## 移动端相关

### 1. 移动端1px边框问题的解决方案

1. 使用小数写px值 通过媒体查询判断设备的像素比 DRP 根据不同像素比写不同的边框大小

.border { border: 1px solid #ccc }  
@media screen and (-webkit-min-device-pixel-ratio: 2) {  
 .border { border: 0.5px solid #ccc }  
}  
@media screen and (-webkit-min-device-pixel-ratio: 3) {  
 .border { border: 0.333333px solid #ccc }  
}

1. 使用边框图片

这样的1张6X6的图片, 9宫格等分填充border-image, 这样元素的4个边框宽度都只有1px

@media screen and (-webkit-min-device-pixel-ratio: 2){   
 .border{   
 border: 1px solid transparent;  
 border-image: url(border.gif) 2 repeat;  
 }  
}

1. 使用CSS3 box-shadow

* .shadow {  
  -webkit-box-shadow:0 1px 1px -1px rgba(255, 0, 0, 0.5);  
  box-shadow:0 1px 1px -1px rgba(255, 0, 0, 0.5);  
  }

1. viewport结合rem (推荐使用)

//devicePixelRatio=2设置meta  
<meta name="viewport" content="initial-scale=0.5, maximum-scale=0.5, minimum-scale=0.5, user-scalable=no">  
  
//devicePixelRatio=3设置meta  
<meta name="viewport" content="initial-scale=0.3333333333333333, maximum-scale=0.3333333333333333, minimum-scale=0.3333333333333333, user-scalable=no">

1. 使用 :before , :after 与 transform（推荐使用）

//第一种方法  
//构建1个伪元素, 将它的长宽放大到2倍, 边框宽度设置为1px, 再以transform缩放到50%  
.radius-border{  
 position: relative;  
}  
@media screen and (-webkit-min-device-pixel-ratio: 2){  
 .radius-border:before{  
 content: "";  
 pointer-events: none; /\* 防止点击触发 \*/  
 box-sizing: border-box;  
 position: absolute;  
 width: 200%;  
 height: 200%;  
 left: 0;  
 top: 0;  
 border-radius: 8px;  
 border:1px solid #999;  
 -webkit-transform(scale(0.5));  
 -webkit-transform-origin: 0 0;  
 transform(scale(0.5));  
 transform-origin: 0 0;  
 }  
}  
  
@media screen and (-webkit-min-device-pixel-ratio: 3) {  
 .radius-border:before{  
 content: "";  
 pointer-events: none; /\* 防止点击触发 \*/  
 box-sizing: border-box;  
 position: absolute;  
 width: 200%;  
 height: 200%;  
 left: 0;  
 top: 0;  
 border-radius: 8px;  
 border:1px solid #999;  
 -webkit-transform(scale(0.3333));  
 -webkit-transform-origin: 0 0;  
 transform(scale(0.3333));  
 transform-origin: 0 0;  
 }  
}

### 2. 移动屏幕适配方案

1. 使用rem+媒体查询实现
2. 使用rem+js实现
3. 使用rem+vw实现

rem是什么是一个相对单位参照根元素的字体大小 只要动态修改跟元素字体大小所有rem单位都会跟着改变 那么rem的根元素改变参照点是谁： 设计图 以设计图大小对应的的屏幕作为参照点 假设一个设计图大小对应的真实屏幕的根元素值 例如750设计图真实屏幕是375 根元素100px 计算其他屏幕就使用 其他屏幕大小/375 \* 100 求到其他屏幕的根元素大小