

一、简介

1、如何约束大小

2、规则

二、BoxConstraints的种类

1、Tightly BoxConstraints (严格约束)

2、Loose Constraints (松散约束)

3、Bounded Constraints(有界约束)

4、UnboundedConstraints(无界约束)

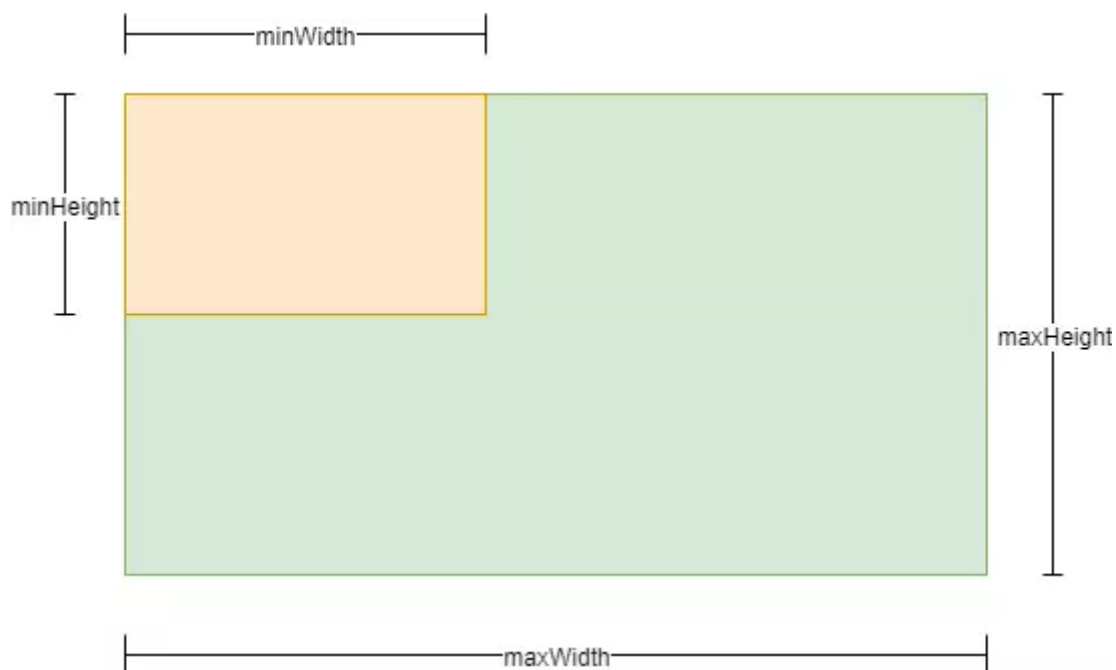
5、Infinite Constraints(无限约束)

一、简介

BoxConstraints即盒约束，用于指定Widget大小的约束

Flutter中的Widget都是通过BoxConstraints来约束大小，这一点类似于Android中的wrap_content和match_parent来约束大小

1、如何约束大小



以左上角为原点，建立x和y轴，通过x和y的最小和最大值四个值来确定。

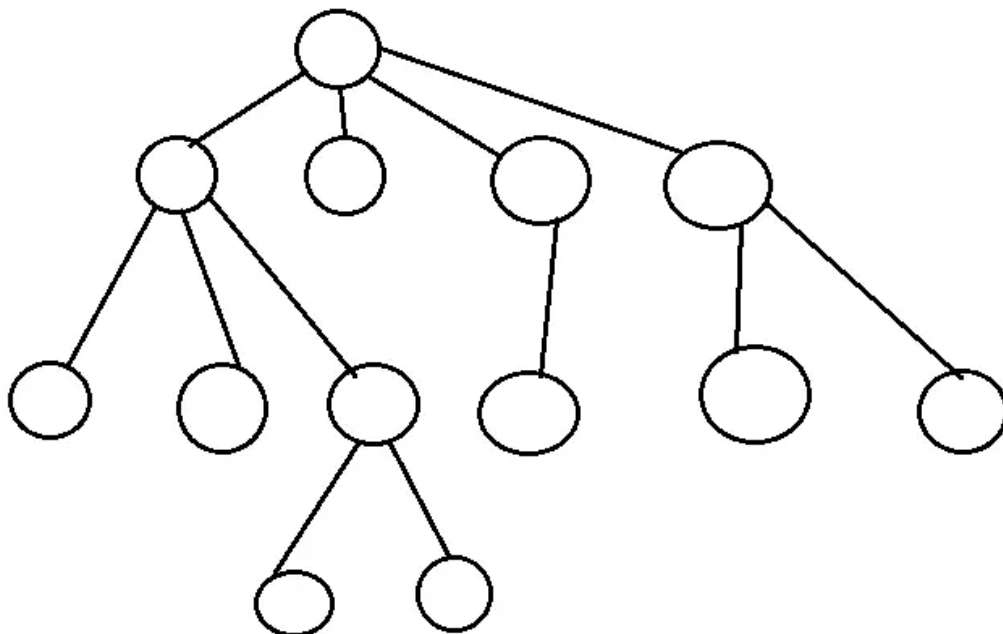
- minWidth
- maxWidth
- minHeight
- maxHeight

$0.0 \leq \text{minWidth} \leq \text{Widget宽度实际大小} \leq \text{maxWidth} \leq \text{double.infinity}$

$0.0 \leq \text{minHeight} \leq \text{Widget的高度实际大小} \leq \text{maxHeight} \leq \text{double.infinity}$

通过minWidth和minHeight形成的最小大小矩形和maxWidth和maxHeight形成的最大矩形，而Widget的实际大小范围就是不小于min矩形不大于max矩形

2、规则



(1) 父Widget将自己的BoxConstraints传递给子Widget，直至传递到叶Widget，因此子Widget都会拥有父Widget的BoxConstraints

(2) 叶Widget会根据BoxConstraints和自己的BoxConstraints计算出自己的大小，并将大小返回给父Widget

(3) 父Widget得到子Widget的大小后，根据子Widget的大小、以及父Widget的boxConstraints和自己的BoxConstraints,通过计算出自己的大小，然后依次传递至root widget

因此Widget的大小受到父Widget的boxConstraints和其子Widget的大小的影响

二、BoxConstraints的种类

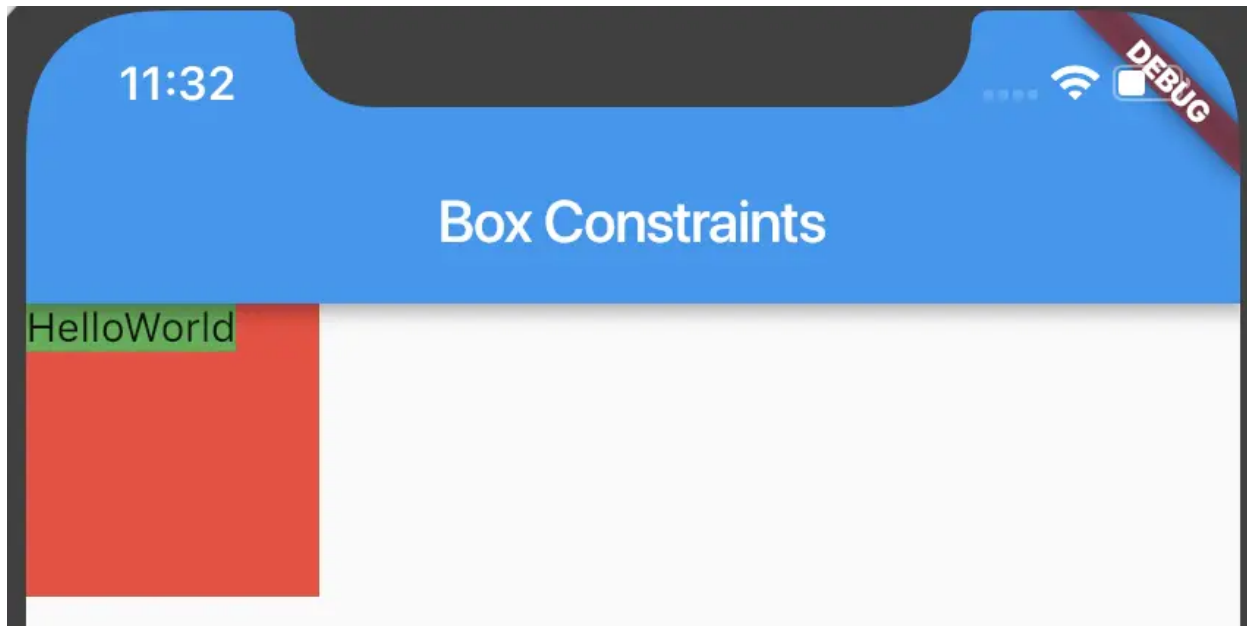
1、Tightly BoxConstraints (严格约束)

当某一轴上最小值和最大值相等时，那么该轴上对应组件的大小就确定了

- 当minWidth= maxWidth，则width就确定了
- 当minHeight= maxHeight，则height就确定了
- 当minWidth= maxWidth且minHeight= maxHeight，则width和height就确定了

代码：

```
1
2 class MaterialBoxConstraintsApp extends StatelessWidget {
3   @override
4   Widget build(BuildContext context) {
5     Paint paint = Paint();
6     paint.color = Colors.green;
7     // TODO: implement build
8     return MaterialApp(
9       title: "Test App",
10      home: Scaffold(
11        appBar: AppBar(
12          title: Text("MaterialTestApp App"),
13        ),
14        body: Container(
15          constraints: BoxConstraints.tight(Size(100, 100)),
16          color: Colors.red,
17          child: Text(
18            "Hello world",
19            style: TextStyle(background: paint),
20          ),
21        ),
22      ),
23    );
24  }
25 }
```



规则如下：

当 Widget 是 **Tightly Constraints** 时，它的宽或高是固定的。

2、Loose Constraints（松散约束）

当某一轴最小值为0即为松散约束

若该轴为x轴，

- 宽度最大值为确定值时，如果子Widget的宽度小于最大值，则宽度为子Widget的宽度，如果子Widget的宽度大于最大值，则为最大值

```
1 body: Container(  
2   constraints: BoxConstraints.loose(Size(100, 100)), //添加 Loose Constraints  
3   color: Colors.red,  
4   child: Text(  
5     "HelloWorld",  
6     style: TextStyle(background: paint),  
7   ));
```

这里使用loose (maxWidth,maxHeight)

- 宽度最大值为infinite，则宽度为子Widget的宽度

```
1 body: Container(  
2   constraints: BoxConstraints.tightFor(), //添加 Loose Constraints  
3   color: Colors.red,  
4   child: Text(  
5     "HelloWorld",  
6     style: TextStyle(background: paint),  
7   ));
```

这里使用BoxConstraints.tightFor(), 表示宽高都是最大值

- 若没有子Widget, 且宽度最大值为infinite, 则宽度为infinite
高度也是如此

```
1 body: Container(  
2   constraints: BoxConstraints.tightFor(), //添加 Loose Constraints  
3   color: Colors.red,));
```

规则如下:

当 Widget 是 Loose Constraints 时, 无论最大值是确定值还是无限制, 如果有子Widget, 那么 Widget 的大小就是 子Widget 的大小, 相当于 wrap_content, 如果没有 子Widget, 那么 Widget 就是最大值, 如果这个最大值是 Infinite,就相当于 match_parent

3、Bounded Constraints(有界约束)

当某一轴的最大值是确定的值时, 就是有界约束

规则如下:

当 Widget 是 Bounded Constraints 时,如果有 子Widget, 子Widget 的大小小于有界约束的最小值, 则显示的是有界约束的最小值, 而当 子Widget 的大小大于有界约束的最小值, 小于有界约束的最大值, 则显示的是 子Widget 的大小, 否则显示的是有界约束的最大值; 如果没有 子Widget 就显示有界约束的最大值

```
1 body: Container(  
2   constraints: BoxConstraints(minWidth: 100,maxWidth: 300,minHeight: 0,max  
   Height: 300), //添加 Bounded Constraints  
3   color: Colors.red,  
4   child: Text(  
5     "HelloWorld",  
6     style: TextStyle(background: paint),  
7   ));
```

4、UnboundedConstraints(无界约束)

当某一个轴上最大值是无限值时即无界约束

规则如下:

其实 Unbounded Constraints 很像 Loose Constraints。当 Unbounded Constraints 有 子Widget 时, 若 子Widget 的大小小于 Unbounded Constraints 的最小值时, 则 Widget 显示 Unbounded Constraints 的最小值, 若 子Widget 的大小大于 Unbounded Constraints 的最小值时, 则显示 子Widget 的大小; 若没有 子Widget, 就相当于 match_parent

```
1 body: Container(  
2   constraints: BoxConstraints(minWidth: 100,maxWidth: double.infinity,minH  
   eight: 100,maxHeight: double.infinity), //添加 Unbounded Constraints  
3   color: Colors.red,  
4   child: Text(  
5     "HelloWorld",  
6     style: TextStyle(background: paint),  
7   ))));
```

5、Infinite Constraints(无限约束)

当某一轴的最大值和最小值都是无限值时就是无限约束

规则：

Infinite Constraints 就相当于 match_parent

```
1 body: Container(  
2   constraints: BoxConstraints.expand(), //添加 Infinite Constraints  
3   color: Colors.red,  
4   child: Text(  
5     "HelloWorld",  
6     style: TextStyle(background: paint),  
7   ))));
```