ThoughtWorks®

ThoughtWorks® | IT大咖说 知识分享平台

前端布局从古至今

分享人：赵国旭

分享的内容

ThoughtWorks® | IT大咖说 知识分享平台

前端布局演化历史

CSS1的的黑暗时代，到CSS2的混乱时代，最后到CSS3的美好新生活

布局方案

在CSS2到CSS3时期，广大劳动人民的智慧结晶

选择方案时的思考

针对需求和用户群体，我们需要不同的方案


ThoughtWorks®
IT大咖说
知识分享平台
布局演化历史

黑暗时代

http://www.arngren.net/

CSS1

- width & height
- float & clear
- margins & padding
- background & borders
- fonts
- …

CSS最开始的目的是用来替换一些格式化的标签，

比如<font>和
标签等。

将CSS拿来控制所有页面样式的这种想法在当时还很激进。

黑暗时代


```
<FONT face=Arial color=#0000FF size=6>
<BR>
<CENTER>
<H1>Big Font<H1>
<TABLE width=200 height=300 border=5 bordercolor=navy bgcolor=lightblue>
<TD background=clouds.jpg>
<IMG src=spacer.gif width=4 height=20>
<IMG align=left border=2>
<BR clear=left>
```

那个时代，布局的工作和CSS还没有任何关系。

在当时，大多数布局的工作都是交给table的。

CSS2 Table display

ThoughtWorks® | IT大咖说 知识分享平台

拿table直接做布局的问题

- 拿table直接做布局最大的一个问题就是语义化。
- 性能问题

display: table-cell

简单的两栏布局：

```
.sidebar, .main { display: table-cell; }
```

当时的主流浏览器不支持！

CSS2 布局模型

- **流动模型**

  流动模型是默认网页布局模式，所有块状元素宽度都为100%，独霸一行，垂直分布。

- **浮动模型**

  浮动的框可以左右移动，直到外边缘碰到包含框或另一个浮动框的边缘

- **层模型**

  绝对相对定位，类似于Ps里面图层的概念

CSS2 存在的问题

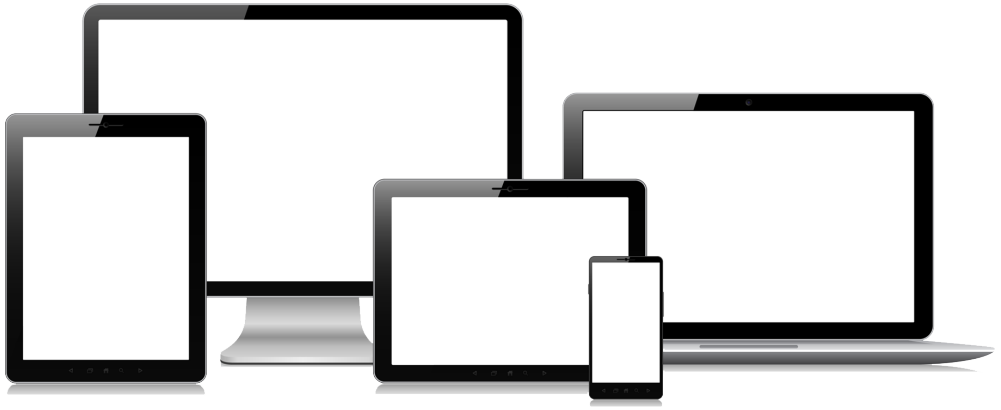
- 难以理解的语法
- 在各个浏览器上的兼容bug，不一样的行为
- 灵活性不高，为了解决问题，通常需要各种hack

什么玩意，又要加班了！

Web 布局中的问题

ThoughtWorks® | IT大咖说 知识分享平台

- 浏览器，字体支持不一样
- 屏幕大小比例，分辨率
- 用户会拖拽改变视口大小
- 用户会缩放字体大小
- 动态内容，导致样式混乱



真的谢谢W3C的叔叔哥哥姐姐阿姨们。

CSS3 布局设计原则

ThoughtWorks® | IT大咖说 知识分享平台

- **灵活(Flexible)** — 适应不同的屏幕字体大小，和任意多的内容
- **功能强大(Powerful)** — 任何设计师的稿纸都能完美的实现。小功能能简单的实现，复杂的也能hold住
- **稳健(Robust)** — 在不被期待的环境下不会坏掉，比如长段落，窄屏等
- **易于理解(Understandable)** — 概念清晰，易于维护
- **高性能(Performant)** — 绘制渲染快

CSS3 布局模型

- Multi-column Layout
- Flexible Box Layout
- Grid Template Layout
- Media Queries

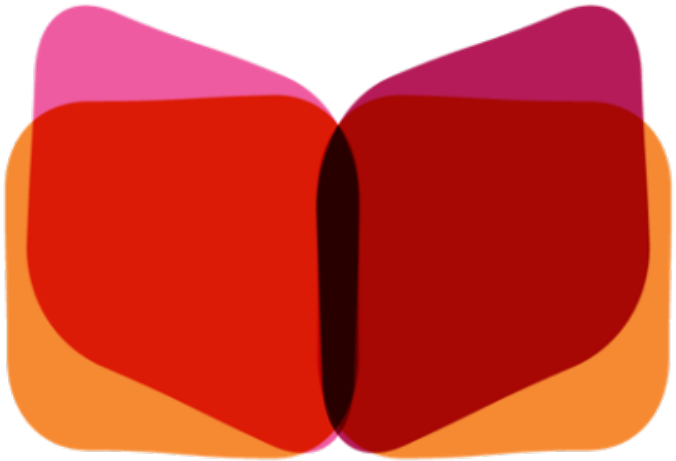
其中Flex box 和CSS grid将会在后面详细介绍。


ThoughtWorks®
IT大咖说
知识分享平台
布局方案

CSS 布局方案

吹了半天牛，是时候来点干货了。

- 传统布局
- CSS table
- 内滚动布局
- 响应式布局
- rem布局
- flex box
- CSS Grid



传统布局

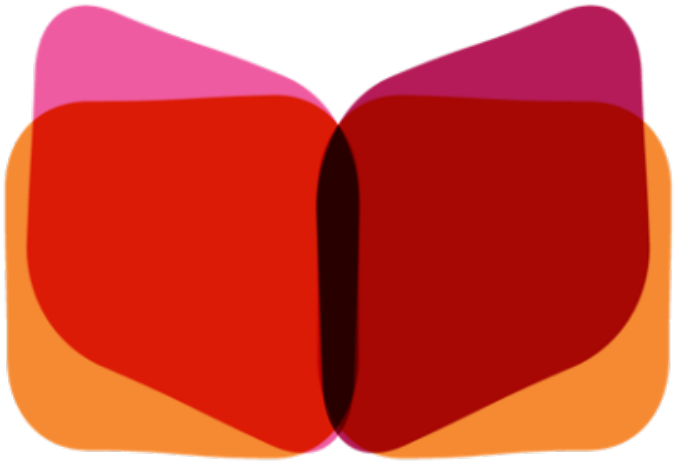
浮动布局

基于浮动模型的一系列布局

使用BFC

BFC就是页面上的一个隔离的独立容器，容器里面的子元素不会影响到外面的元素。反之也如此。

BFC的区域不会与float box重叠。



CSS Table

.

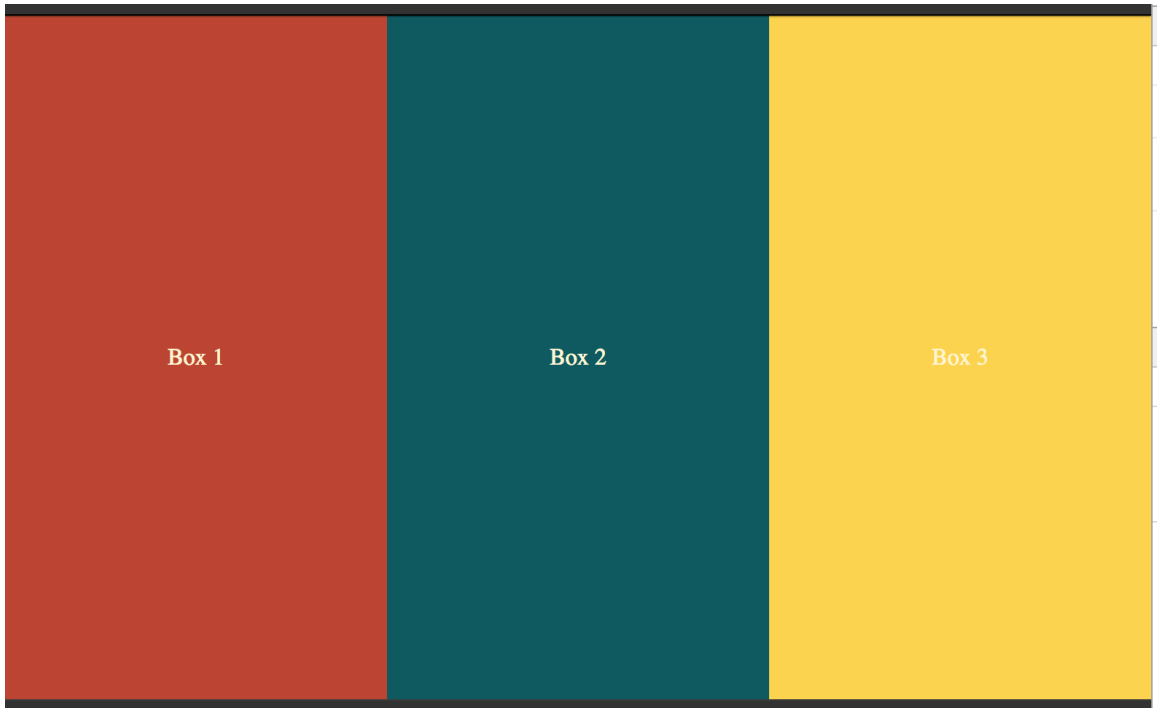
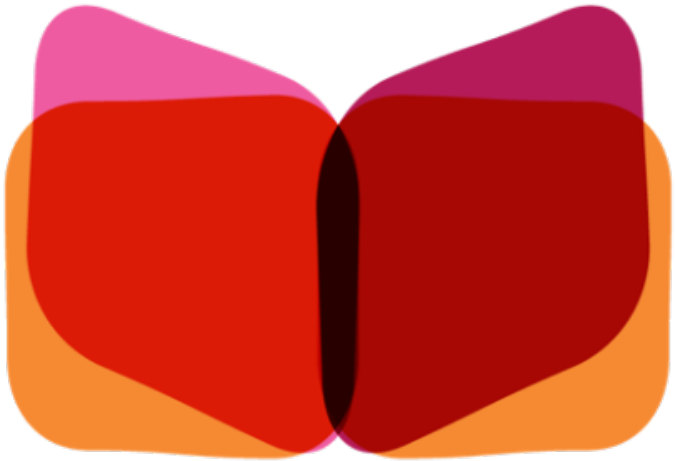


table 的一个table-cell其实就相当于一个弹性盒，他会根据内容，动态的调整大小。也有属性(tabel-layout)控制外部大小。



内滚动布局

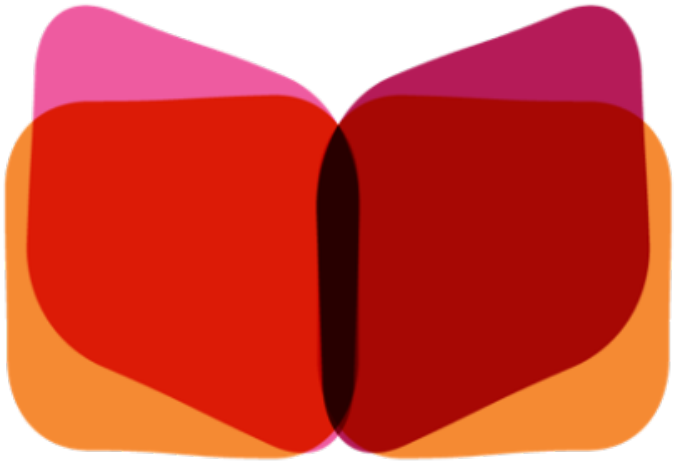
特定需求布局

ThoughtWorks® | IT大咖说 知识分享平台

什么叫做内滚动?

现代web的发展趋势是从以前的瀑布式信息流向PC端应用类型发展。

在这个发展进程中，内滚动布局必然会成为需求之一。



响应式布局

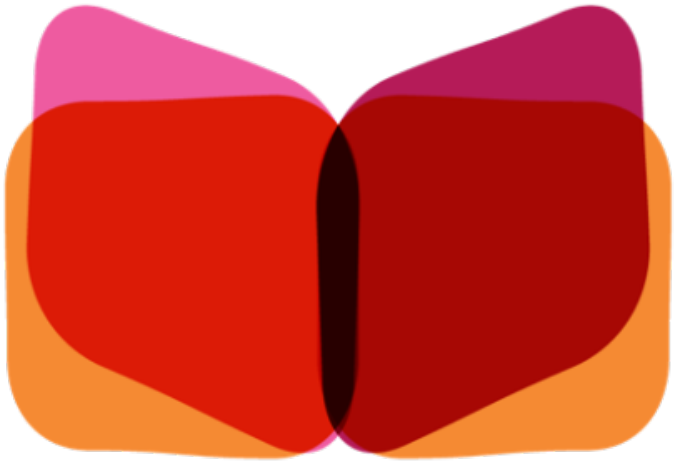
ThoughtWorks® | IT大咖说 知识分享平台

Web布局中的问题：

- 浏览器，字体支持不一样
- 屏幕大小比例，分辨率
- 用户会拖拽改变视口大小
- 用户会缩放字体大小
- 动态内容，导致样式混乱

响应式的解决方案：

- view-port 定义视口
- Media Queries 媒体查询
- 百分比缩放（此处有代码）



REM布局

百分比定义栅栏的问题

- 计算困难
- 容易引发bug

REM 是啥?

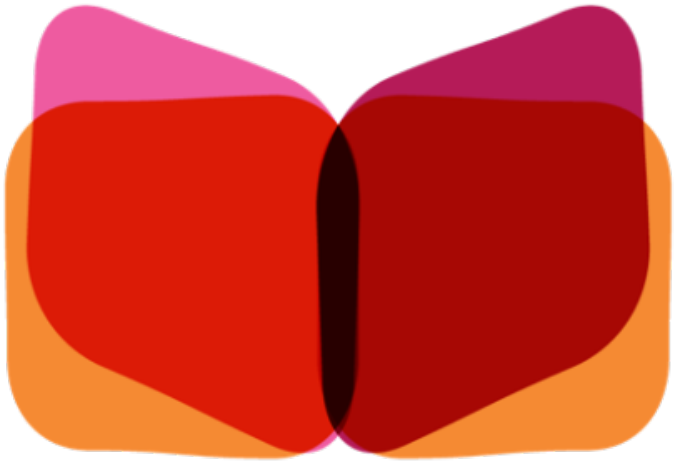
rem是相对于根元素(html)字体大小的单位。

window: 375px

html: font-size: 37.5px

1rem = 37.5px

10rem = 375px

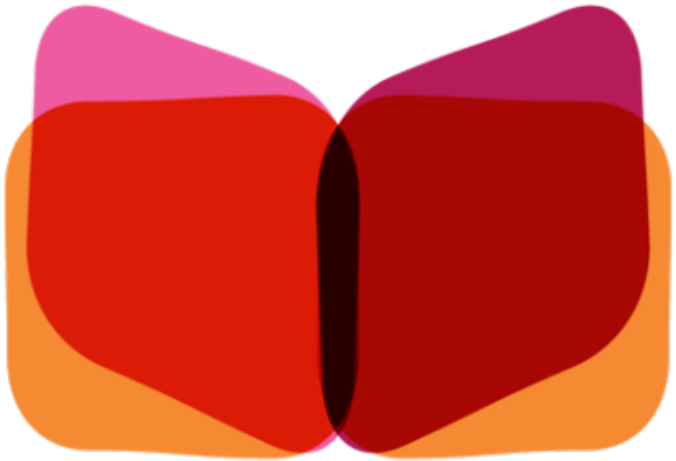


flex box

ThoughtWorks® | IT大咖说 知识分享平台

浮动布局的问题

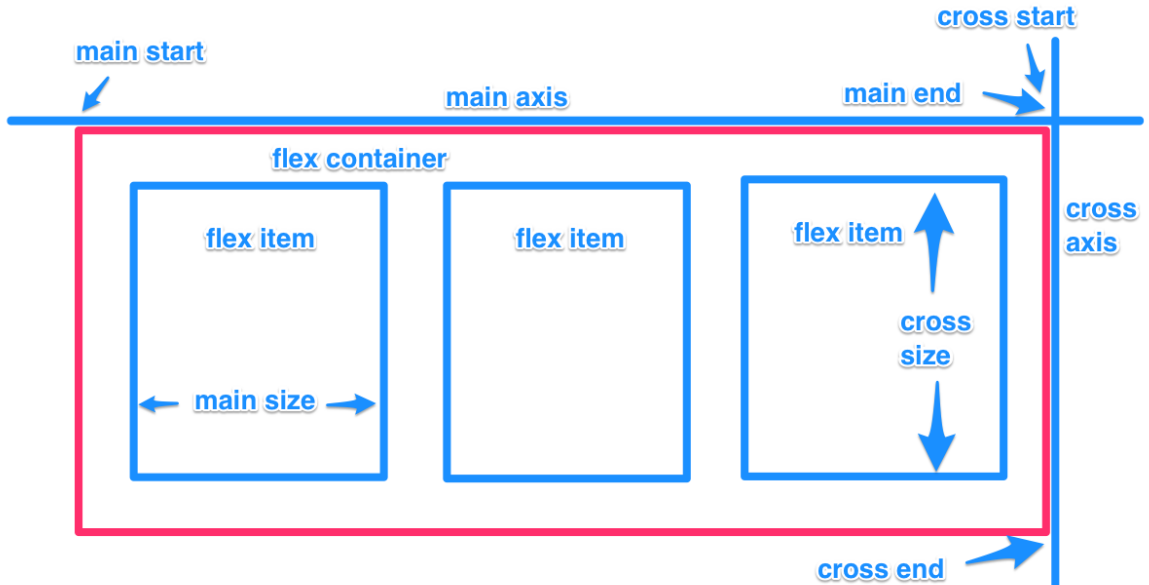
- 难以控制
- 源码顺序依赖
- 列等高问题
- 内容居中

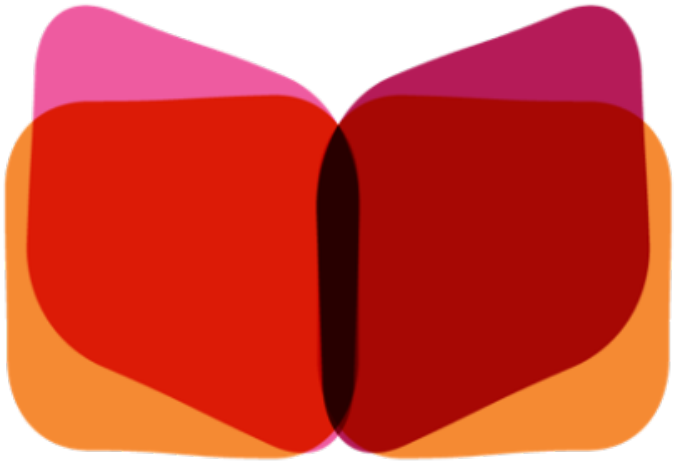


flex box

Flex box 的处理

- 通过将弹性元素拉伸或缩小来充满可用空间和避免溢出。
- 给予弹性元素成比例的尺寸。
- 弹性容器内的弹性元素可以从任意方向布局。





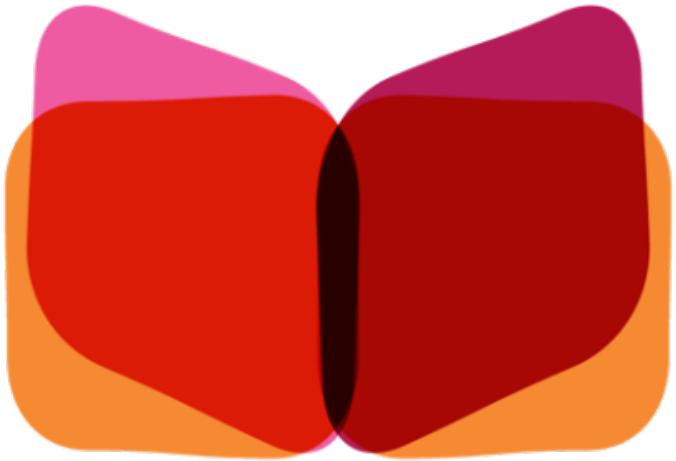
flex box 的属性

弹性盒子的属性

- flex-direction
- flex-wrap
- flex-flow
- justify-content
- align-items
- align-content

弹性元素的属性

- order
- align-self
- flex-grow
- flex-shrink
- flex-basis



flex box 的bug

ThoughtWorks® | IT大咖说 知识分享平台

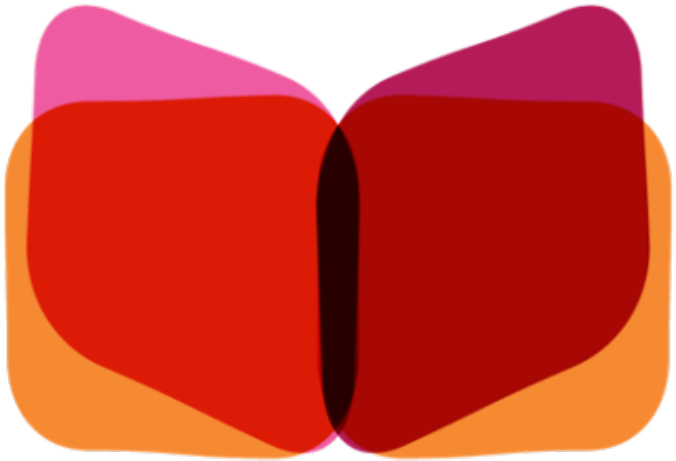
兼容性

Current aligned | Usage relative | Date relative | Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		52	49			9.3		4.4	
	14	53	58		45	10.2		4.4.4	
4 11	15	54	59	10.1	46	10.3	all	56	59
	16	55	60	11	47	11			
		56	61	TP	48				
		57	62						

BUG

- PC端主要集中在IE11
- 移动端主要是国内某U开头的浏览器



CSS grid

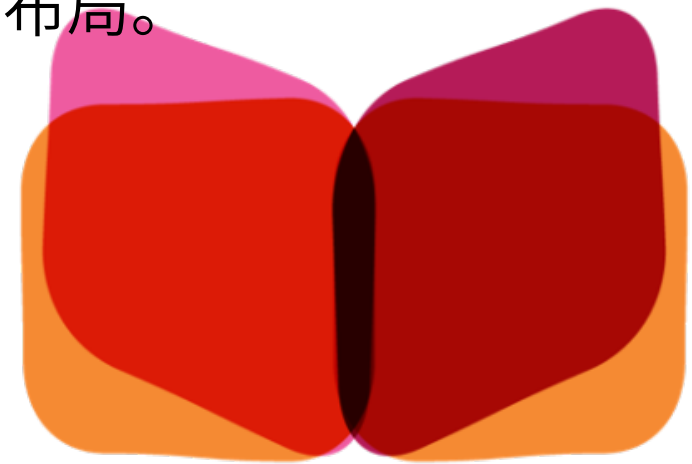
ThoughtWorks® | IT大咖说 知识分享平台

什么是网格布局?

网格布局就是把一个页面划分成一个一个的区域(regions)，并定义他们之间相互关系的一个布局模型。

CSS grid与flex box的比较

他只能在一条直线上放置你的内容区块；而grid是一个二维布局。



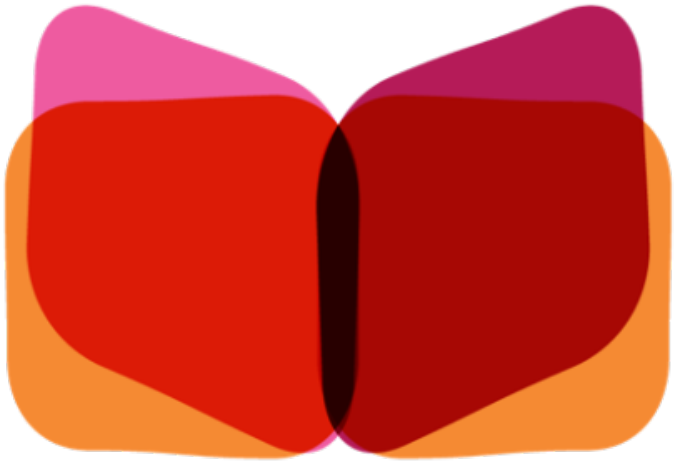
CSS grid 术语

网格容器

类似于flex box，设置为display: flex的元素是弹性盒子模型

display为grid和inline-grid的元素就是网格容器。

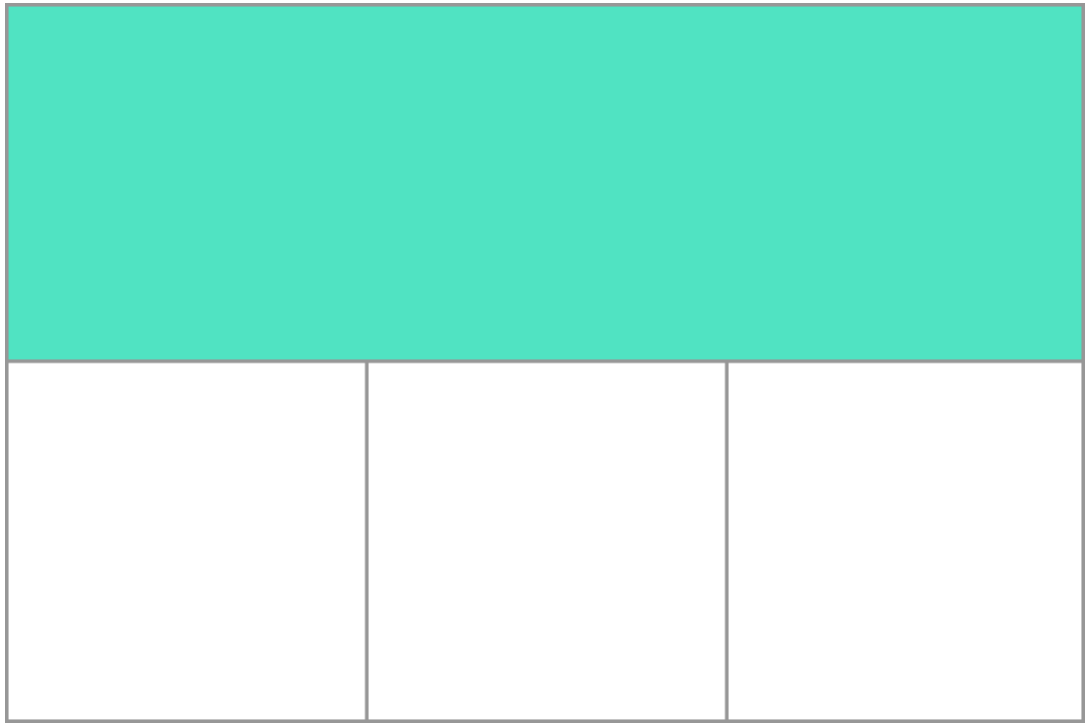

```
.wrapper {
  display: grid;
}
```

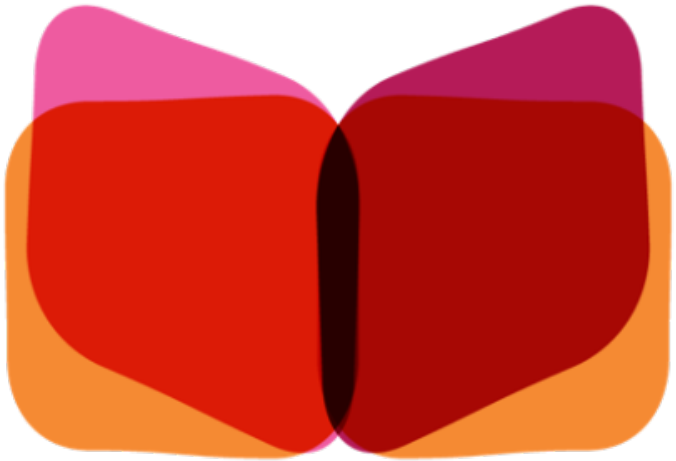


CSS grid 术语

ThoughtWorks® | IT大咖说 知识分享平台

网格轨道track

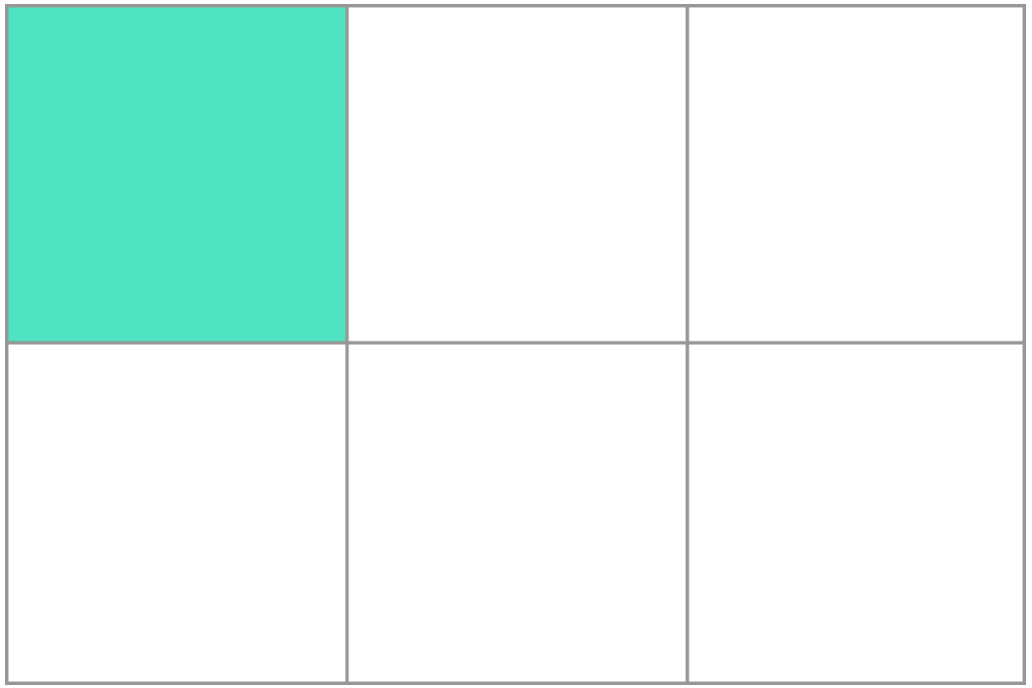


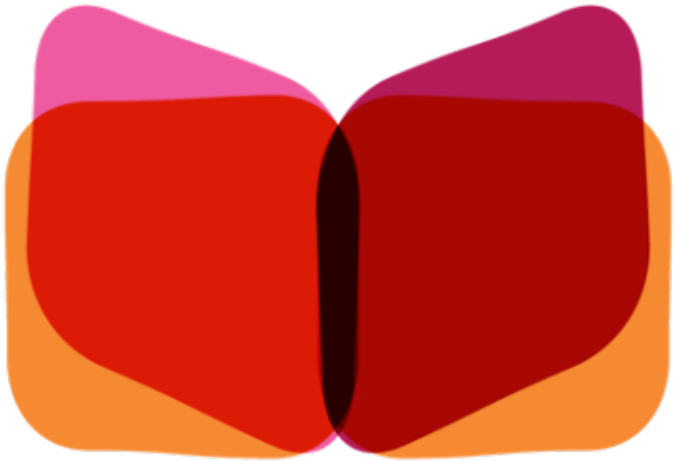


CSS grid 术语

ThoughtWorks® | IT大咖说 知识分享平台

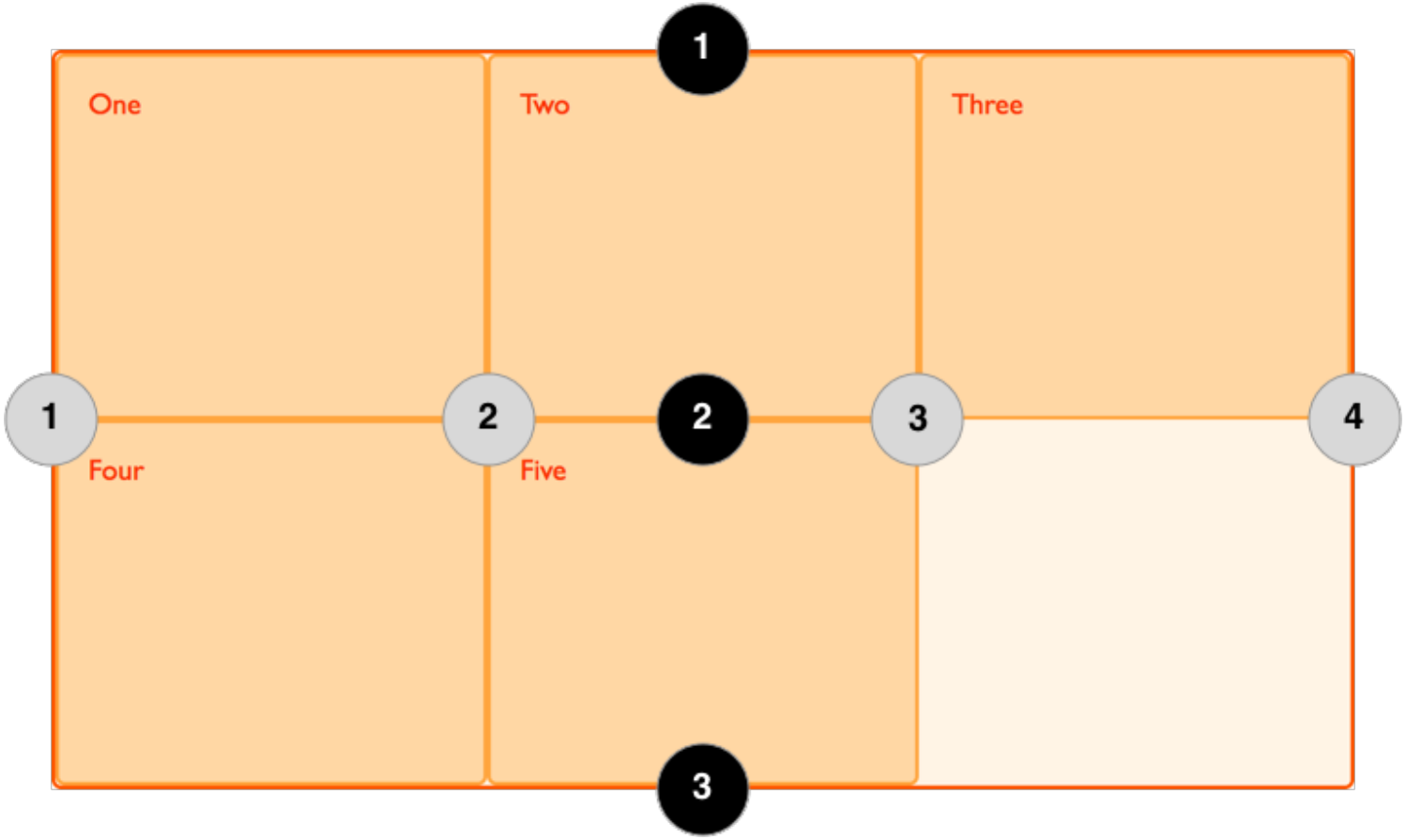
单元格 Cells

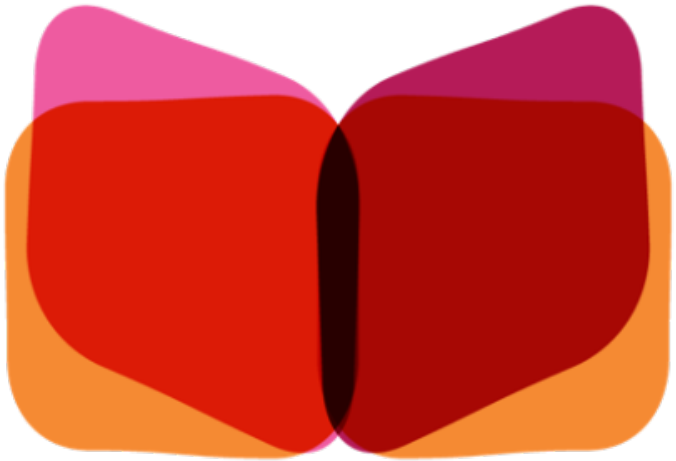




CSS grid 术语

网格线line

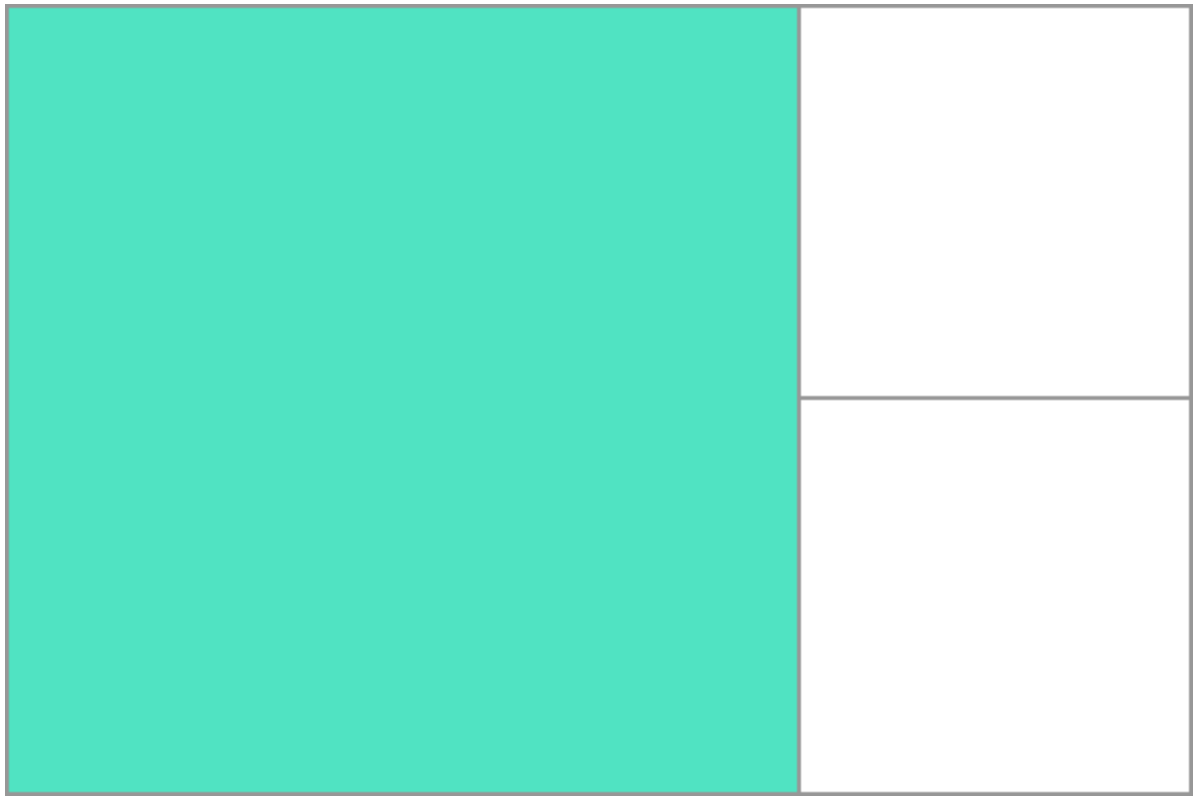


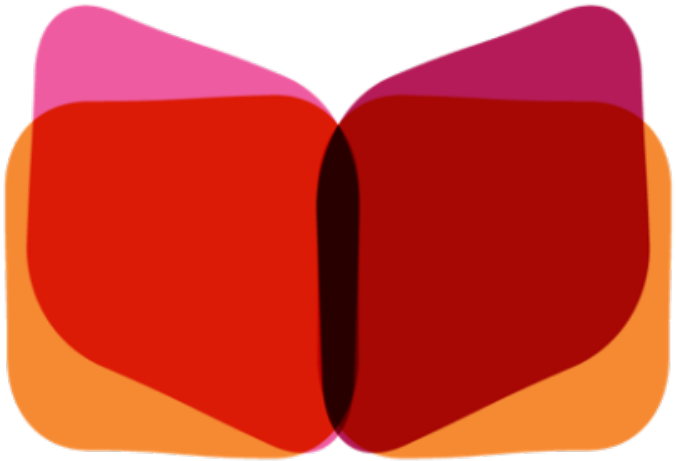


CSS grid 术语

ThoughtWorks® | IT大咖说 知识分享平台

网格区域areas

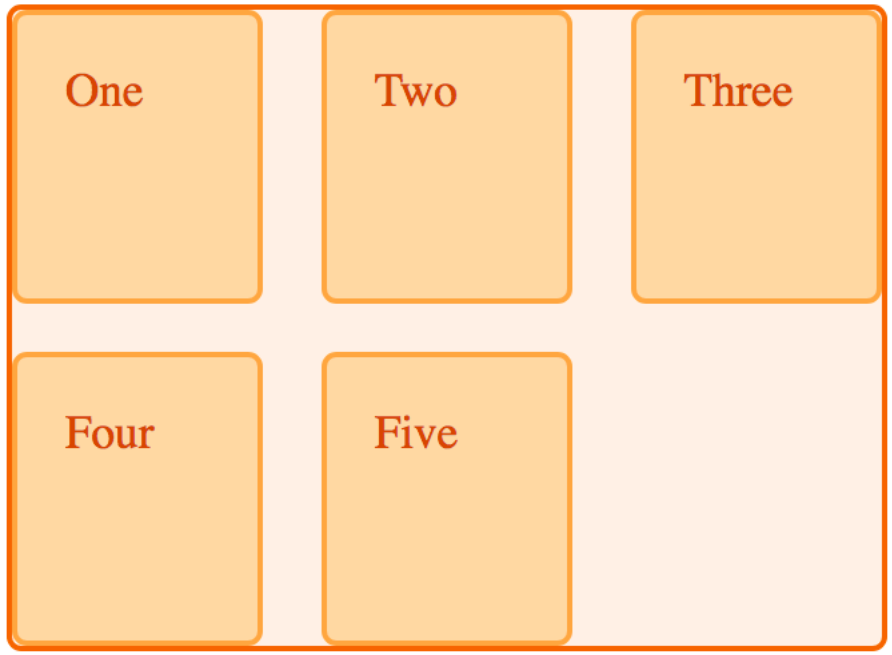


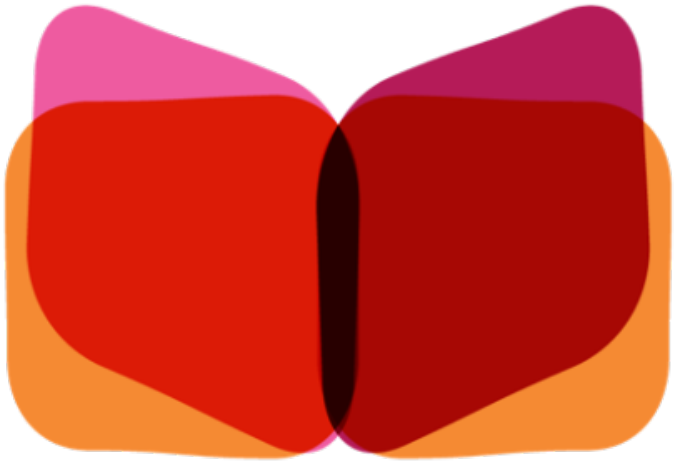


CSS grid 术语

ThoughtWorks® | IT大咖说 知识分享平台

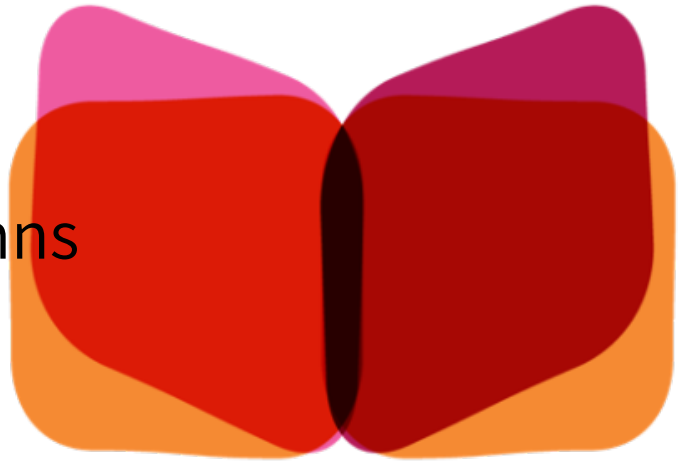
网格间隙





CSS grid 属性

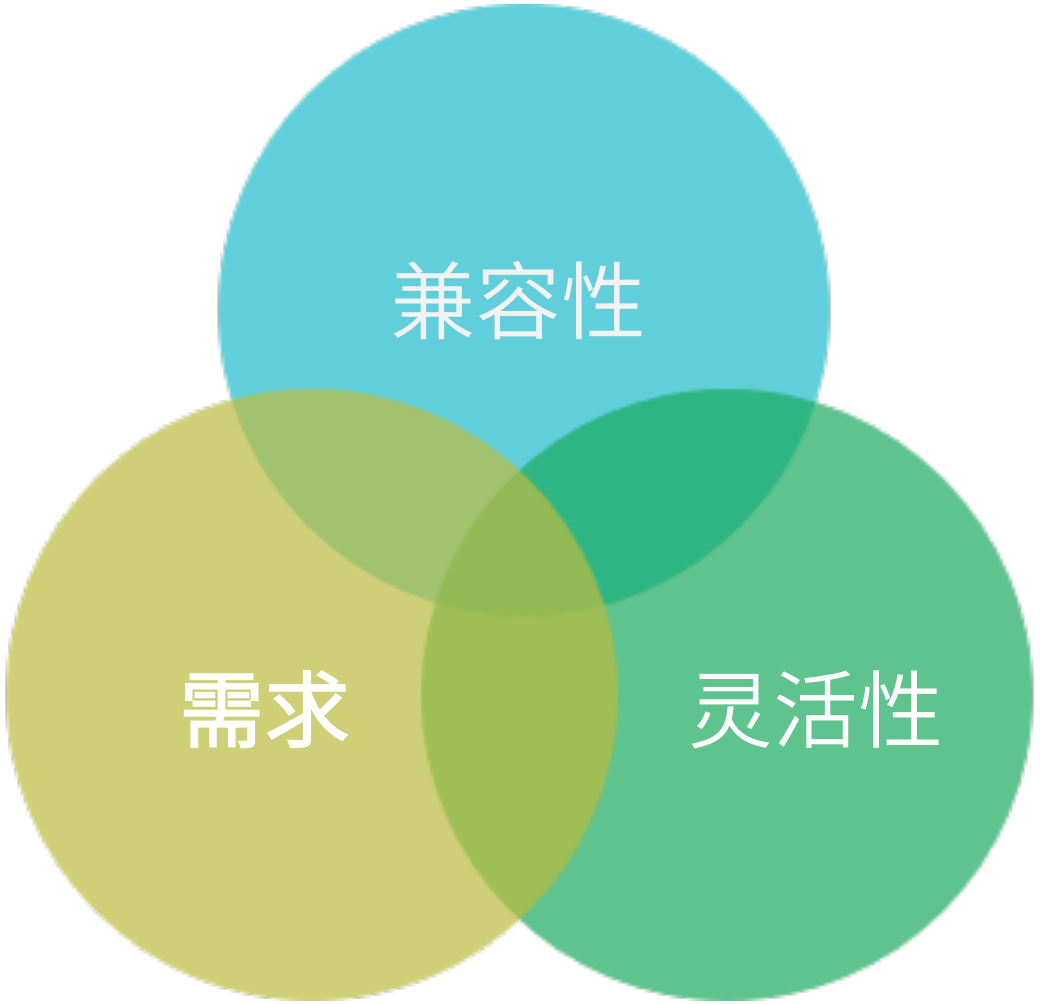
- grid-area
- grid-auto-columns
- grid-auto-flow
- grid-auto-rows
- grid-column
- grid-column-end
- grid-column-gap
- grid-column-start
- grid-gap
- grid-row
- grid-row-end
- grid-row-gap
- grid-row-start
- grid-template
- grid-template-areas
- grid-template-columns



真的谢谢W3C的叔叔哥哥姐姐阿姨们。🙂


ThoughtWorks®
IT大咖说
知识分享平台
方案选择

方案选择



+

团队?

向前看

向后看

PC端

PC端支持现状

flex box:

IE11及以上是支持的，其他浏览器基本上没问题

CSS grid:

支持非常不全，不建议使用

考察用户群体

判断用户主要使用的浏览器类型，用数据说服产品和老板。

纯移动端 推荐

flex+rem

- 兼容性：除了U开头的浏览器，基本没有bug
- 需求：没有任何问题
- 灵活性：很爽🙂

CSS grid

对于比较冲动，比较帅的同学，可以在新项目中使用。

不过要做好填坑的心理准备。

ThoughtWorks® | IT大咖说 知识分享平台

THANK YOU

ThoughtWorks®