

# CSS布局说——可能是最全的

CSS 发布于 2017-09-26 · 约 22 分钟

## 前言

现在，我们被称为前端工程师。然而，早年给我们的称呼却是页面仔。或许是职责越来越大，整体的前端井喷式的发展，使我们只关注了js，而疏远了css和html。

其实，我们可能经常在聊组件化，咋地咋地。但是，回过头来思考一下，如果你看到一张设计稿的时候，连布局都不清不楚，谈何组件化呢。所以，我们需要在分清楚组件之前，先来分清楚布局。废话说了这么多，只是想告诉你，布局这个东西真的很重要。本篇内容概括了布局的基础+基本的PC端布局+移动端适配等内容。如果你喜欢我的文章，欢迎评论，欢迎Star~。欢迎关注我的[github博客](#)

## 正文

或许对于你来说，喜欢js的背后，是看不懂css的深层。入门级的css非常简单，但是，精通它却没有想象的容易。我们本篇聊的是布局。前端开发，从拿到设计稿的那一刻，布局的思考就已经开始了。

举个例子，建筑师在设计房屋的时候，需要丈量开发地块的长度，以及每幢房屋相对于其他房屋的位置。

在css布局中，似乎也是这样开始的。我们也会去区分每个元素的尺寸和定位，力争完美的实现整个设计稿。所以，我们的布局应该从定位和尺寸开始聊起

## 定位

定位的概念就是它允许你定义一个元素相对于其他正常元素的位置，它应该出现在哪里，这里的其他元素可以是父元素，另一个元素甚至是浏览器窗口本身。还有就是浮动了，其实浮动并不完全算是定位，它的特性非常的神奇，以至于它在布局中被人广泛的应用。我们会在后文中专门提及它的。

谈及定位，我们就得从position属性说起。你能准确的说出position的属性值吗？相信你可以完美地说出这么六个属性值：static、relative、absolute、fixed、sticky和inherit。

- static(默认)：元素框正常生成。块级元素生成一个矩形框，作为文档流的一部分；行内元素则会创建一个或多个行框，置于其父元素中。
- relative：元素框相对于之前正常文档流中的位置发生偏移，并且原先的位置仍然被占据。发生偏移的时候，可能会覆盖其他元素。
- absolute：元素框不再占有文档流位置，并且相对于包含块进行偏移(所谓的包含块就是最近一级外层元素position不为static的元素)
- fixed：元素框不再占有文档流位置，并且相对于视窗进行定位
- sticky：(这是css3新增的属性值)粘性定位，官方的介绍比较简单，或许你不能理解。其实，它就相当于relative和fixed混合。最初会被当作是relative，相对于原来的位置进行偏移；一旦超过一定阈值之后，会被当成fixed定位，相对于视口进行定位。

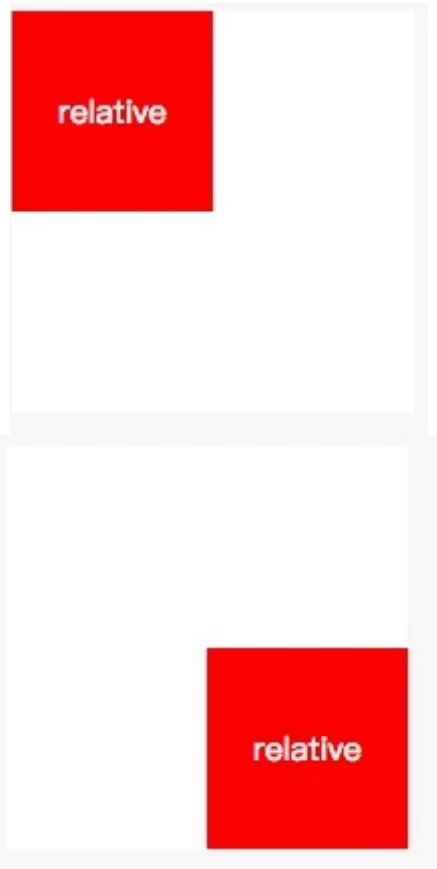
[demo地址](#)

简单地，介绍一下position的属性值的含义后，在来看一下偏移量top、right、bottom、left四个属性。

不清楚，当初在初学css的时候，会不会与margin这个属性混淆？其实，它们之间是很容易去辨识地。因为这四个属性值，其实是，定位时的偏移量。偏移量不会对static的元素起到作用。而margin，相对应的是盒子模型的外边距，它会对每个元素框起到作用，使得元素框与其他元素之间产生空白。

下面：我们来看一下一些常用定位的偏移

- relative：它的偏移是相对于原先在文档流中的位置，如图：



这里设置了top: 100px, left: 100px。

- absolute：它的偏移量是相对于最近一级position不是static的祖先元素的
- fixed：它的偏移量是相对于视口的。

其实，这里说描述的内容，应该都是需要理解的。这些相对于布局来说是基础的，同时也是非常重要的。需要注意的是，这里的偏移量其实已经涉及到了接下来要说的尺寸。在做自适应布局设计时，往往希望这些偏移量的单位能够使用百分比，或者相对的单位例如rem等。

## 尺寸

那之前上面谈到过尺寸的单位——百分比。那么，下面部分我们就围绕着尺寸单位展开。

尺寸，我们就应该从单位聊起，对于px这个单位，做网页的应该在熟悉不过了，因此不多做介绍。

那么，我们可以来介绍一下下面几个单位：

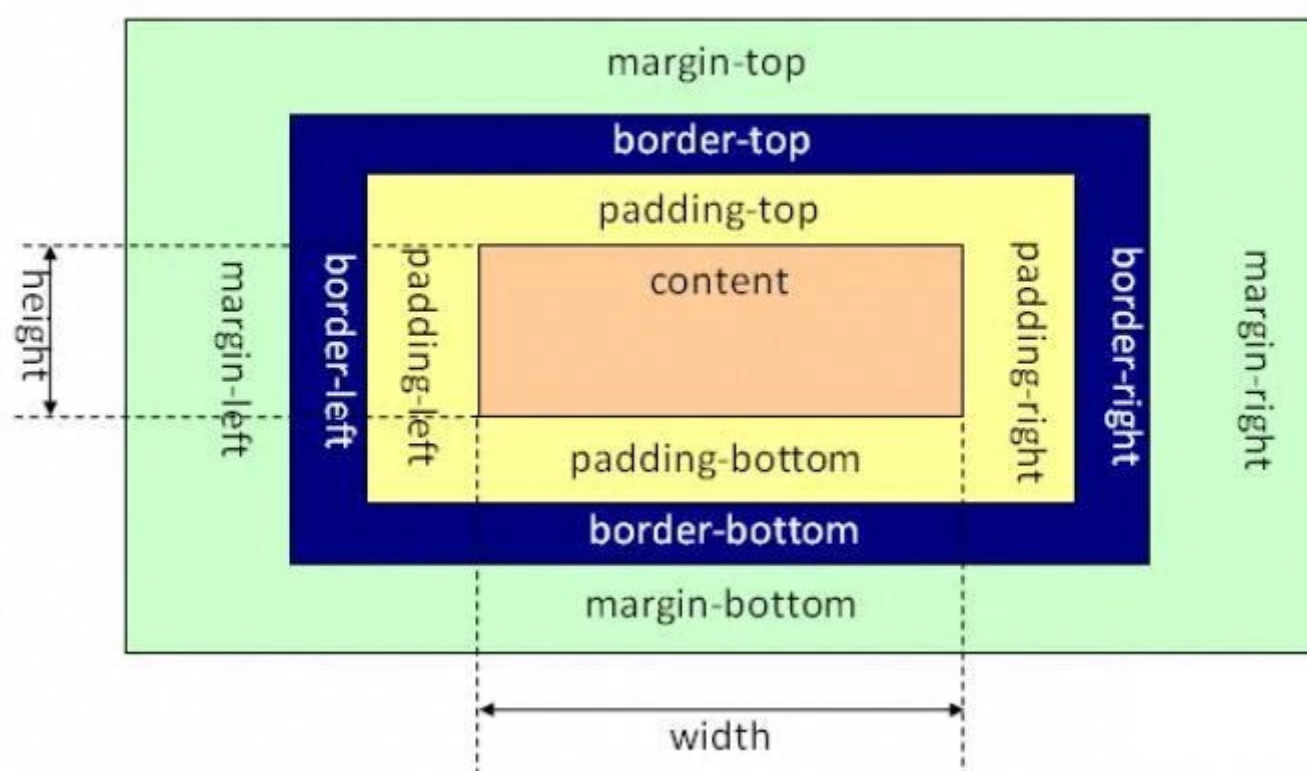
- 百分比：百分比的参照物是父元素，50%相当于父元素width的50%
- rem：这个对于复杂的设计图相当有用，它是html的font-size的大小
- em：它虽然也是一个相对的单位，相对于父元素的font-size，但是，并不常用，主要是计算太麻烦了。

单位只是一个来定义元素大小的相应参考。另一个概念，或许可以用房子来打一个比方，在早年每幢房子都会在房子的外围建一层栅栏，使得整一块地区可以看成房子+内部地块+栅栏+外围地块的模型。而在css中，每个元素也会有**盒子模型**的概念。

**盒子模型**：每个元素，都会形成一个矩形块，主要包括四部分：margin(外边距)+border(边框)+padding(内边距)+content(内容)

css中存在两种不同的盒子模型，可以通过box-sizing设置不同的模型。两种盒子模型，主要是width的宽度不同。如图：

## ■ 标准盒子模型



这是标准盒子模型，可以看到width的长度等于content的宽度；而当将box-sizing的属性值设置成border-box时，盒子模型的width=border+padding+content的总和。

可以看出，对于不同的模型的宽度是不同的。宽度默认的属性值是auto，这个属性值会使得内部元素的长度自动填满父元素的width。如图：



但是，height的属性值也是默认的auto，为什么没有像width一样呢？

其实，auto这个属性值表示的是浏览器自动计算。这种自动计算，需要一个基准，一般浏览器都是允许高度滚动的，所以，会导致一个问题——浏览器找不到垂直方向上的基准。

同样地道理也会被应用在margin属性上。相信如果考察居中时，水平居中你可能闭着眼睛都能写出来，但是垂直居中却绕着脑袋想。这是因为如果是块级元素水平居中只要将水平方向上的margin设置成auto就可以了。但是，垂直方向上却没有这么简单，因为你设置成auto时，margin为0。这个问题，还是需要仔细思考一下的。

到此为止，布局最基本的部分我们已经将去大半，还有就是一块浮动。

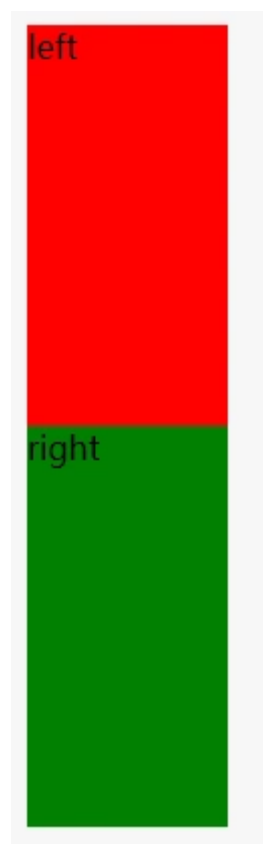
## 浮动

浮动，这是一个非常有意思的东西，在布局中被广泛的应用。最初，设计浮动时，其实并不是为了布局的，而是为了实现文字环绕的特效，如图：

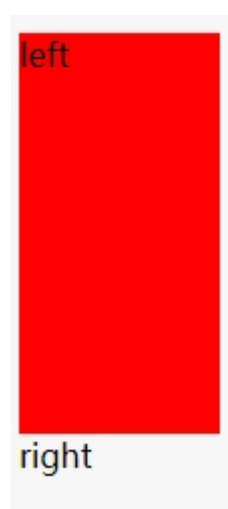


但是，浮动并不是仅仅这样而已。何为浮动？浮动应该说是‘自成一派’，类似于ps中的图层一样，浮动的元素会在浮动层上面进行排布，而在原先文档流中的元素位置，会被以某种方式进行删除，但是还是会影响布局。你可能会觉得有疑问，什么叫影响布局？我们可以来举个例子：

首先，我们准备两个颜色块，如图：



之后我们将left的块设置成左浮动，如图：



可以，发现虽然left块因为左浮动，而使得原先元素在文档流中占有的位置被删除，但是，当right块补上原先的位置时，right块中的字体却被挤出来了。这就是所谓的影响布局。

浮动为什么会被使用在布局中呢？因为设置浮动后的元素会形成BFC（使得内部的元素不会被外部所干扰），并且元素的宽度也不再自适应父元素宽度，而是适应自身内容。这样就可以，轻松地实现多栏布局的效果。

浮动的内容还需要介绍一块——清除浮动。可以看到，浮动元素，其实对于布局来说，是特别危险的。因为你可能这一块做过浮动，但未做清除，那么造成高度塌陷的问题。就是上面图示的那种情况。

**清除浮动**，最常用的方法有两种：

- overflow: 将父元素的overflow，设置成hidden。
- after伪类：对子元素的after伪类进行设置。

这里只是稍微的提上一嘴。下面我们正式来介绍一下网页的布局，本篇最核心的东西。

## 最初的布局——table

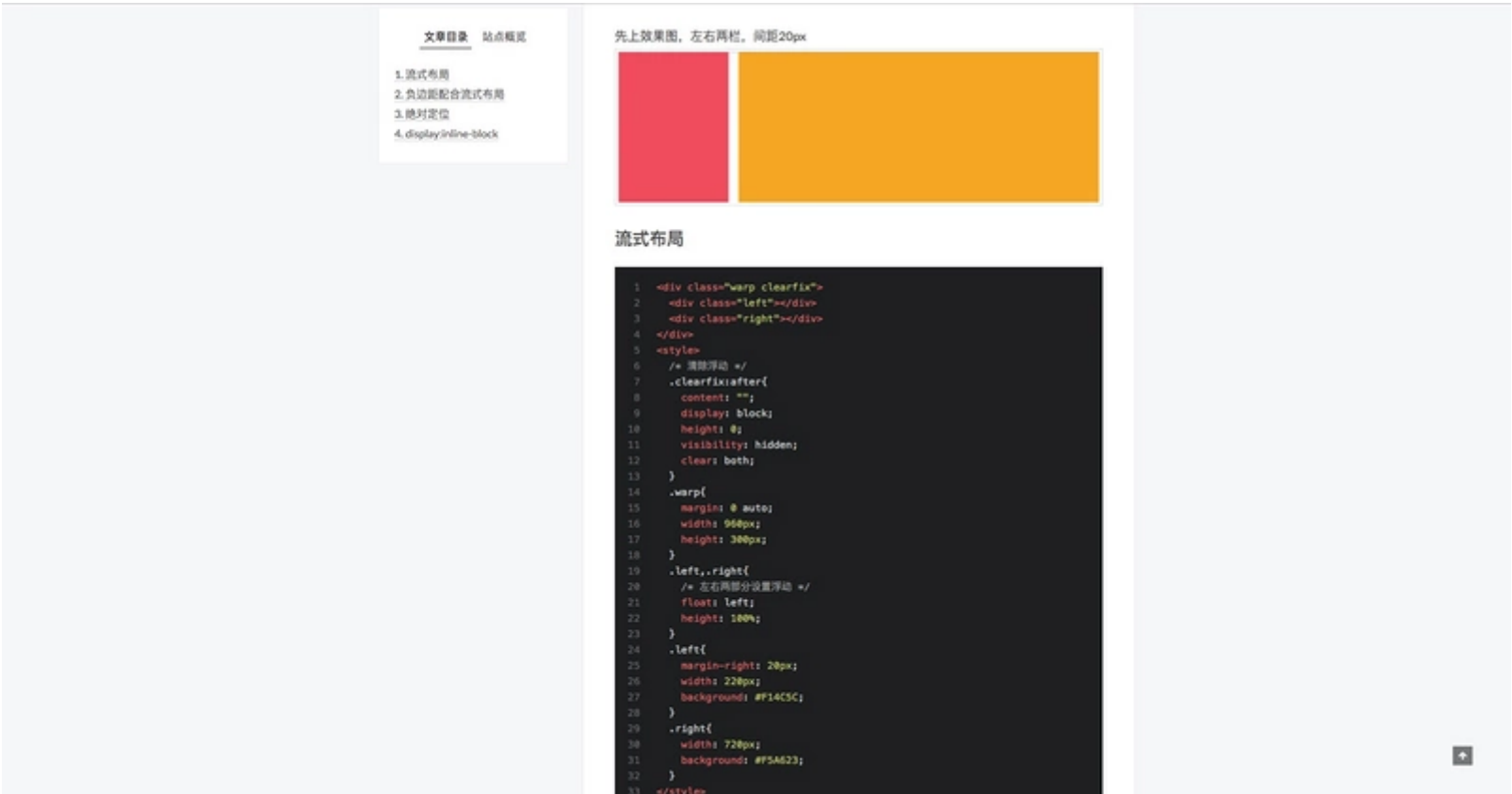
最初的时候，网页简单到可能只有文字和链接。这时候，大家最常用的就是table。因为table可以展示出多个块的排布。

这种布局现在应该不常用了，因为在形色单一时，使用起来方便。但是，现在的网页变得越来越复杂，适配的问题也是越来越多，这种布局已经不再时候了。

主要是div块的出现，可以使得网页进行灵活的排布，使得网页变得繁荣。这时，开发者也开始思索如何去更加清晰地分辨网页的层次。接下来，我们可以看看有哪些比较出名的布局方式。

## 两栏布局

是否记得，那些一边主体内容，一边目录的网页，如图：



类似于上图的布局，可以定义为**两栏布局**。

**两栏布局：**一栏定宽，一栏自适应。这样子做的好处是定宽的那一栏可以做广告，自适应的可以作为内容主体。

**实现的方式：**

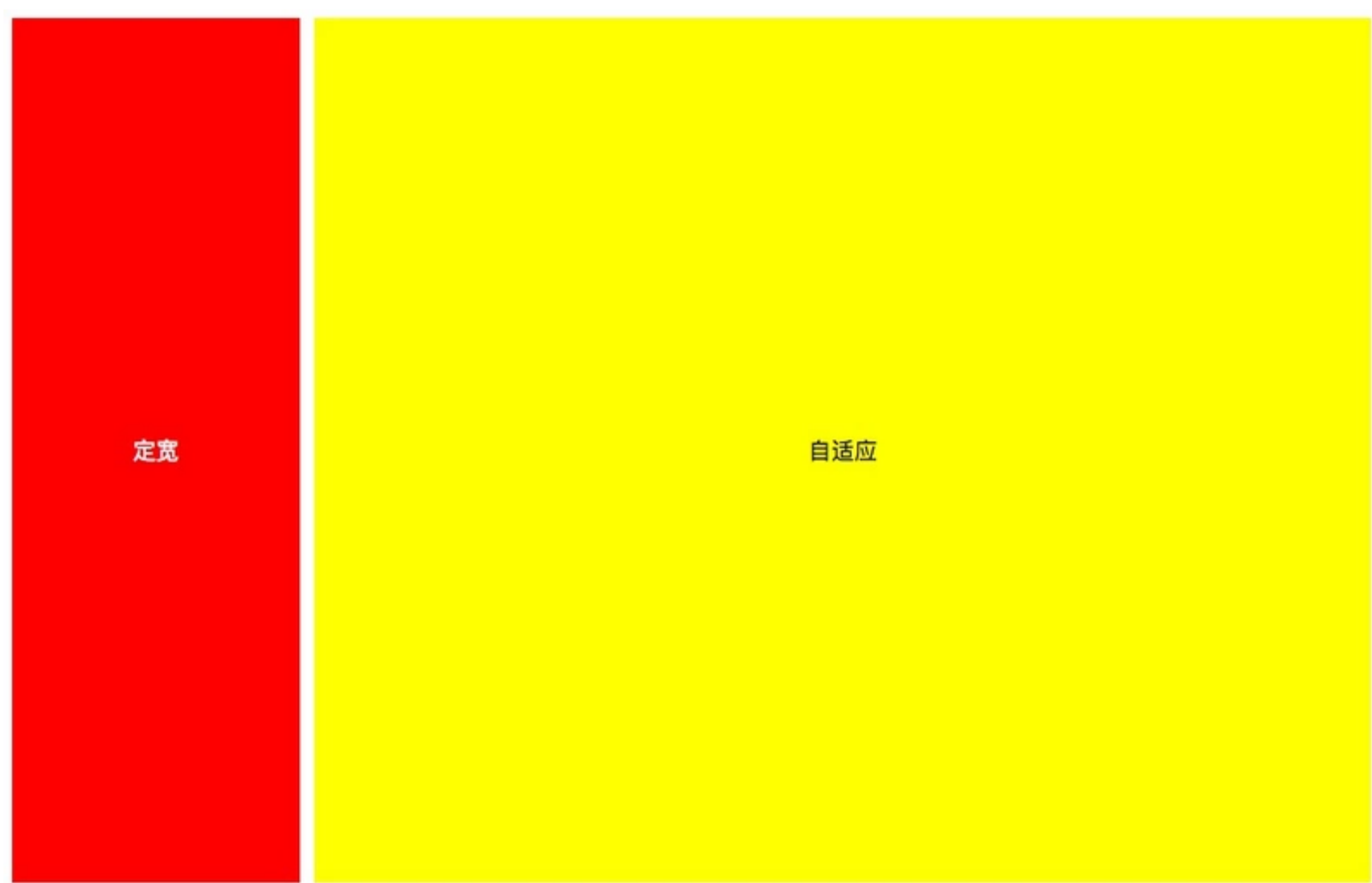
- 1. float + margin:

```
<body>
  <div class="left">定宽</div>
  <div class="right">自适应</div>
</body>
```

```
.left{
  width: 200px;
  height: 600px;
  background: red;
  float: left;
  display: table;
  text-align: center;
  line-height: 600px;
  color: #fff;
}

.right{
  margin-left: 210px;
  height: 600px;
  background: yellow;
  text-align: center;
  line-height: 600px;
}
```

如图所示：



其他的方法：还可以使用position的absolute，可以使用同样的效果

## 三栏布局

三栏布局，也是经常会被使用到的一种布局。

它的特点：两边定宽，然后中间的width是auto的，可以自适应内容，再加上margin边距，来进行设定。

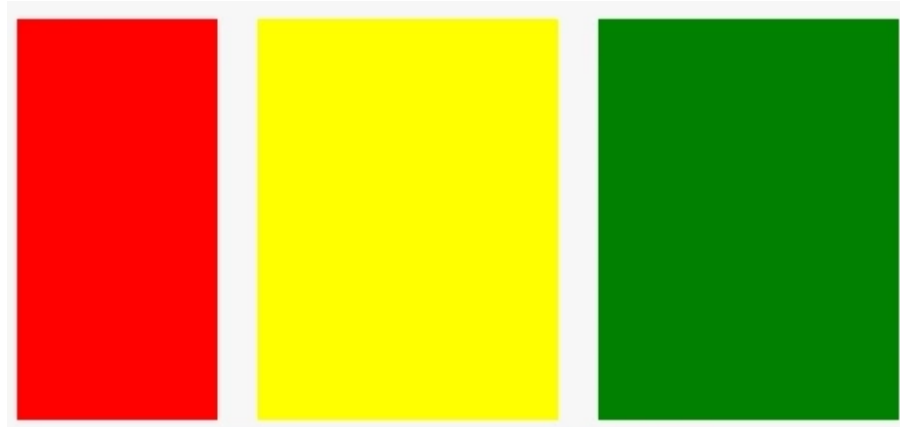
三栏布局可以有4种实现方式，每种实现方式都有各自的优缺点。

### 1.使用左右两栏使用float属性，中间栏使用margin属性进行撑开，注意的是html的结果

```
<div class="left">左栏</div>
<div class="right">右栏</div>
<div class="middle">中间栏</div>
```



```
.left{
  width: 200px;height: 300px; background: yellow; float: left;
}
.right{
  width: 150px; height: 300px; background: green; float: right;
}
.middle{
  height: 300px; background: red; margin-left: 220px; margin-right: 160px;
}
```

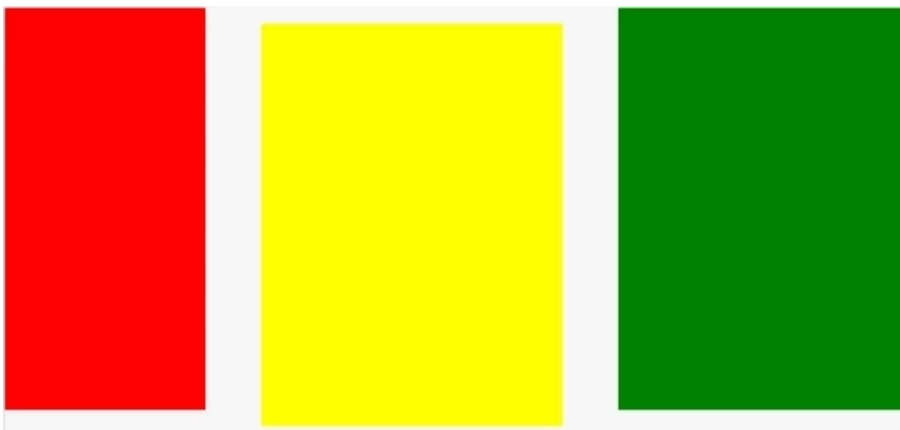


缺点是：1. 当宽度小于左右两边宽度之和时，右侧栏会被挤下去；2. html的结构不正确

## 2. 使用position定位实现，即左右两栏使用position进行定位，中间栏使用margin进行定位

```
<div class="left">左栏</div>
<div class="middle">中间栏</div>
<div class="right">右栏</div>
```

```
.left{
  background: yellow;
  width: 200px;
  height: 300px;
  position: absolute;
  top: 0;
  left: 0;
}
.middle{
  height: 300px;
  margin: 0 220px;
  background: red;
}
.right{
  height: 300px;
  width: 200px;
  position: absolute;
  top: 0;
  right: 0;
  background: green;
}
```



好处是：html结构正常。

缺点时：当父元素有内外边距时，会导致中间栏的位置出现偏差

3. 使用float和BFC配合圣杯布局

将middle的宽度设置为100%，然后将其float设置为left，其中的main块设置margin属性，然后左边栏设置float为left，之后设置margin为-100%，右栏也设置为float: left，之后margin-left为自身大小。

```
<div class="wrapper">
  <div class="middle">
    <div class="main">中间</div>
  </div>
  <div class="left">
    左栏
  </div>
  <div class="right">
    右栏
  </div>
</div>
```

```
.wrapper{
  overflow: hidden; //清除浮动
}
.middle{
  width: 100%;
  float: left;
}
.middle .main{
  margin: 0 220px;
  background: red;
}
.left{
  width: 200px;
  height: 300px;
  float: left;
  background: green;
  margin-left: -100%;
}
.right{
  width: 200px;
  height: 300px;
  float: left;
  background: yellow;
  margin-left: -200px;
}
```



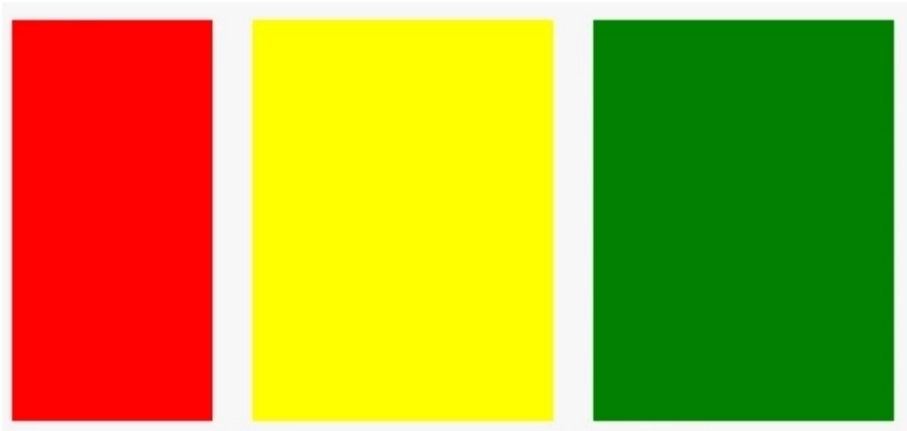
缺点是：1. 结构不正确 2. 多了一层标签

4. flex布局



```
<div class="wrapper">
  <div class="left">左栏</div>
  <div class="middle">中间</div>
  <div class="right">右栏</div>
</div>
```

```
.wrapper{
  display: flex;
}
.left{
  width: 200px;
  height: 300px;
  background: green;
}
.middle{
  width: 100%;
  background: red;
  margin: 0 20px;
}
.right{
  width: 200px;
  height: 300px;
  background: yellow;
}
```



除了兼容性，一般没有太大的缺陷

三栏布局使用较为广泛，不过也是比较基础的布局方式。对于PC端的网页来说，使用较多，但是移动端，本身宽度的限制，很难做到三栏的方式。因此，移动端盛行的现在，我们应该掌握一些自适应布局技巧和flex等方式。

## 移动端的时代

或许，手机占用了人们大部分的时间，对于前端工程师来说，不仅需要会h5和大前端的技术，还需要去适配不同的手机屏幕。PC端称王的时代已经过去，现在要求的网页都是需要能够去适配PC和移动端的。

之前，所聊的两栏和三栏布局，一般只能在PC中去使用，在移动端，由于屏幕尺寸有限，很难去做到类似的操作，所以，我们需要来了解新的东西。

### 1. 媒体查询

如果你需要一张网页能够在PC和移动端都能按照不同的设计稿显示出来，那么你需要做的就是去设置媒体查询。

媒体查询的css标识符是@media，它的媒体类型可以分为：

1. all， 所有媒体
2. braille 盲文触觉设备
3. embossed 盲文打印机
4. print 手持设备
5. projection 打印预览

- 6. screen 彩屏设备
- 7. speech ‘听觉’类似的媒体类型
- 8. tty 不适用像素的设备
- 9. tv 电视

代码示例：

```
@media screen {
  p.test {font-family:verdana,sans-serif;font-size:14px;}
}
@media print {
  p.test {font-family:times,serif;font-size:10px;}
}
@media screen,print {
  p.test {font-weight:bold;}
}
/* 移动端样式*/
@media only screen and (min-device-width : 320px) and (max-device-width : 480px) {
  /* Styles */
}
```

媒体查询的主要原理：它像是给整个css样式设置了断点，通过给定的条件去判断，在不同的条件下，显示不同的样式。例如：手机端的尺寸在750px，而PC端则是大于750px的，我们可以将样式写成：

```
@media screen and (min-width: 750px){
  .media{
    height: 100px;
    background: red;
  }
}

@media (max-width: 750px){
  .media{
    height: 200px;
    background: green;
  }
}
```

[demo地址](#)

效果图：





**flex弹性盒子**

其实移动端会经常使用到flex布局，因为在简单的页面适配方面，flex可以起到很好的拉伸的效果。我们先看几张图体会一下：





可以从图中看出，这个网页不管屏幕分辨率怎么发生变化，它的高度和位置都不变，而且里面的元素排布也没有发生变化，总是图标信息在左边和薪资状况在右边。

这就是很明显的，flex布局。flex可以在移动端适配比较简单的，元素比较单一的页面。

具体的flex布局内容，在这里不详细说明。[flex传送门](#)

rem适配

rem可以说是移动端适配的一个神器。类似于手淘等界面，都是使用的rem进行的适配。这种界面有个特点就是页面元素的复杂度比较高，而使用flex进行布局会导致页面被拉伸，但是上下的高度却没有变化等问题。示例图：



具体的讲解可以看这篇比较好的文章。[rem传送门](#)

# 总结

分析到这里，布局的很多东西都已经非常的清晰了。相信这是一篇值得去收藏的一篇文章。内容可能有点多，所以这里做一个总结：

- 定位
- 尺寸
- 浮动
- 最初的布局——table
- 两栏布局
- 三栏布局
- 移动端的布局

相信你看完这些，在以后，一定可以拿到设计稿的时候，内心大致有个算盘，应该如何区分，如何布局。

最后，如果你对我写的有疑问，可以与我讨论。如果我写的有错误，欢迎指正。你喜欢我的博客，请给我关注Star~呦。大家一起总结一起进步。欢迎关注我的[github博客](#)

阅读 70.5k • 发布于 2017-09-26

 赞 188

 收藏 904

 分享

本作品系 原创 ， 采用《署名-非商业性使用-禁止演绎 4.0 国际》许可协议



**zimo**  
2.2k

关注作者


20 条评论



得票 • 时间







撰写评论 ...



提交评论



- 


**farmerz**： 个人觉得明白了bfc和flex之后布局就很简单了，因为剩下的都是万变不离其宗的一些设置。  
 6 • [回复](#) • 2017-09-27
- 

**mango**： 那是因为你还没有碰见其他问题  
 1 • [回复](#) • 2018-06-07
- 

**ChangFaiX**： 这个标题很罗永浩  
 5 • [回复](#) • 2017-09-27
- 

**全真教人尹志平**： 重构高手  
 4 • [回复](#) • 2017-09-27
- 

**changli**： flex 进行3栏布局中 .middle {width: 100%} 会导致弹性盒处在压缩状态, 导致两个侧边栏的宽度压缩,并小于设置的200px  
正确写法是 修改.middle { flex: 1} 或者是添加 .left, .right {flex-shrink: 0}  
 4 • [回复](#) • 2018-05-29
- 

**thomas**： 丹丹  
 2 • [回复](#) • 2017-09-27





**mengdu**：多列布局，这种实用些 <https://jsfiddle.net/s6xzvk3f/1/>

👍 1 • 回复 • 2017-09-27

**zimo**：学到了，谢谢

👍 . • 回复 • 2017-09-27

**1862**：也是一种方法, 缺点是注重在DOM中的顺序

👍 . • 回复 • 2018-05-03



**phping**：只有我一个人看不见图吗

👍 1 • 回复 • 2018-09-13

**sixdein**：我也看不到

👍 . • 回复 • 2018-09-14

**dubi**：+1 怎么解决呀

👍 . • 回复 • 2018-09-18

**英林**：+1...

👍 . • 回复 • 2018-10-07

[展开显示更多](#)



**clearlivings**：谢谢作者啊

👍 . • 回复 • 2017-12-21



**卧龙软件**：前端技术也是博大精深、浩瀚无垠那哪！

👍 . • 回复 • 2018-02-17



**杜灿**：麻烦问一下图中打开网易的工具是什么？

👍 . • 回复 • 2018-03-22

推荐阅读

CSS常见布局解决方案

想必是个前端都见过，这定宽的水平居中，我们还可以用下面这种来实现不定宽的display:table在表现上类似block元素，但是宽度...  
[山海](#) • 阅读 30

CSS 单位

CSS中的长度单位有很多，觉得有必要整理备忘。网上介绍的也比较多和全，详见参考资料。单位大体分为两大类：绝对单位，不...  
[Chinaxiang](#) • 阅读 3

常见布局方法总结-居中布局

(1)width:(xxx)px;margin:0auto;使用场景：页面整体水平居中，有具体宽度要求。(2)text-align:center;display:inline-block;使用场景...  
[nanie](#) • 阅读 2

margin详解

a.适用于没有设定widthheight的普通block水平元素float元素、absolutefixed元素、inline元素、table-cell元素都不可以b.只适用于...  
[hopeCoder](#) • 阅读 1

Flutter样式和布局控件简析(二)



继续接着分析Flutter相关的样式和布局控件，但是这次内容难度感觉比较高，怕有分析不到位的地方，所以这次仅仅当做一个参考...

tain335 · 阅读 289

## 从零开始 搭建自己的vue 移动项目

在main.js中写下1，去定义阿里图标，并引入2，设置icon组件，并写html3，因为vue模板没有引入scss，所以要自己安装sass-load...

lidog · 阅读 960

## 如何用CSS进行网页布局-imooc-【更新完毕】

网页可以自适应宽度网页的长度理论上可以无限延长页面为：头部主体部分底部分栏又称为分列：一列布局二列布局三列布局混合...

太宰夜 · 阅读 12

## MDL布局组件

MDL的布局Layout组件用来作为整个页面其他元素的容器，可以自动适应不同的浏览器、屏幕尺寸和设备。布局Layout组件需要按...

whjin · 阅读 10

### 前端随笔

用户专栏

或许，我们之间只是一篇文章的距离；知识，只是几分钟的阅读。希望我们能够在此专栏里面一起成长，一起...

76 人关注    15 篇文章

关注专栏    专栏主页

#### 产品

热门问答  
热门专栏  
热门课程  
最新活动  
技术圈  
酷工作  
移动客户端

#### 课程

Java 开发课程  
PHP 开发课程  
Python 开发课程  
前端开发课程  
移动开发课程

#### 资源

每周精选  
用户排行榜  
徽章  
帮助中心  
声望与权限  
社区服务中心

#### 合作

关于我们  
广告投放  
职位发布  
讲师招募  
联系我们  
合作伙伴

#### 关注

产品技术日志  
社区运营日志  
市场运营日志  
团队日志  
社区访谈

#### 条款

服务条款  
隐私政策  
下载 App