

浅谈Web自适应

2017-06-04 前端大全

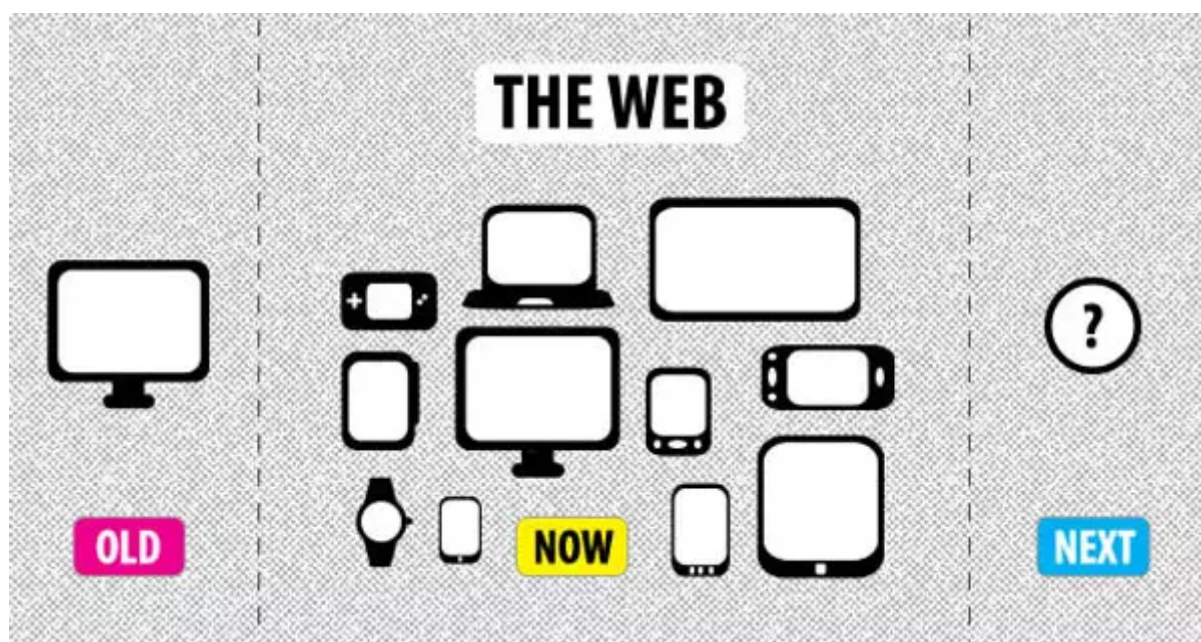
(点击上方公众号，可快速关注)

作者：卖烧烤夫斯基

<http://www.cnblogs.com/constantince/p/5708930.html>

[如有好文章投稿，请点击 → 这里了解详情](#)

前言



随着移动设备的普及，移动web在前端工程师们的工作中占有越来越重要的位置。移动设备更新速度频繁，手机厂商繁多，导致的问题是每一台机器的屏幕宽度和分辨率不一样。这给我们在编写前端界面时增加了困难，适配问题在当下显得越来越突出。记得刚刚开始开发移动端产品的时候向设计MM要了不同屏幕的设计图，结果可想而知。本篇博文分享一些卤煮处理多屏幕自适应的经验，希望有益于诸君。

特别说明：在开始这一切之前，请开发移动界面的工程师们在头部加上下面这条meta：

```
<meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0">
```

简单事情简单做-宽度自适应

所谓宽度自适应严格来说是一种PC端的自适应布局方式在移动端的延伸。在处理PC端的前端界面时候需要用到全屏布局时采用的就是此种布局方式。它的实现方式也比较简单，将外层容器元素按照百分比铺满地方式，里面的子元素固定或者左右浮动。

```
.div {  
  width:100%; height:100px;  
}  
.child {  
  float: left;  
}  
.child {  
  float:right;  
}
```

由于父级元素采用百分比的布局方式，随着屏幕的拉伸，它的宽度会无限的拉伸。而子元素由于采用了浮动，那么它们的位置也会固定在两端。该宽度自适应在新的时代有了新的方法，随着弹性布局的普及，它经常被flex或者box这样的伸缩性布局方式替代，变得越来越“弹性”十足。需要了解弹性布局，请前往Flex布局教程和卤煮box布局教程比较。

大小之辨-完全自适应

“完全自适应式”是卤煮对越此方案的叫法，由于卤煮现在找不到官方名称，所以暂时就这样叫它。这种解决方案相对前一种来说进步不少，不仅仅宽度实现了自适应，而且界面所有的元素大小和高度都会根据不同分辨率和屏幕宽度的设备来调整元素、字体、图片、高度等属性的值。简单来说就是在不同的屏幕下，你看到的字体和元素高宽度的大小是不一样的。在这里，有人就会说利用的是媒体查询熟悉，根据不同的屏幕宽度，调整样式。卤煮之前也是这样想的，但是你需要考虑到界面上的许多元素需要设置字体，如果用media query为每个元素在不同的设备下都设置不同的属性的话，那么有多少种屏幕我们的css就会增加多少倍。实际上在这里，我们采用的是js和css熟悉rem来解决这个问题的。

REM属性指的是相对于根元素设置某个元素的字体大小。它同时也可以用作作为设置高度等一系列可以用px来标注的单位。

```
html {  
  font-size: 10px;  
}  
  
div {  
  font-size: 1rem;  
  height: 2rem;  
  width: 3rem;  
  border: .1rem solid #000;  
}
```

采用以上写法，div继承到了html节点的font-size，为本身定义了一系列样式属性，此时1em计算为10px，即根节点的font-size值。所以，这时div的高度就是20px，宽度是30px，边框是1px，字体大小则是10px；一旦有了这样的方法，我们自然可以根据不同的屏幕宽度设置不同的根节点字体大小。假设我们现在设计的标准是iphone5s，iphone5系列的屏幕分辨率是640。为了统一规范，我们将iphone5 分辨率下的根元素font-size设置为100px；

```
<!--iphone5-->
```

```
html {
  font-size: 100px;
}
```

那么以此为基准，可以计算出一个比例值6.4。我们可以得知其他手机分辨率的设备下根元素字体大小：

```
/*
```

数据计算公式 $640/100 = device-width / x$ 可以设置其他设备根元素字体大小

iphone5: 640 : 100

iphone6: 750 : 117

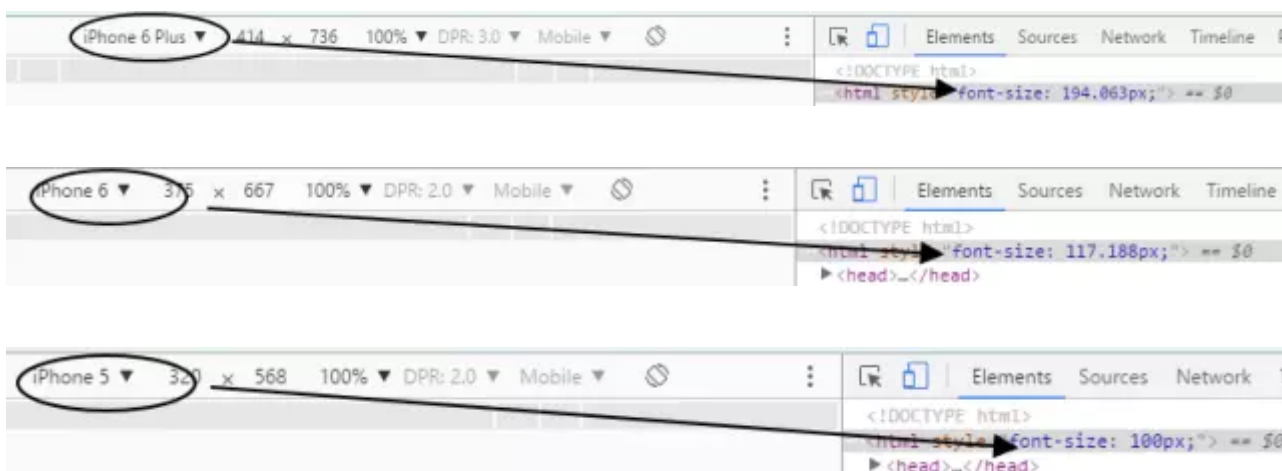
iphone6s: 1240 : 194

```
*/
```

```
var deviceWidth = window.documentElement.clientWidth;
```

```
document.documentElement.style.fontSize = (deviceWidth / 6.4) + 'px';
```

在head中，我们将以上代码加入，动态地改变根节点的font-size值，得到如下结果：



接下来我们可以根据根元素的字体大小用rem设置各种属性的相对值。当然，如果是移动设备，屏幕会有一个上下限制，我们可以控制分辨率在某个范围内，超过了该范围，我们就不再增加根元素的字体大小了：

```
var deviceWidth = document.documentElement.clientWidth > 1300 ? 1300 : document.documentElement.clientWidth;
```

```
document.documentElement.style.fontSize = (deviceWidth / 6.4) + 'px';
```

一般的情况下，你是不需要考虑屏幕动态地拉伸和收缩。当然，假如用户开启了转屏设置，在网页加载之后改变了屏幕的宽度，那么我们就需要考虑这个问题了。解决此问题也很简单，监听屏幕的变化就可以做到动态切换元素样式：

```
window.onresize = function(){  
    var deviceWidth = document.documentElement.clientWidth > 1300 ? 1300 : document.documentElement.clientWidth;  
    document.documentElement.style.fontSize = (deviceWidth / 6.4) + 'px';  
};
```

为了提高性能，让代码开起来更加完美，可以为它加上节流阀函数：

```
window.onresize = _debounce(function() {  
    var deviceWidth = document.documentElement.clientWidth > 1300 ? 1300 : document.documentElement.clientWidth;  
    document.documentElement.style.fontSize = (deviceWidth / 6.4) + 'px';  
}, 50);
```

顺带解决高保真标注与实际开发值比例问题

如果你们设计稿标准是iphone5，那么拿到设计稿的时候一定会发现，完全不能按照高保真上的标注来写css，而是将各个值取半，这是因为移动设备分辨率不一样。设计师们是在真实的iphone5机器上做的标注，而iphone5系列的分辨率是640，实际上我们在开发只需要按照320的标准来。为了节省时间，不至于每次都需要将标注取半，我们可以将整个网页缩放比例，模拟提高分辨率。这个做法很简单，为不同的设备设置不同的meta即可：

```
var scale = 1 / devicePixelRatio;  
document.querySelector('meta[name="viewport"]').setAttribute('content', 'initial-scale=' + scale + ', maximum-scale=' + scale + ', minimum-scale=' + scale + ', user-scalable=no');
```

这样设置同样可以解决在安卓机器下1px像素看起来过粗的问题，因为在像素为1px的安卓下机器下，边框的1px被压缩成了0.5px了。总之是一劳永逸！淘宝和网易新闻的手机web端就是采用以上这种方式，自适应各种设备屏幕的，大家有兴趣可以去参考参考。下面是完整的代码：

```
<!DOCTYPE html>  
<html>  
<head>  
    <title>测试</title>  
    <meta name="viewport" content="width=device-width,user-scalable=no,maximum-scale=1" />
```

```

<script type="text/javascript">
(function() {
    // devicePixelRatio : 设备像素
    var scale = 1 / devicePixelRatio;
    //设置meta 压缩界面 模拟设备的高分辨率
    document.querySelector('meta[name="viewport"]').setAttribute('content', 'initial-scale=' + scale + ', maximum-
scale=' + scale + ', minimum-scale=' + scale + ', user-scalable=no');
    //debounce 为节流函数, 自己实现。或者引入underscource即可。
    var reSize = _.debounce(function() {
        var deviceWidth = document.documentElement.clientWidth > 1300 ? 1300 : document.documentElement.client
Width;
        //按照640像素下字体为100px的标准来, 得到一个字体缩放比例值 6.4
        document.documentElement.style.fontSize = (deviceWidth / 6.4) + 'px';
    }, 50);

    window.onresize = reSize;
})();
</script>
<style type="text/css">
html {
    height: 100%;
    width: 100%;
    overflow: hidden;
    font-size: 16px;
}

div {
    height: 0.5rem;
    widows: 0.5rem;
    border: 0.01rem solid #19a39e;
}

.....
</style>
<body>
    <div>
    </div>
</body>
</html>

```

让元素飞起来-媒体查询

运用css新属性media query 特性也可以实现我们上说到过的布局样式。为尺寸设置根元素字体大小:

```
@media screen and (device-width: 640px) { /*iphone4/iphon5*/  
    html {  
        font-size: 100px;  
    }  
}  
  
@media screen and (device-width: 750px) { /*iphone6*/  
    html {  
        font-size: 117.188px;  
    }  
}  
  
@media screen and (device-width: 1240px) { /*iphone6s*/  
    html {  
        font-size: 194.063px;  
    }  
}
```

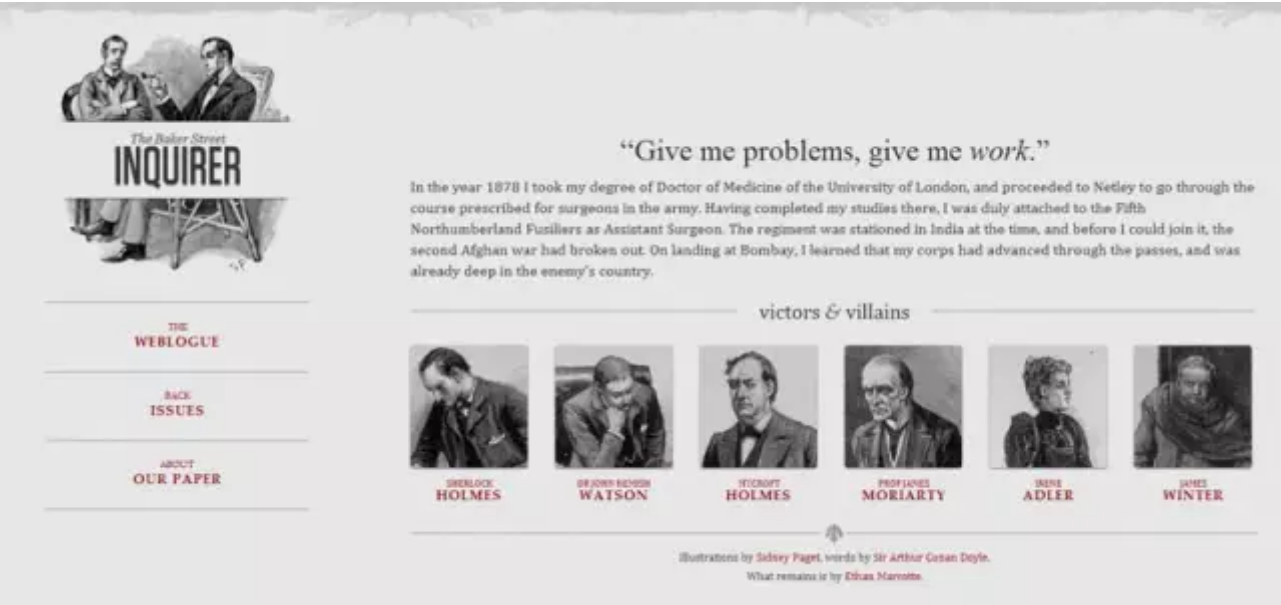
这种方式也是可行的，缺点是灵活性不高，取每个设备的精确值需要自己去计算，所以只能取范围值。考虑设备屏幕众多，分辨率也参差不齐，把每一种机型的css代码写出来是不太可能的。但是它也有优点，就是无需监听浏览器的窗口变化，它会跟随屏幕动态变化。媒体查询的用法当然不仅仅像在此处这么简单，相对于第二种自适应来说有很多地方是前者所远远不及的。最明显的就是它可以根据不同设备显示不同的布局样式！请注意，这里已经不是改变字体和高度那么简单了，它直接改变的是布局样式！

```
@media screen and (min-width: 320px) and (max-width: 650px) { /*手机*/  
    .class {  
        float: left;  
    }  
}  
  
@media screen and (min-width: 650px) and (max-width: 980px) { /*pad*/  
    .class {  
        float: right;  
    }  
}  
  
@media screen and (min-width: 980px) and (max-width: 1240px) { /*pc*/  
    .class {  
        float: clear;  
    }  
}
```


```
}  
}
```

此种自适应布局一般常用在兼容PC和手机设备，由于屏幕跨度很大，界面的元素以及远远不是改大小所能满足的。这时候需要重新设计整界面的布局和排版了：


如果屏幕宽度大于1300像素



如果屏幕宽度在600像素到1300像素之间，则6张图片分成两行。



The Baker Street
INQUIRER



THE
WEBLOGUE


BACK
ISSUES

ABOUT
OUR PAPER


“Give me problems, give me *work*.”

In the year 1878 I took my degree of Doctor of Medicine of the University of London, and proceeded to Netley to go through the course prescribed for surgeons in the army. Having completed my studies there, I was duly attached to the Fifth Northumberland Fusiliers as Assistant Surgeon. The regiment was stationed in India at the time, and before I could join it, the second Afghan war had broken out. On landing at Bombay, I learned that my corps had advanced through the passes, and was already deep in the enemy's country.


victors & villains




SHERLOCK
HOLMES




DR. JOHN HEMISH
WATSON




MYCROFT
HOLMES



PROF. JAMES
MORIARTY



IRENE
ADLER



JAMES
WINTER

Illustrations by Sidney Paget. words by Sir Arthur Conan Doyle.
What remains is by Ethan Marcotte.

如果屏幕宽度在400像素到600像素之间，则导航栏移到网页头部。



许多css框架经常用到这样的多端解决方案，著名的bootstrap就是采用此种方式进行栅格布局的。

总结

不管哪一种自适应方式，我们的目的是使得开发网页在各种屏幕下变得好看：如果你的项目定位的用户群仅仅是使用某种机型的人，那么可以采用第一种自适应方式。如果你的客户主要是移动端，但是客户的设备类型庞杂，建议采用第二种方式。如果你雄心勃勃地需要建立一套兼容PC、

PAD、mobile多端的一体化web应用，那么第三种选择显然是最适合你的。每种方式都有自己的利弊，根据需求权衡利害，合理地实现自适应布局，需要不停的实践和摸索。路漫漫其修远兮，吾将上下而求索。

参考资料

- 自适应网页设计 (Responsive Web Design)
- 移动前端自适应解决方案和比较
- 移动web适配利器-rem

觉得本文对你有帮助？请分享给更多人
关注「前端大全」，提升前端技能

前端大全

分享前端相关技术干货·资讯·高薪职位·教程



微信号：FrontDev



长按识别二维码关注

伯乐在线 旗下微信公众号

商务合作QQ：2302462408

[阅读原文](#)