

TOC

2. 参考
3. 本节目标
4. 视频
5. 正文
6. 总结

Flutter 布局 - 理解约束、布局调试工具

2021-12-23 · flutter



猫哥微信订阅号

第一时间推送阅读



前言

每个渲染引擎都有一套自己的规则，比如浏览器 html 流式布局，Flutter 里的是约束布局，这是一种科学的设计。

我想引擎设计出来肯定是有考虑，如：

- 容易学习，快速上手
- 零配置可用
- 高容错性（溢出、默认尺寸）
- 常见场景方案（横向、纵向、滚动、叠加、嵌套、响应式、自适应、浮动、大列表...）

参考

经典的约束规则文章 Marcelo Glasberg 撰写，后来被 flutter.dev 收录，掘金也有人翻译。

- <https://medium.com/flutter-community/flutter-the-advanced-layout-rule-even-beginners-must-know-edc9516d1a2>
- <https://juejin.cn/post/6846687593745088526>
- <https://docs.flutter.dev/development/ui/layout/constraints>

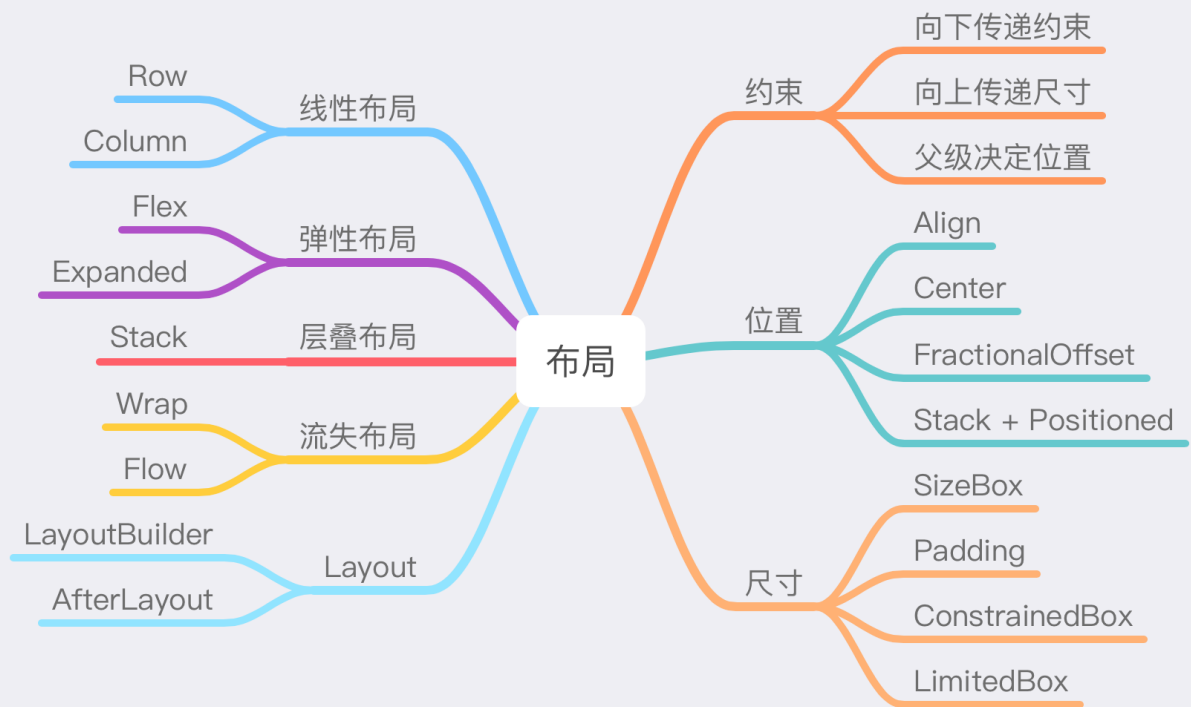
本节目标

- 理解布局约束原则
- 掌握调试布局工具
- 名词，紧约束、松约束、unbounded

视频

<https://www.bilibili.com/video/BV1X44y177Ns/>

正文

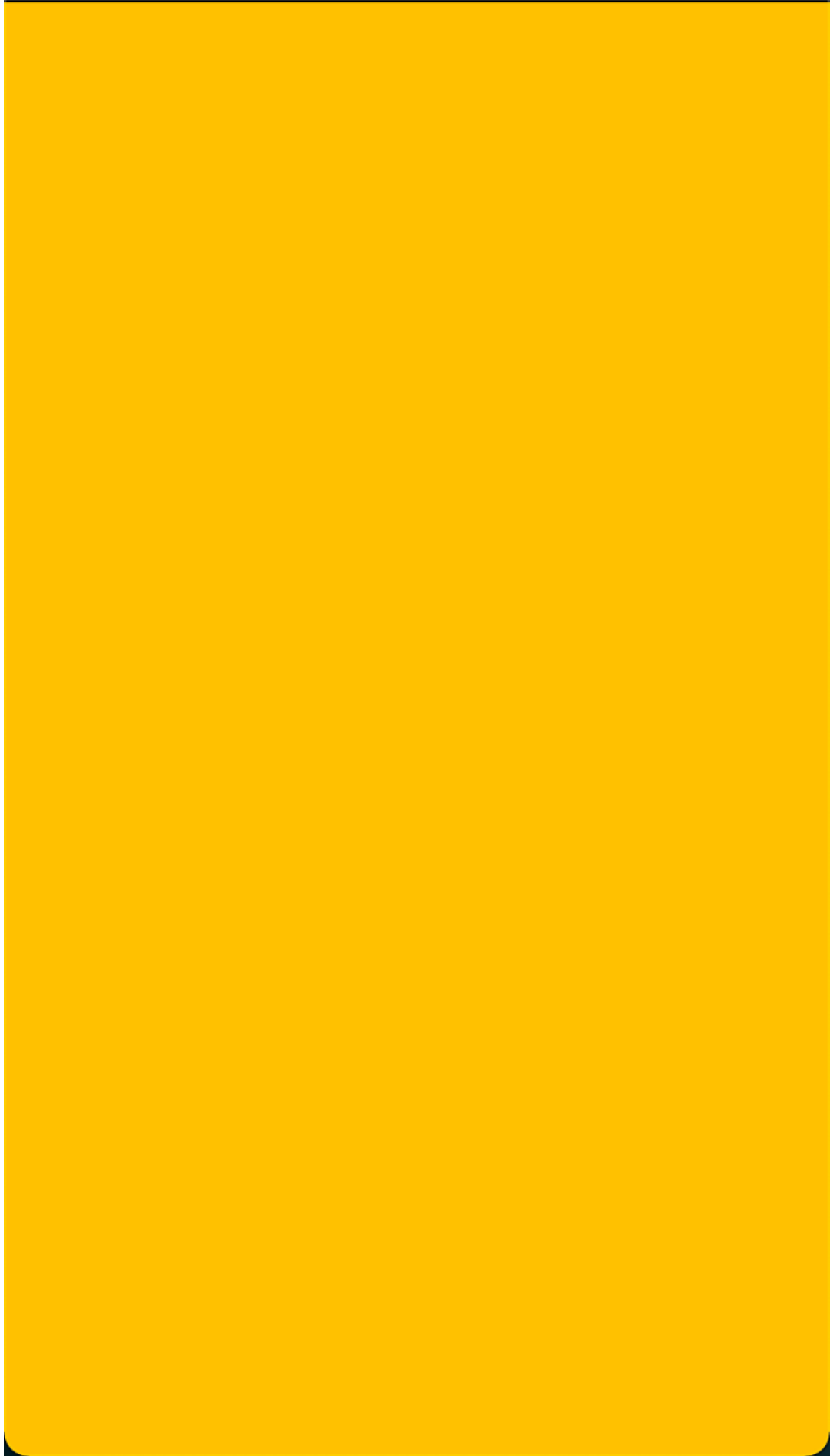


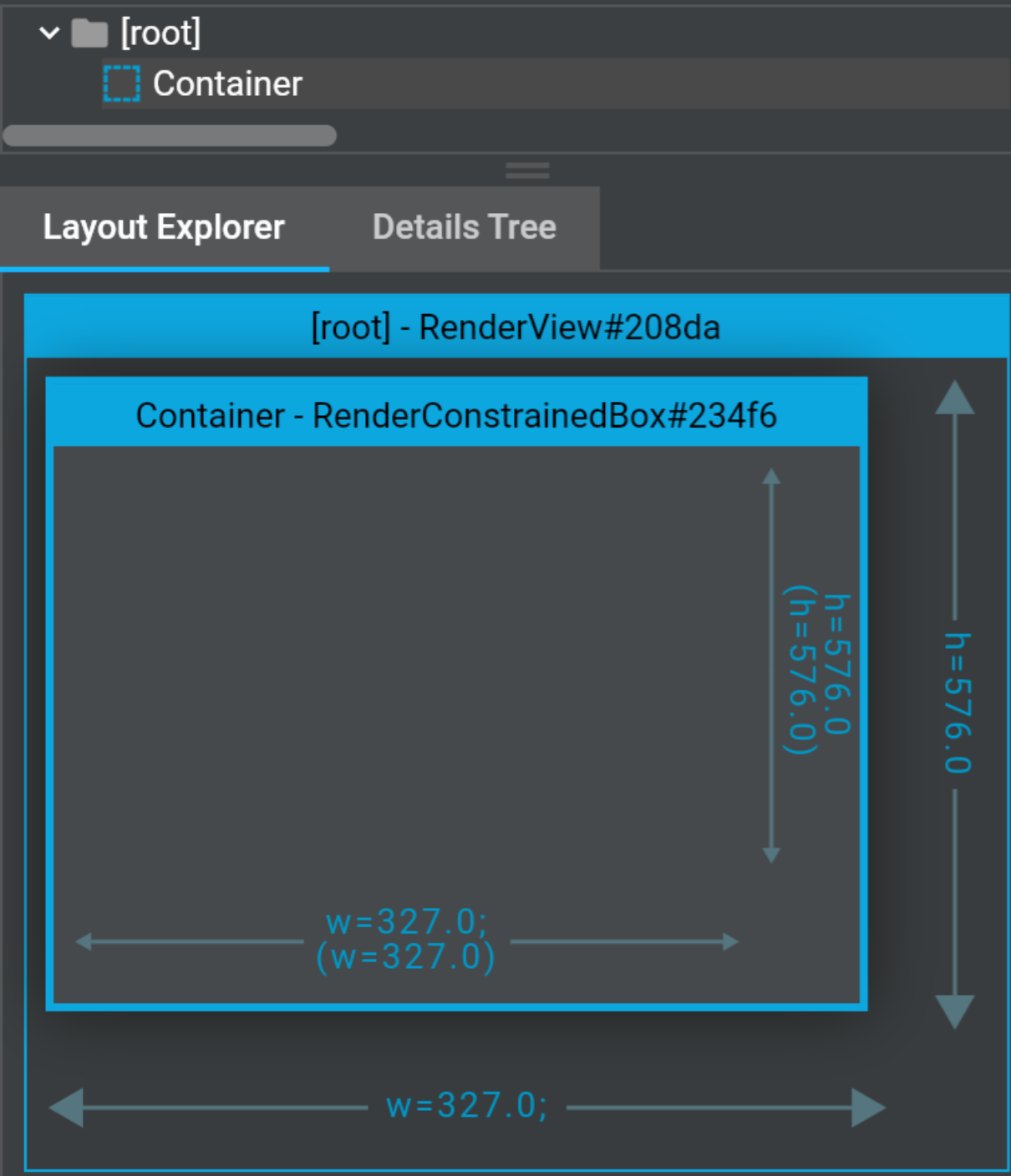
布局的话题展开说，就是各种布局方案+具体组件的使用。

但是约束布局是核心，这也是本文的侧重。

让子元素尽可能的大，撑满父元素

```
1 void main(List<String> args) {
2   runApp(build());
3 }
4
5 Widget build() {
6   return Container(
7     width: 200,
8     height: 200,
9     color: Colors.amber,
10  );
11 }
```

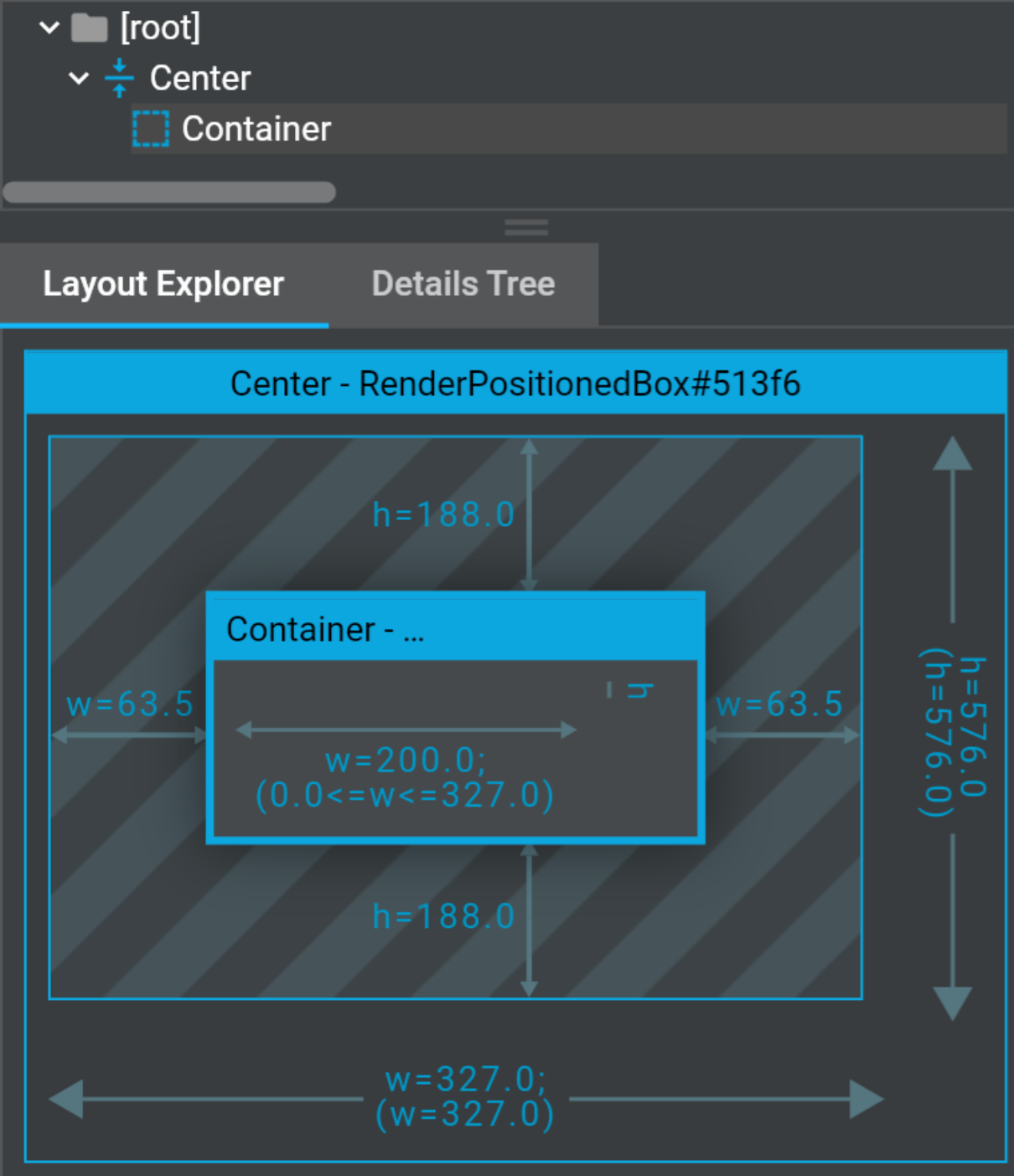




确认位置后，按子元素大小显示

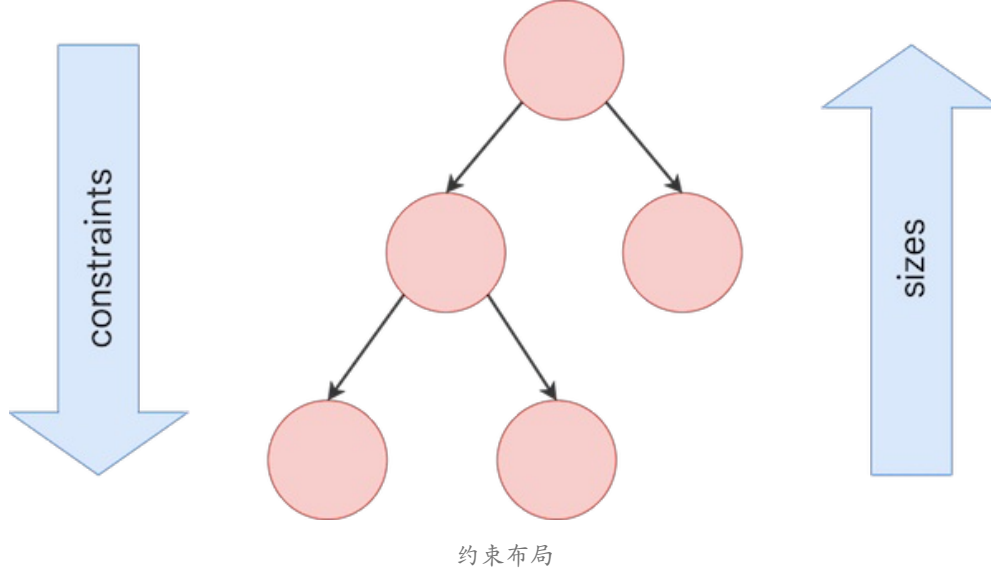
```
1 void main(List<String> args) {  
2   runApp(build());  
3 }  
4  
5 Widget build() {  
6   return Center(  
7     child: Container(  
8       width: 200,  
9       height: 200,  
10      color: Colors.amber,  
11    ),  
12  );  
13 }
```



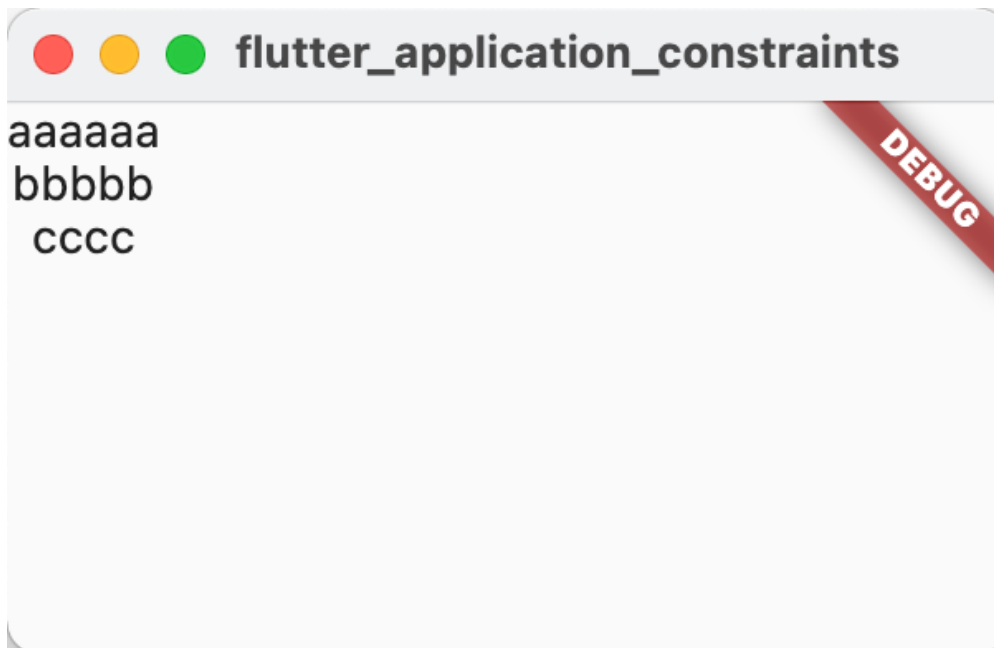


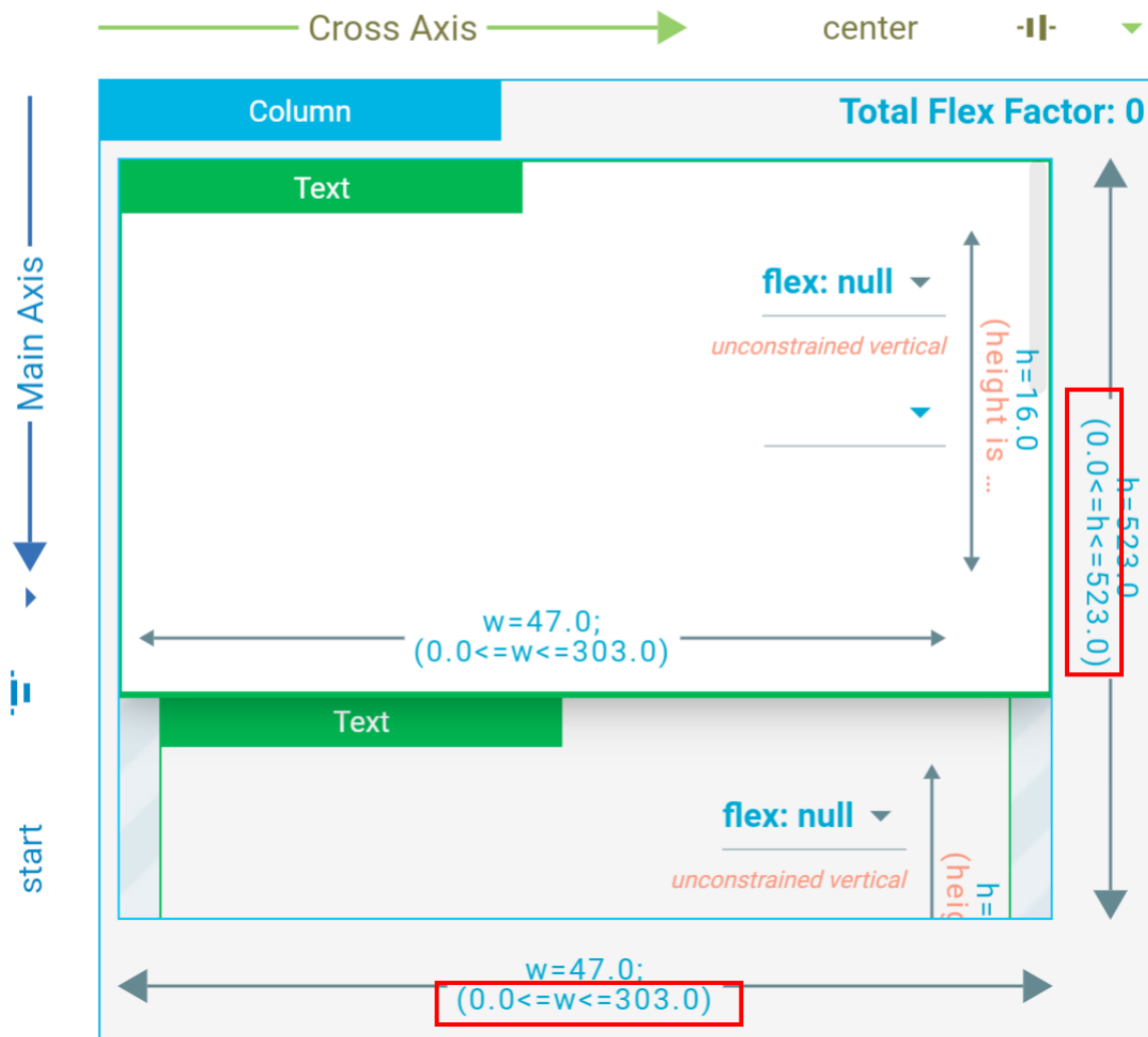
核心规则：Constraints go down. Sizes go up. Positions are set by parents.

- 上层 widget 向下层 widget 传递约束条件。
- 下层 widget 向上层 widget 传递大小信息。
- 上层 widget 决定下层 widget 的位置。

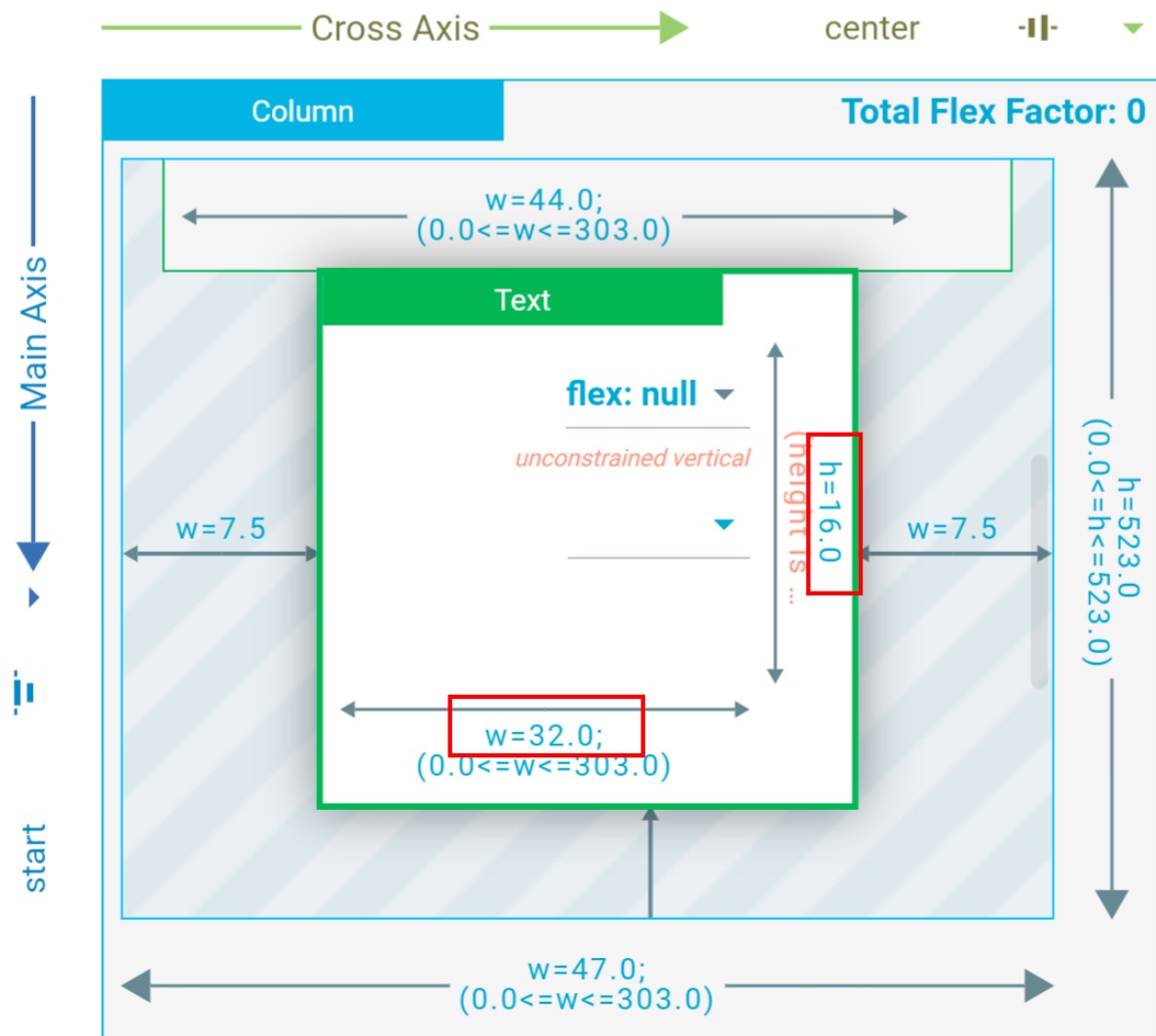


```
1 Widget _buildScaffold() {  
2   return Scaffold(  
3     body: Column(  
4       children: const <Widget>[  
5         Text("aaaaaa"),  
6         Text("bbbbbb"),  
7         Text("cccc"),  
8       ],  
9     ),  
10  );  
11 }
```

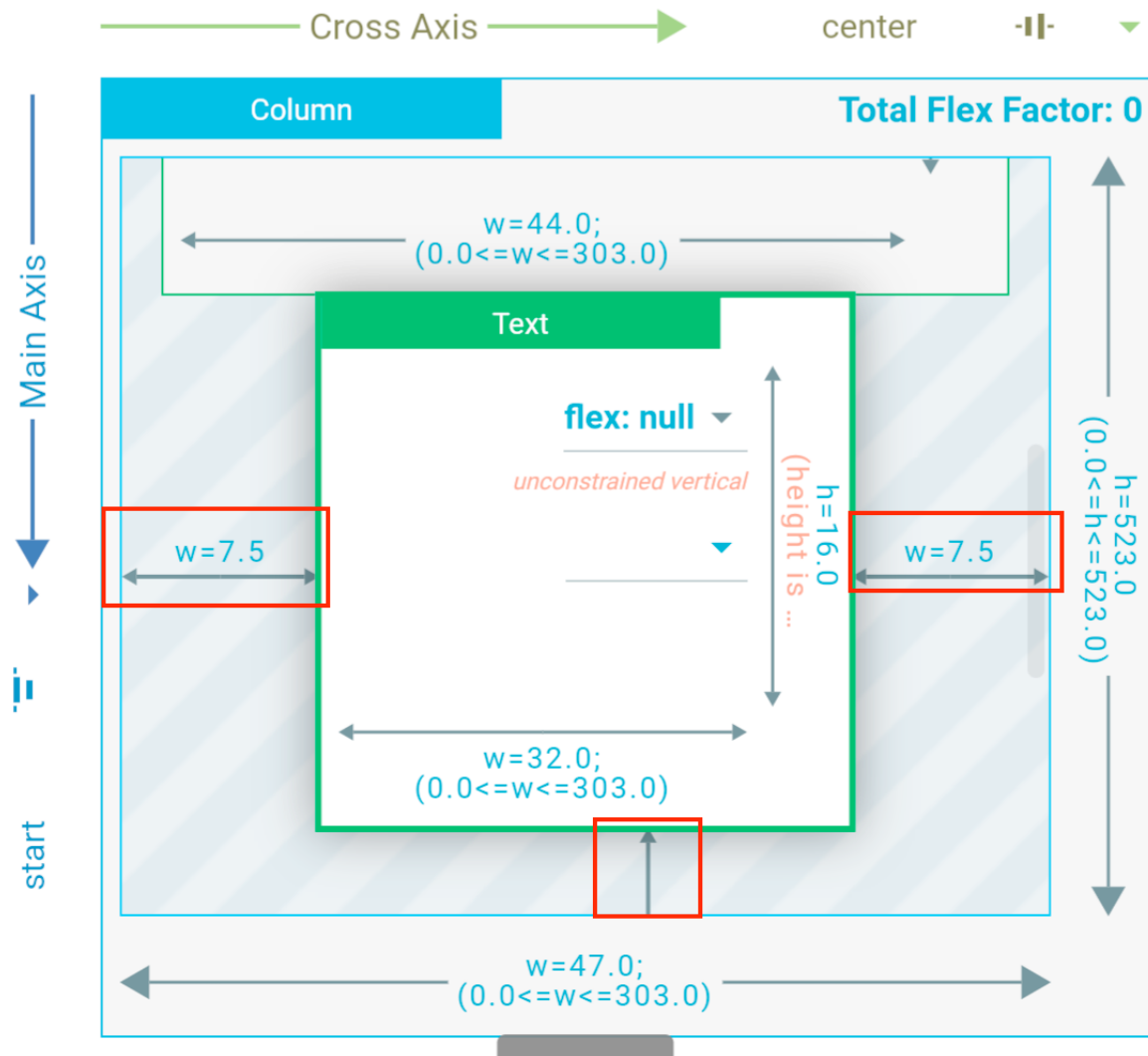




宽度 $0.0 \leq w \leq 303.0$, 高度 $0.0 \leq h \leq 523.0$, 就是上层传下来的约束



宽度 `w=32.0` , 高度 `h=16.0` 就是组件向上层传递的大小信息



元素左边 `w=7.5`，右边 `w=7.5`，就是上层决定下层的组件位置

紧约束、松约束

- 紧约束 tight

它的最大/最小宽度是一致的，高度也一样。

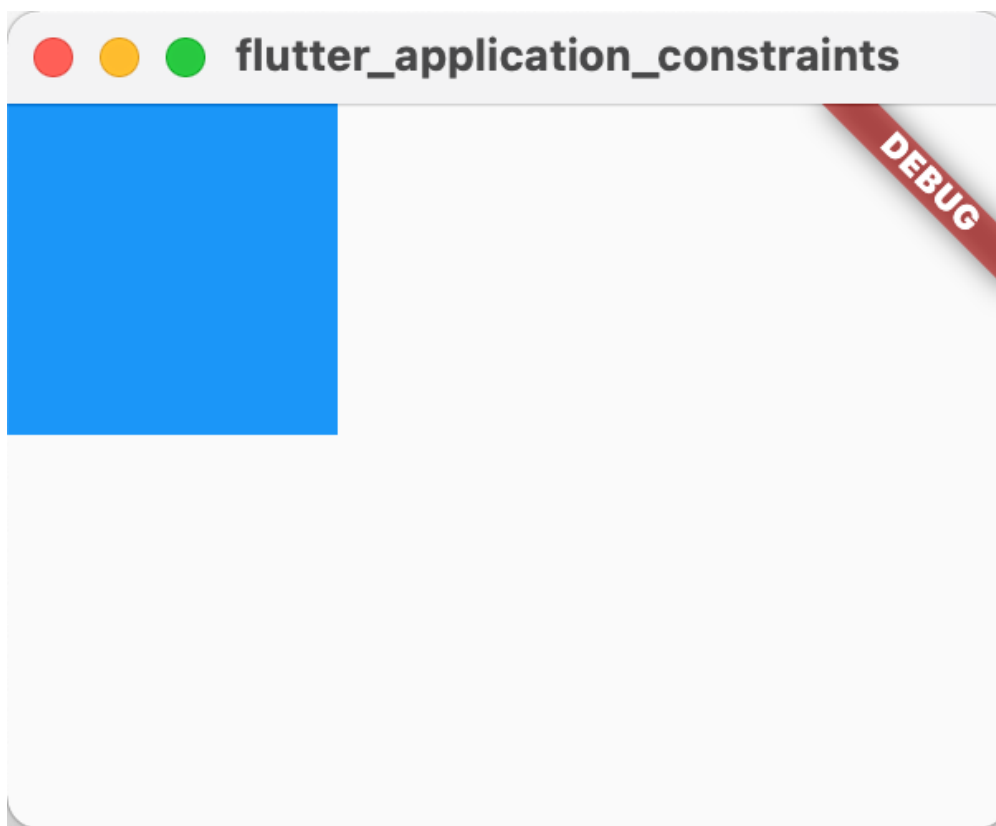
- 松约束 loose

最小宽度/高度为 0

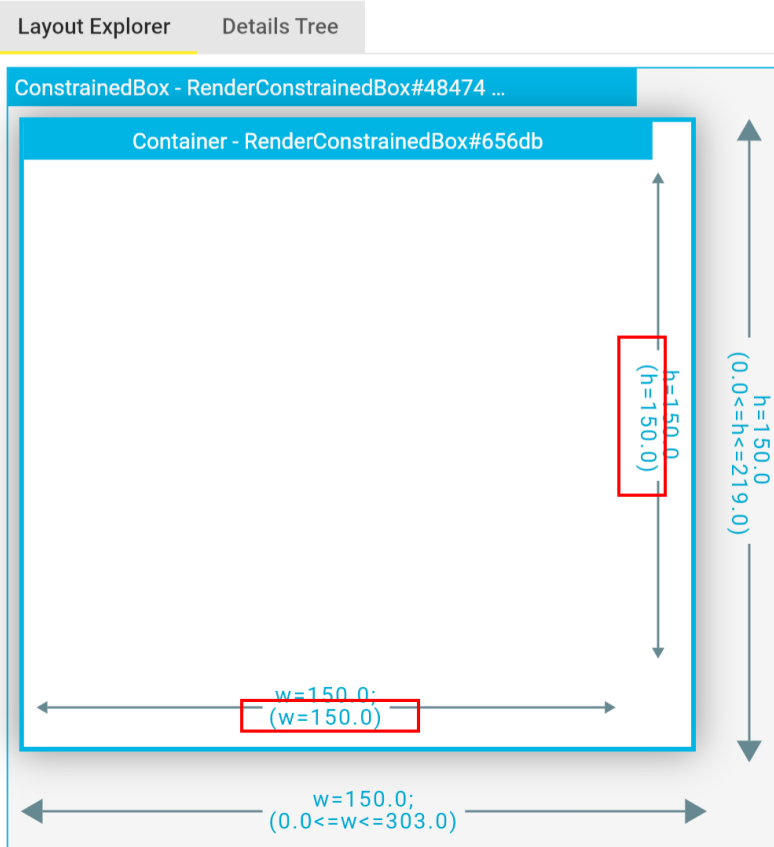
- 同时是紧约束、松约束

如果最大最小都是 0

```
1 Widget _buildScaffold() {  
2   return Scaffold(  
3     body: ConstrainedBox(  
4       constraints: const BoxConstraints(  
5         minWidth: 100,  
6         minHeight: 100,  
7         maxWidth: 150,  
8         maxHeight: 150,  
9       ),  
10    child: Container(  
11      width: 10,  
12      height: 10,  
13      color: Colors.blue,  
14    ),  
15  ),  
16 );  
17 }
```

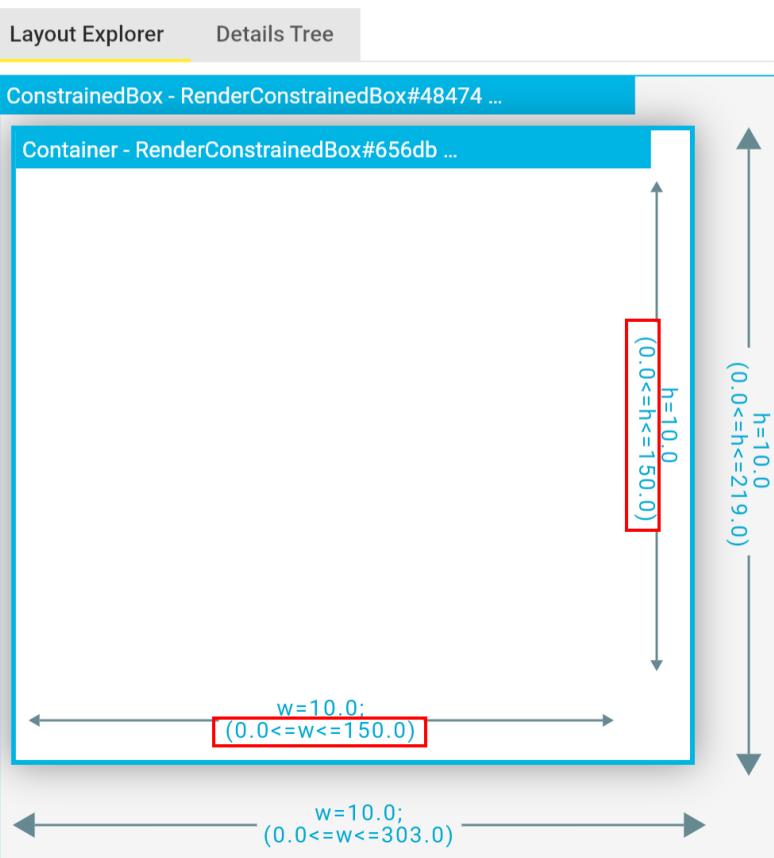


- ▼ [root]
- ▼ MyApp
- ▼ MaterialApp
- ▼ Scaffold
- ▼ ConstrainedBox
- Container



当宽 $150 \leq w \leq 150$, 高 $150 \leq h \leq 150$, 最大/最小宽度是一致的情况, 称为紧约束

- ▼ [root]
- ▼ MyApp
- ▼ MaterialApp
- ▼ Scaffold
- ▼ ConstrainedBox
- Container

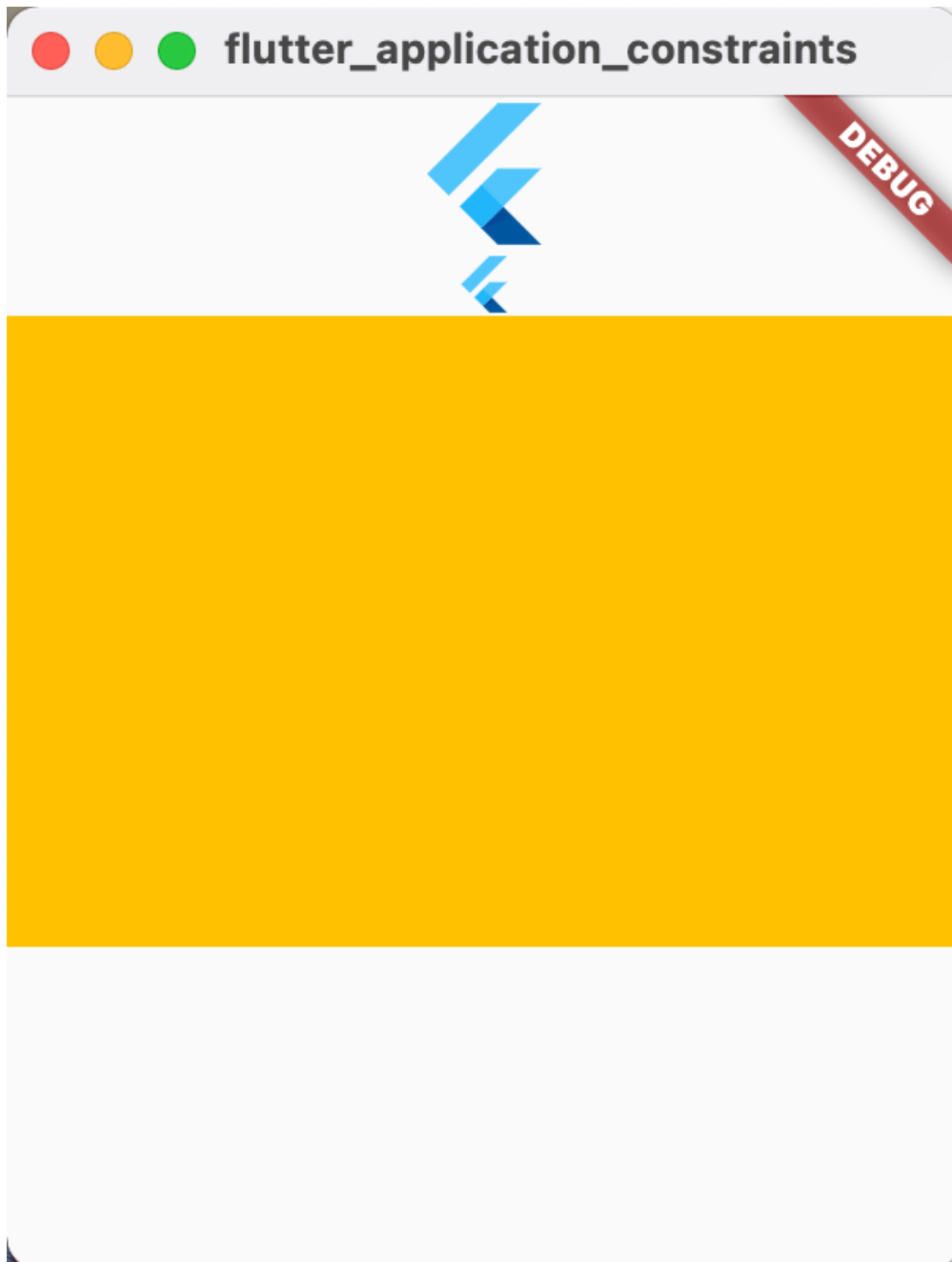


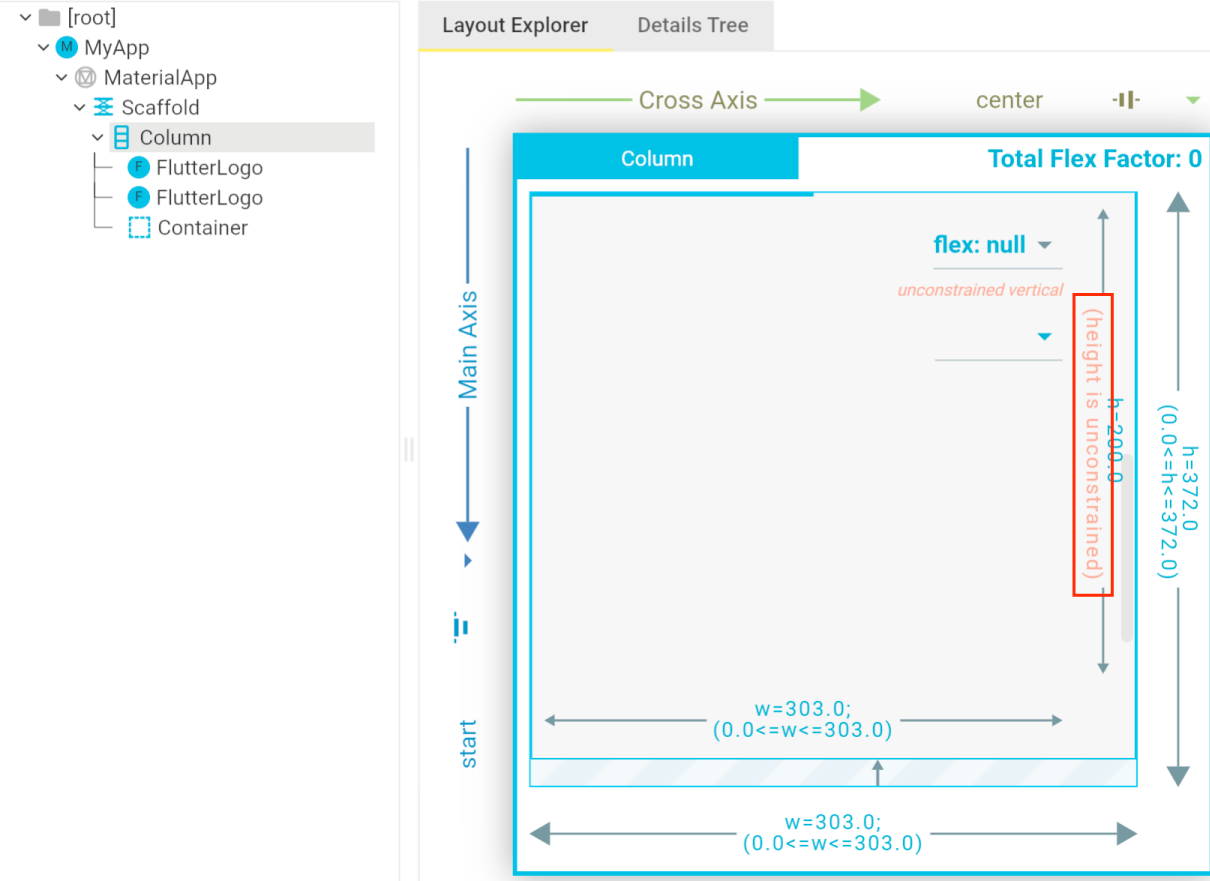
最小宽度/高度为 0 时, 称为松约束

有边界 bounded、无边界 unbounded

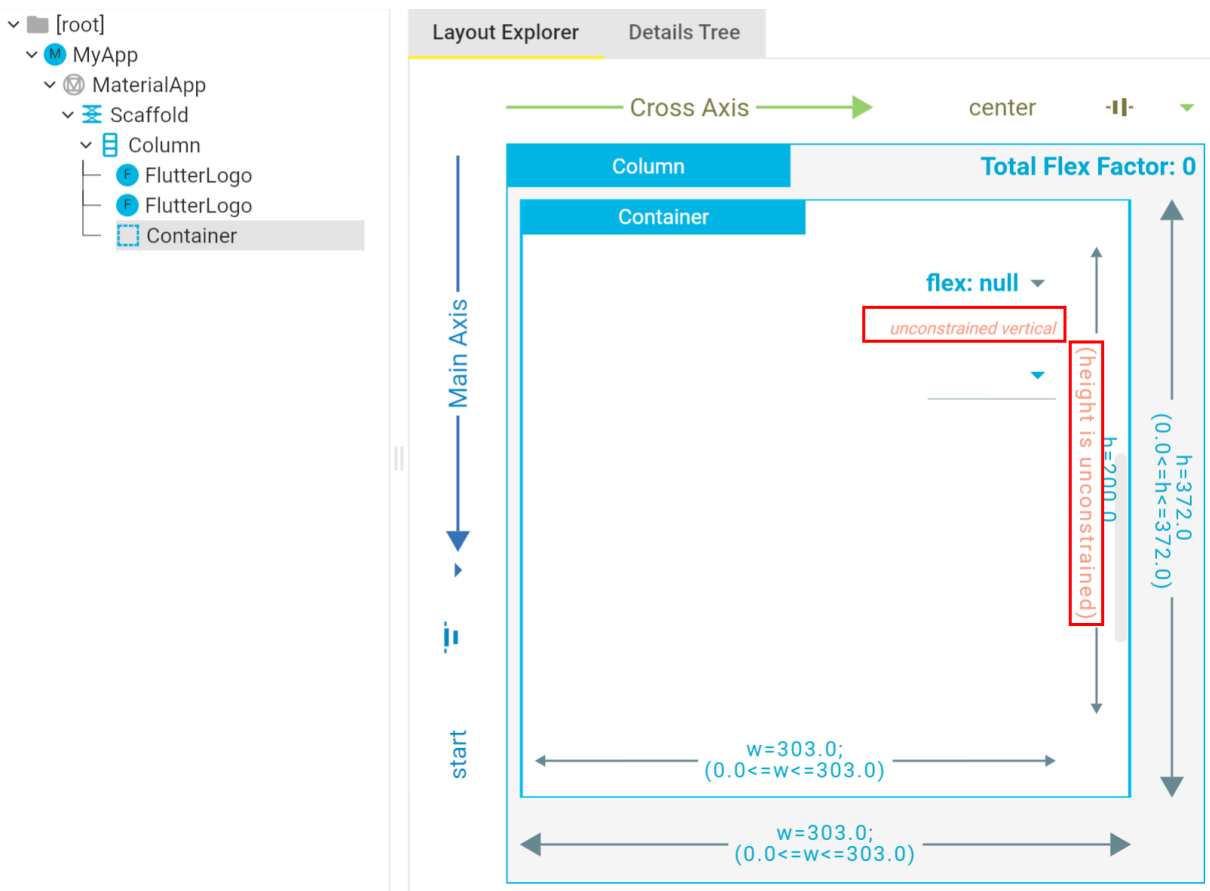
Row Column ListView 这种组件 属于 unbounded

```
1 Widget _buildScaffold() {  
2   return Scaffold(  
3     body: Column(  
4       children: [  
5         const FlutterLogo(size: 50),  
6         const FlutterLogo(size: 20),  
7         Container(  
8           height: 2000,  
9           color: Colors.amber,  
10        ),  
11      ],  
12    ),  
13  );  
14 }
```





我们可以发现 `height unconstrained` 不受限制的，这种就是无边界

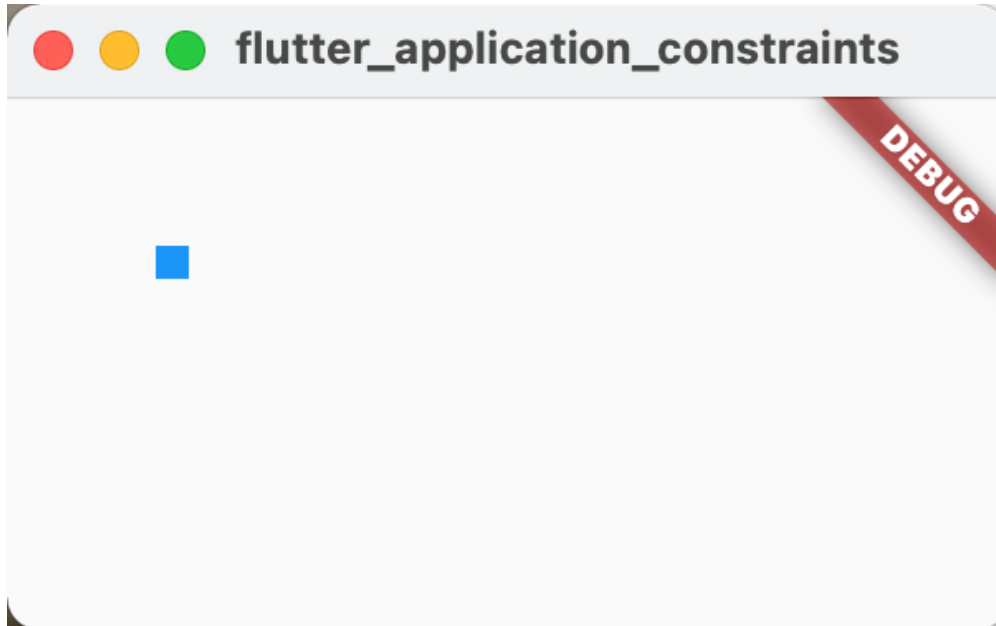


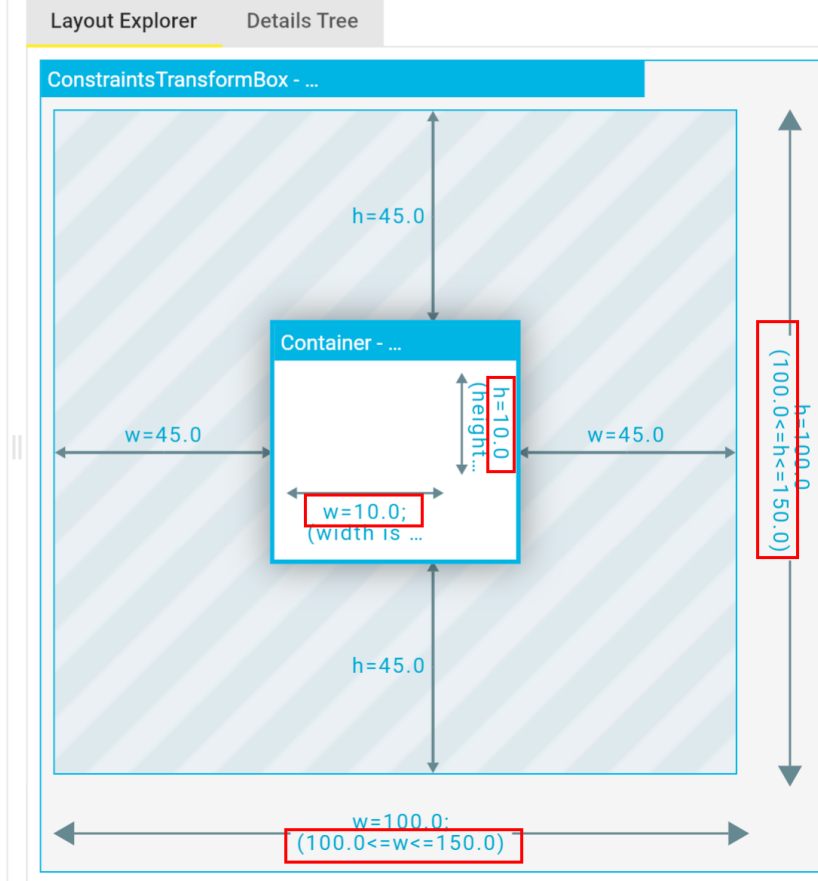
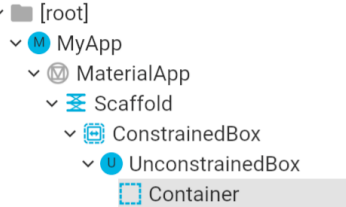
再看子元素在垂直方向高度是不受限制的

UnconstrainedBox 不受约束

`UnconstrainedBox` 可以不受约束自己控制大小

```
1 Widget _buildScaffold() {
2   return Scaffold(
3     body: ConstrainedBox(
4       constraints: const BoxConstraints(
5         minWidth: 100,
6         minHeight: 100,
7         maxWidth: 150,
8         maxHeight: 150,
9       ),
10    child: UnconstrainedBox(
11      child: Container(
12        width: 10,
13        height: 10,
14        color: Colors.blue,
15      ),
16    ),
17  ),
18 );
19 }
```





可以发现外部约束是, 宽 $100.0 \leq w \leq 150$, 高 $100.0 \leq h \leq 150$, 但是 `UnconstrainedBox` 不受约束影响, 但是看起来没有左对齐, 而是居中了, 我们可以通过 `Align` 来调整位置。

总结

今天我们就把布局很核心的约束、限制规则讲了, 大家在布局的时候还是要多思考。

如果你的布局代码写的很复杂, 就要去思考重构了。

当然了业务需要是另外一回事, 尽量让布局引擎默认规则来处理, 这样兼容性好。

© 猫哥

- ducafecat.tech
- [github](https://github.com)
- [b 站](https://www.bilibili.com)

<https://ducafecat.tech/img/banner-gzh.png>

邮箱 ducafecat@gmail.com / 微信 [ducafecat](#) / 留言板 [disqus](#)

猫哥

⌕ Prev

Next ⌕

Flutter 最好的导航插件

最佳的 VS Code Flutter 开发插件

^^ · This blog is licensed under a [Creative Commons Attribution 4.0 International License](#).

猫哥 © 2018 - 2022 · [京ICP备2021009050号](#)
Power by [Hexo](#) Theme [indigo](#) · 网站备案号: 京ICP备2021009050号-1