**移动端适配**

# 一 百分比适配。

设计稿1080，某一个元素距离顶部200px，写死top:200px/margin-top:200px。现在转换成百分比，200/1080=18.51%。

流体布局下，百分比的值是根据父层计算的；absolute布局的元素是根据最近的relative父层计算的；fixed布局的元素是根据window的可视区域计算的。

所以，如果要达到适配的效果，不只是当前元素需要用百分比值，其参考计算的元素也是要动态变化的。

**二rem适配**

rem适配是改变一个元素在不同设备上占据的CSS像素个数

1. 使用完美视口

<meta name="viewport" content="width=device-width,initial-scale=1.0,minimum-scale=1.0,maximum-scale=1.0">

1. 使用js改变网页的font-size。

Var styleNode = document.createElement（"style"）；

Var w = document.documentElement.clientWidth/16;

styleNode.innerHtml = "html{fontsize:"+w+"px!important}";

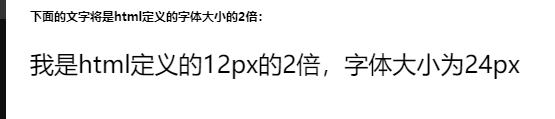
document.head.appendChild("styleNode")

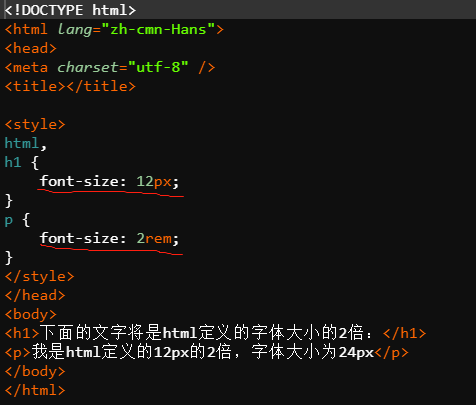
## [flexible.js如何实现rem自适应](http://caibaojian.com/flexible-js-2.html)

1. flexible.js正是利用rem单位相对根元素<html>的font-size来做计算，而我们需要做的就是根据不同的屏幕算出html的font-size，而页面内的大小单位都根据rem来写，从而实现了自适应。

rem: 相对长度单位，相对于根元素(即html元素)[font-size](https://www.html.cn/book/css/properties/font/font-size.htm)计算值的倍数

浏览器默认的字号16px。如果css里面没有设定html的font-size，则默认浏览器以1rem=16px来换算。





# It管家应用实例

1. rem等于多少px呢？

html { font-size:14px }

那么后面的CSS里面的rem值则是以这个14来换算，例如设定一个div宽度为3rem,高度为2.5rem.则它换算成px为width:42px.height:35px，同理，假如一个设计稿为宽度42px,高度为35px，则换成成rem，则是42/14=3rem，35/14=2.5rem。

为了方便计算，时常将在<html>元素中设置font-size值为62.5%:相当于在<html>中设置font-size为10px。

1. 字体自适应。先定义全局的字体大小，然后使用rem单独定义不同文件的字体大小。

如果需要适配不同屏幕的话，使用媒体查询改变全局自定义的字体即可。

\* {  
 padding: 0;  
 margin: 0;  
}  
  
html {  
 font-size: 62.5%;  
}  
body {  
 font-size: 62.5%;  
}  
@media only screen and (min-width: 320px) and (max-width: 340px) {  
 html {  
 font-size: 55%;  
 }  
 body {  
 font-size: 55%;  
 }  
}

.FontTitle{  
  
 text-align: center;  
  
 font-size: 2.0rem;  
  
 font-weight: bold;  
}  
.FontTitle img{  
  
 width: 43px;  
}

1. 优点：字体大小适配可以，但是div 布局啥的适应不是很完美。