-start：交叉轴的起点对齐。
* flex-end：交叉轴的终点对齐。
* center：交叉轴的中点对齐。
* baseline: 项目的第一行文字的基线对齐。
* stretch（默认值）：如果项目未设置高度或设为auto，将占满整个容器的高度。

**alignContent**

alignContent属性定义了多根轴线的对齐方式。如果项目只有一根轴线，该属性不起作用。

* flex-start：与交叉轴的起点对齐。
* flex-end：与交叉轴的终点对齐。
* center：与交叉轴的中点对齐。
* space-between：与交叉轴两端对齐，轴线之间的间隔平均分布。
* space-around：每根轴线两侧的间隔都相等。所以，轴线之间的间隔比轴线与边框的间隔大一倍。
* stretch（默认值）：轴线占满整个交叉轴。

**5子元素属性**

除以上之外，FlexboxLayout还支持如下子元素属性：

**layout\_order**

默认情况下子元素的排列方式按照文档流的顺序依次排序，而order属性可以控制排列的顺序，负值在前，正值灾后，按照从小到大的顺序依次排列。我们说之所以 FlexboxLayout 相对LinearLayout强就是因为一些属性比较给力，order就是其中之一。

**layout\_flexGrow**

layout\_flexGrow 属性定义项目的放大比例，默认为0，即如果存在剩余空间，也不放大。一张图看懂。跟 LinearLayout 中的weight属性一样。

如果所有项目的 layout\_flexGrow  属性都为1，则它们将等分剩余空间（如果有的话）。如果一个项目的 layout\_flexGrow  属性为2，其他项目都为1，则前者占据的剩余空间将比其他项多一倍。

**layout\_flexShrink**

layout\_flexShrink  属性定义了项目的缩小比例，默认为1，即如果空间不足，该项目将缩小。

如果所有项目的 layout\_flexShrink  属性都为1，当空间不足时，都将等比例缩小。如果一个项目的flex-shrink属性为0，其他项目都为1，则空间不足时，前者不缩小。  
负值对该属性无效。

**layout\_alignSelf**

layout\_alignSelf  属性允许单个子元素有与其他子元素不一样的对齐方式，可覆盖 alignItems 属性。默认值为auto，表示继承父元素的 alignItems 属性，如果没有父元素，则等同于stretch。

* auto (default)
* flex\_start
* flex\_end
* center
* baseline
* stretch

该属性可能取6个值，除了auto，其他都与align-items属性完全一致。

**layout\_flexBasisPercent**

layout\_flexBasisPercent 属性定义了在分配多余空间之前，子元素占据的main size主轴空间，浏览器根据这个属性，计算主轴是否有多余空间。它的默认值为auto，即子元素的本来大小。

**6不同之处**

**跟传统的CSS中的Flexbox布局有些不同的是：**

1. 没有 flex-flow 属性

2. 没有 flex 属性

3. layout\_flexBasisPercent 属性即为CSS中 flexbox 中的 flexBasis 属性

4. 不支持 min-width 和 min-height 两个属性

以上就是 FlexboxLayout 的一些基本介绍与基本用法，值得提醒大家的是，本身这个项目也是一个很好的自定义View的学习资料，值得大家学习借鉴！

**参考：**

本文的很多 Flexbox 方面的知识大量参考了我司同事的这篇文章，想要更多的了解 Flexbox 相关的知识建议阅读这里：

http://w4lle.github.io/2016/05/08/Flexbox/