

从自适应到响应式布局

一、问题引出

刘江老师的能源看板项目中，我们前端在自己一直觉得是按照原型图写的样式，然后交付的时候老师一直反应样式有问题，我们一直看不出问题在哪里，然后开会的时候老师发了一个他的效果图。上面这张是我们电脑上显示效果，下面这张是老师看到的显示效果。



二、理论体系背景

1 问题域

通过和刘江老师沟通，我们知道他用的是macbook pro 13，设备不同，分辨率不一样。其中字体的自适应和盒子的**自适应**都存在问题。

1.1 自适应布局

定义：自适应布局、又称宽度百分比布局或者流式布局。在不同的设备上呈现相同的网页。

1.2 响应式布局

定义：根据不同的设备显示不同的页面。

1.3 两者示例

自适应布局示例：<https://www.bilibili.com/>

响应式布局示例：<http://doc.jetlinks.cn/>

2 术语体系

2.1 基本单位介绍

单位	作用	特性
px	页面按精确像素展示	绝对单位
em	相对父节点字体大小	相对单位
%	相对于父元素的大小设定的比率	相对单位
rem	相对根节点html字体大小来计算	相对单位
vw	视窗宽度的百分比，1vw代表视窗宽度的1%	视窗单位
vh	视窗高度的百分比，1vh代表视窗高度的1%	视窗单位
vmin	当前vw和vh中较小的一个值	视窗单位
vmax	当前vw和vh中较大的一个值	视窗单位

实验一：在没有css3新特性前实现自适应的方式。

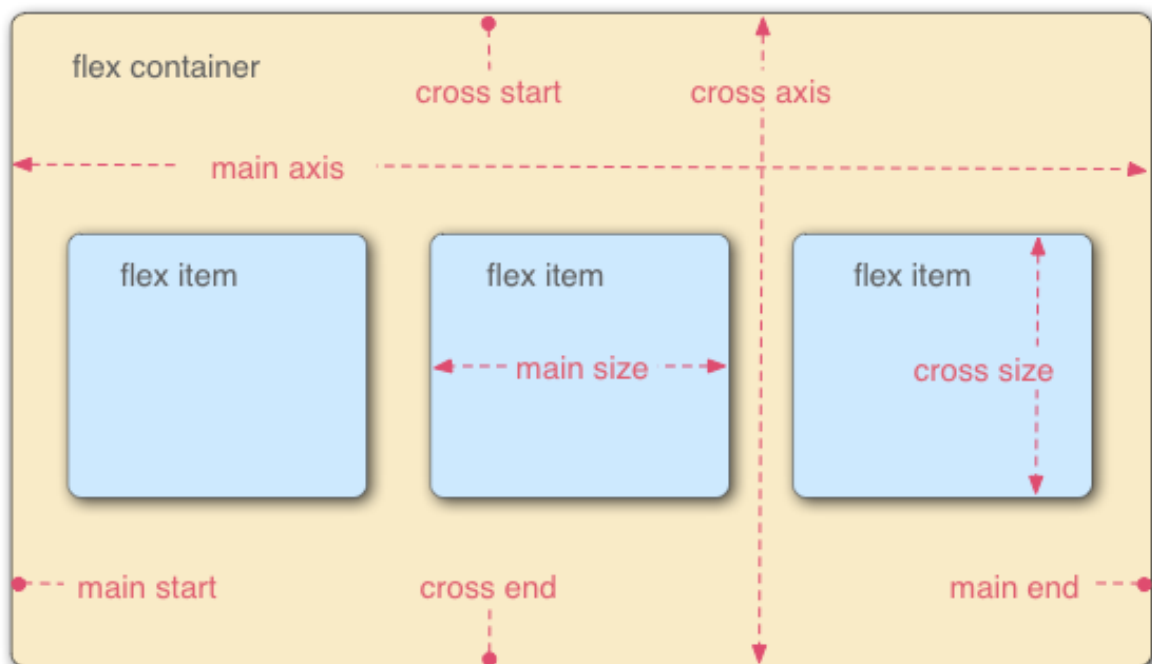
实验思路：通过百分比设置根节点的字体和高度，然后通过em和%元素从父节点继承大小，然后用js实时监听界面窗口大小，当窗口改变后操作dom树更改根节点字体大小。

实验二：引入css3新特性后实现自适应的方式。

实验思路：只需要用vh和vw单位就可以实现。

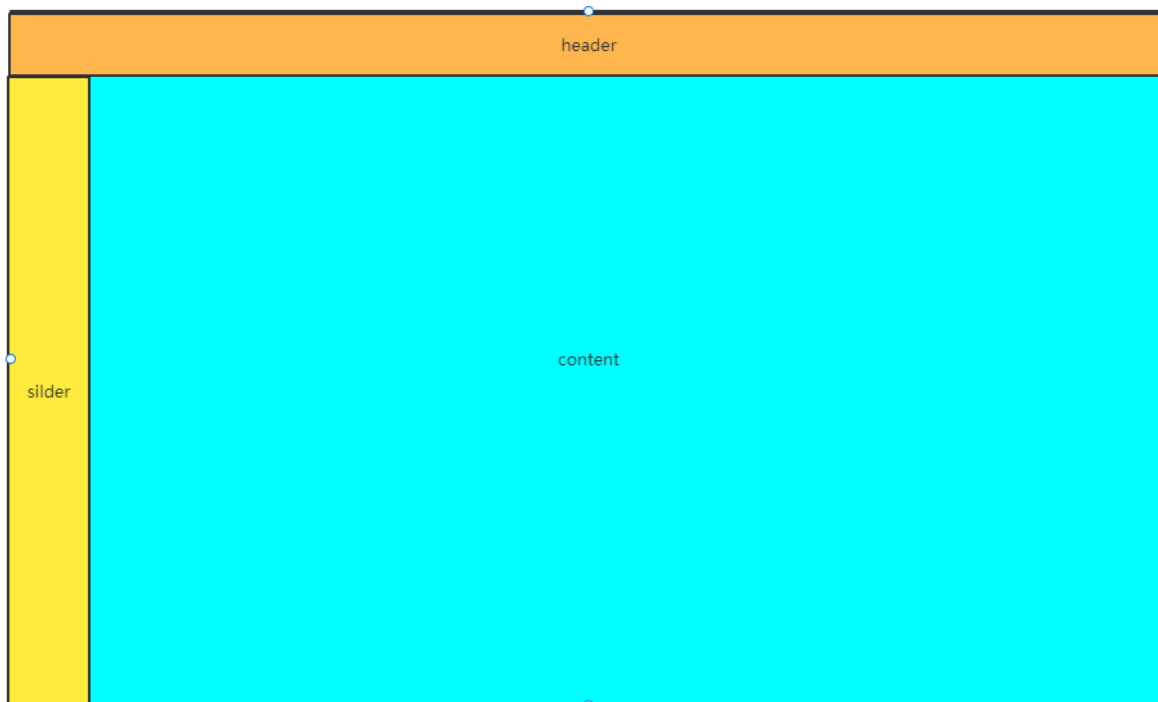
结论：在没有css3特性之前，实现字体的自适应是需要通过js技术的，css3新特性就没有了这个问题，并且提供了一种类似于"全局的变量"的单位进行相对。

2.2 flex弹性盒子



容器属性	作用
flex-direction	它可以指定主轴的方向，它默认值是 <code>row</code> ，这使得它们在按你浏览器的默认语言方向排成一行（在英语/中文浏览器中是从左到右）
flex-wrap	决定容器内项目是否可换行，默认是 <code>flex-wrap:no-wrap</code> (不换行),当盒子放不下时也不会溢出,而是盒子伸缩
justify-content	定义了项目在主轴上的对齐方式
align-items	设置侧轴上的子元素排列方式(单行)

子项属性	作用
order	排列顺序，越小的排前面，默认为0
flex-grow	放大比例，默认为0，如果所有都为1，等分剩余空间
flex-shrink	缩小比例，默认为1
flex-basis	设置固定宽高，可以占据固定空间，默认auto
flex	flex-grow flex-shrink flex-basis的简写，后两个可选
align-self	项目单独对齐方式，可覆盖align-items属性



实验3：传统的方式进行上图布局格式

实现思路：header部分占整个宽度的20%，设置整个下面，占比80%，距离上面20%。silder宽度占整个下面的5%，content，宽度占整个界面比95%，距离界面左边5%。

实验4：用flex弹性盒子实现

首先把整个界面设置成flex，然后让上下两个元素纵向的1：15，然后再把下面设置成flex，让silder和content的比例1:19。

结论：传统的position方式需要对每一个相对定位都进行较为详细的描述，而flex只需要注重在主轴方向上的占比。

2.3 媒体查询(@media)

响应式设计仅仅是因为媒体查询才兴起的，仅在浏览器和设备的环境与你指定的规则相匹配的时候CSS才会真的被应用，例如用户的屏幕是否大于某个宽度或者某个分辨率，并将CSS选择性地应用在所需页面上。

```
@media media-type and (media-feature-rule) {  
    /* CSS rules go here */  
}
```

media-feature-rule

属性名称	作用
width、height	浏览器可视宽度、高度
max-width(height)	判断最大值
min-width(height))	判断最小值
orientation	判断竖放和横放模式。

媒体常见四个区间

```
/*移动端: 0-768px*/  
@media screen and (max-width:768px)  
/*平板: 768px-992px*/  
@media screen and (min-width:768px) and (max-width:992px)  
/*以前台式电脑: 992px-1200px*/  
@media screen and (min-width:992px) and (max-width:1200px)  
/*主流pc屏幕+电视机>1200px*/  
@media screen and (min-width:1200px)
```

实验5：用media，实现响应式布局。

实现思路：通过媒体查询技术，在屏幕宽度大于1000px的时候引入big.css样式，在屏幕宽度在400px-1000px之间的时候加载small.css的样式。

实验6：自适应布局实现响应式。

实现思路：通过函数判断设备类型，如果是移动设备，那么就跳转到我们提前编写好的move.html的界面，否则就跳转到pc.html界面。

结论：两种方式都可以实现响应式，但是实现的方式不一样，一个是通过媒体查询技术，根据查询规则，加载不同的样式文件，而自适应布局实现响应式，通过js判断设备类型，然后跳转到不同的界面显示。

2.4 总结

随着终端设备的越来越多，屏幕大小也是各种各样，所以前端界面的自适应尤为重要，现今开发的网站都或多或少的做了自适应效果。

- 响应式布局就是通过媒体查询技术，识别不同的设备前端加载不同的css样式，所以响应式我们只需要维护这一个界面的代码
- 自适应布局是通过针对不同的设备，加载不同的界面。用自适应布局，适配了多少种就有多少个界面，增加了维护成本。

三、应用场景

所有场景都是在基于需要做多端适配的需求上，如果没有这个需要，界面做到简单的自适应就够了。

1、场景

- 场景一：界面简单，展示内容不多（绝大部分系统）

响应式和自适应布局都可以用，这个时候用响应式会更好，不会增加维护第二个界面的成本。

- 场景二：界面复杂，设备展示的内容有很都不同，又重视加载速度（淘宝、电商）

这个时候就用自适应布局更好，因为响应式布局的一套代码不论在pc端还是移动端加载的数据是一样的，只是通过了css样式进行展示和不展示，像淘宝这种数据量大，那么在移动端加载整套pc端展示的全部数据就相对费时。

- 场景三：折叠屏幕手机适配

这种情况就是响应式布局会好的多，它可以根据屏幕做出实时的反应更改布局，但是自适应需要刷新界面，用户体验会不好。

2、实验室项目应用

响应式布局实现多端适配

元堡乡信息门户<http://ybxapqts.znqlzx.com/>

自适应布局实现多端适配暂时没有找到

3、总结

- 为什么我们不用响应式？

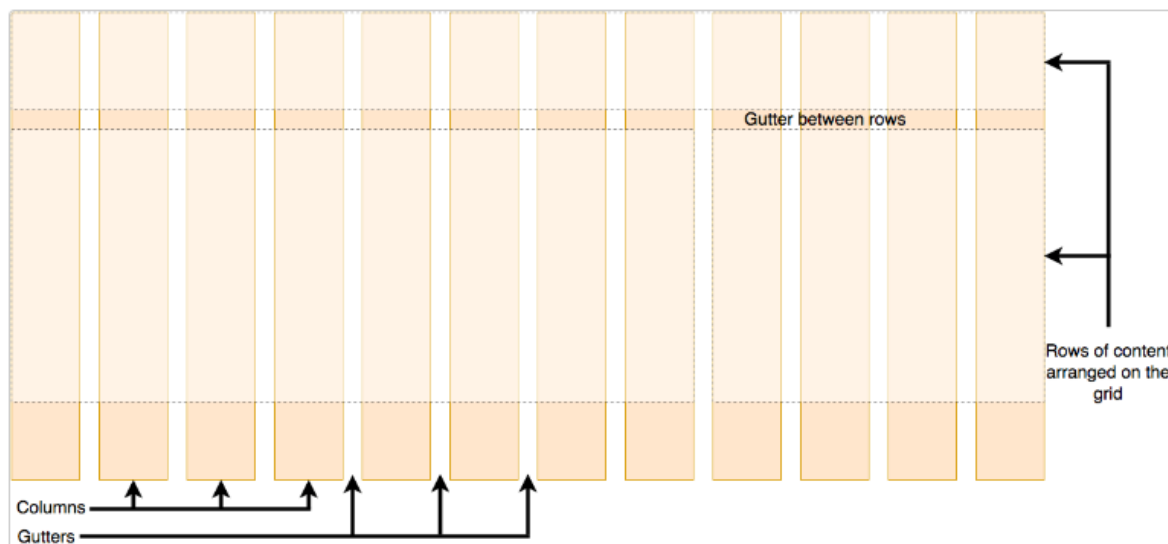
- 1、实验室项目几乎没有多端适配的需求。
- 2、响应式会带来额外的代码量

- 为什么有的自适应没有很好？

- 1、项目需求对自适应不是很大，大家都在PC端上显示，差别不大
- 2、界面是由多人协同开发完成，每个人都有自己的习惯。
- 3、实验室很多项目是复用原来的代码，改动代码太复杂了，成本太高。
- 4、前端的部分组件没有做自适应，我们直接拿来用

四、其他相关技术体系

1、grid 网格布局



网格是由一系列水平及垂直的线构成的一种布局模式。根据网格，我们能够将设计元素进行排列，帮助我们设计一系列具有固定位置以及宽度的元素的页面，使我们的网站页面更加统一。



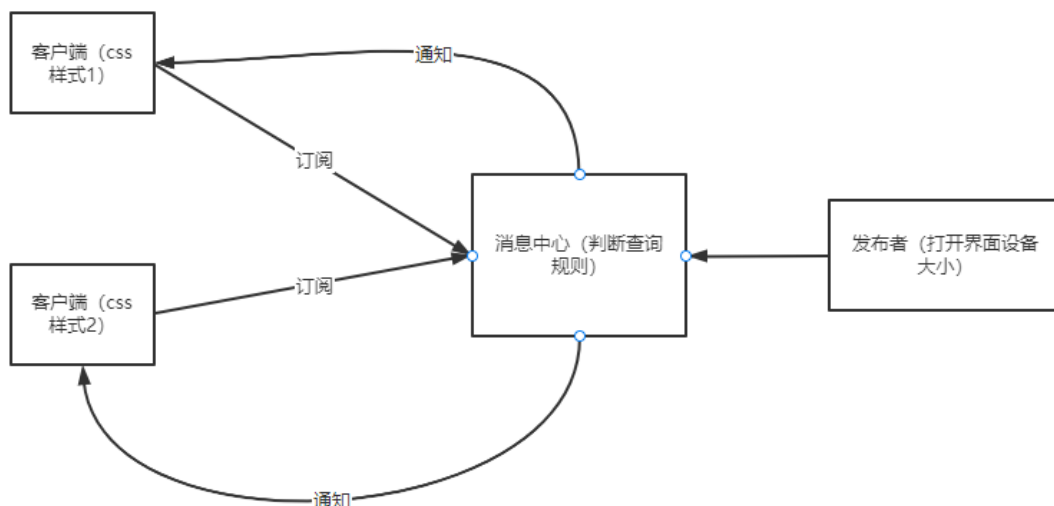
优点：对于整体布局上，非常方便，并且架构清晰，对于复杂布局架构不会有flex的多层嵌套。

2、vue响应式原理

我们可以看到上面的图中的总能耗成本，他是通过用电量、用气量、用水量的成本计算出来的，当其中任意一个价格改变，总价会跟着改变。想要完成这个过程我们需要：

- 侦测数据的变化 -----数据劫持/数据代理
- 收集视图依赖了哪些数据 -----依赖收集
- 数据变化时，自动“通知”更新的视图部分，并进行更新 -----发布订阅模式

3、响应式编程



响应式与`传统编程`最大的区别是：

响应式编程就是基于数据变化的新的编程模式，实现异步非阻塞，就是当请求来了之后进行订阅数据的变化，后续业务处理发布变化，然后进行监听到变化，进行响应。

五、在Jetlinks上的应用

1、界面响应式布局

2、响应式编程

- JetLinks使用[Project Reactor](#)作为响应式编程框架,从网络层到持久层全部封装为 非阻塞, 响应式 调用.
- 响应式可以理解为 观察者模式,通过 订阅 和 发布 数据流中的数据对数据进行处理.

六、总结与展望

- 无论是媒体查询还是响应式编程,亦或者是Vue的响应式,他们都有一个“监听者”,当“监听者”听到想听的消息时,马上做出响应。
- 随着技术的发展,会有愈来愈多的设备涌入,那么如何用一套代码去响应所以设备,在我看来应该是未来的一个趋势,当客户有在多种设备适配的需求的时候,那么就是我们呢界面革新的时候。
- 我们前端开发中,自适应问题一直是一个另人头疼的问题,所以希望大家在写代码的时候,首先就要有一个自适应的观念,构建一个全局的自适应布局,然后对局部可以进行自己的个性化编写。

七、参考文件

- 1、<https://www.runoob.com/bootstrap/bootstrap-intro.html> BootStrap简介 (外部资料)
- 2、https://developer.mozilla.org/zh-CN/docs/Learn/CSS/CSS_layout/Grids Grid网格 (外部资料)
- 3、https://developer.mozilla.org/zh-CN/docs/Learn/CSS/CSS_layout/Flexbox flex (外部资料)
- 4、https://blog.csdn.net/weCat_s/article/details/104346296 响应式布局 and 自适应布局的区别 (外部资料)
- 5、<https://baike.baidu.com/item/%E5%93%8D%E5%BA%94%E5%BC%8F%E5%B8%83%E5%B1%80/1220833?fr=aladdin> 响应式布局百度百科 (外部资料)
- 6、https://tower.im/teams/135591/repository_folders/16285/ Vue响应式原理 (刘畅) -- 数据发生变化后,会重新对页面渲染
- 7、https://tower.im/teams/135591/repository_uploads/32839/ 响应式编程 (姜海) -- 响应式编程是一种新的编程范式,其特点是异步、事件驱动、推送PUSH机制以及观察者模式的衍生。