# **移动端适配**

**移动端适配的目的是为了将视觉稿等比例还原到不同的机型上，又由于移动设备有和桌面端不同的概念，需要针对性的设置才能正确的还原。**

1、由于移动端可以缩放，就造成了视觉视口也会相应的缩放，从而导致呈现的内容会缩放；

2、iPhone首先提出Retina屏幕，造成了设备的实际物理像素是设备的逻辑像素的2倍或者3倍，也造成了同样的1独立像素，在不同的机型上需要4个像素或者9个像素，也就是1px无法正常呈现1px，造成不同的机型图片的模糊、同样的字体大小展示不一致

3、由于iPhone的刘海屏、底部小黑条，又有了安全区的概念

4、更多的操作比如长按屏幕选中，长按图片提示保存等

## 视觉还原

通过视觉稿的分辨率，1:1还原呈现到相应分辨率的屏幕上，对于不一致的分辨率设别，能够按比例的缩放呈现，这是移动端适配的基本要求，相应的w3c标准也设计了这块的接口，那就是rem和vw/vh。

只所以不用百分比，那是由于不同的css属性，使用百分比作为值并没有一个统一的基准，比如font-size百分比是基于父元素字体大小，height百分比是基于相对包含块的大小，而如果包含块没有设置height高度值，子元素的height百分比是没有作用的。

### 1.1 rem设置

rem的含义是表示html根元素的字体大小，如果整个页面都是用rem, 那么只需要设置html的font-size值，就可以等比例的缩放适配页面。

以视觉稿一倍分辨率375x812（标准）为例，通过postcss插件将100px设置为1rem, 即所有的尺寸值都是px值/100 rem。此时将整个页面呈现在414x896的屏幕上，只需要

descript

descript

所以对于414px宽度的屏幕，我们需要将其html元素的font-size设置为110.4px

以上公式转换为代码即为

window.onresize = () => {

document.documentElement.style.fontSize = document.documentElement.clientWidth / 3.75

}

由于我们需要监听onresize, 而在移动端键盘弹出的时候会触发onresize回调，这块就需要特殊处理，所幸我们的代码完全可以使用vw/vh

### 1.2 vw/vh设置

1vw是设备的宽度的百分之一，1vh是设备高度的百分之一。以上面为例，我们在还原视觉稿的时候相当于是认为1rem = 100px = html的font-size。如果使用vw/vh，就是将html的font-size设置为 未知数vw, 如果我们假定当前设备屏幕为414，那么1vw就是4.14, 那么110.4= 4.14 \* 26.66, 其实本质上公式为

descript

descript

使用了vw/vh就不需要考虑屏幕resize的情况，只需要考虑横竖屏切换，这里可以通过css的媒体查询设置

html {

font-size: 100px;

font-size: 26.66vw;

}

@media (orientation: landscape) {

html {

font-size: 100px;

font-size: 26.66vh;

}

}

以上就是视觉还原的设置内容，不过其中隐藏了一个共识的点就是高度的值是按照宽度计算的，也就是表示按照视觉稿设置的高度值会大于屏幕的高度，但是宽度值是不会超出屏幕的宽度的。而对于我们的业务来说，横屏下高度不能超过屏幕的高度，所以使用的是vh

## 移动端缩放

移动端的手势会导致页面视口产生变化, 所以需要在html的header中设置meta

<meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1,user-scalable=0,viewport-fit=cover" />

布局视口的宽高和设备宽高一致，user-scalable表示禁止手势缩放，initial-scale、maximum-scale是指设备尺寸和可视视口的比例，用来调节可视视口的，viewport-fit是针对iPhone的安全区的设置

## 三、Retina屏幕

iPhone首先提出将设备的分辨率提高到超过人眼视网膜识别到极限的值，使得屏幕的图形展示更加逼真，不过对于web页面来说，由于我们设置的px都是设备的逻辑像素，物理设备会将逻辑像素转变成物理像素，对于2倍屏幕来说，x、y轴两个方向各放大2倍，面积放大4倍，也就是1px的图片渲染到屏幕上的4个像素位置，自然也就使得图片模糊。

浏览器中返回物理像素/逻辑像素比值的接口就是

window.devicePixelRatio

如果我们希望2倍屏设备上 代码中的1px对应到屏幕的1px， 有两种途径

1. initial-scale、maximum-scale是设备的小的边和可视视口的比例，一般设置为1，也就是viewport的大小和设备的大小是一样的，如果希望1逻辑像素=1物理像素，只需要缩放viewport的大小，比如initial-scale=0.5、maximum-scale=0.5就是针对2倍屏幕的, 只不过此时字体使用px，对于高倍屏需要放大2倍。
2. 只处理图片和字体大小，图片使用2倍图，字体缩小原先的1/2

### 第一种方案

会造成viewport扩大了2倍或者3倍，影响开发体验，使用px的字体也需要在高倍屏上放大，而且针对我们的业务如果这样设置，canvas就不能二次处理window.devicePixelRatio

具体代码如下

html header中添加

<script>

const meta = document.createElement('meta')

metaEle.name = 'viewport'

const content = 'width=device-width,initial-scale='+(1/window.devicePixelRatio).toFixed(2) + ',maximum-scale='+ (1/window.devicePixelRatio).toFixed(2) + ',user-scalable=0,viewport-fit=cover';

metaEle.content = content

document.head.appendChild(metaEle)

</script>

在main.tsx中添加

document.body.classList.add(`dpr${window.devicePixelRatio}`)

对于用到的字体引用如下less函数

.font-size(@val) {

font-size: @val;

.dpr2 & {

font-size: (@val \* 2);

}

.dpr3 & {

font-size: (@val \* 3);  
 }

}

### 第二种方案

initial-scale和maximum-scale都是1， 字体缩放可以将整个元素使用transform，或者正常的值/2，代码如下

main.tsx中添加

document.body.classList.add(`dpr${window.devicePixelRatio}`)

对于用到的字体引用改less函数

.font-size(@val) {

font-size: @val;

.dpr2 &, .dpr3 & {

// chrome最小字号是12px,所以针对3倍屏保持和2倍一样的效果

font-size: (@val / 2);

}

}

对于1px下划线这类问题，直接在css中引用下方less函数

.scale() {

transform: scale(1);

.dpr2 & {

transform: scale(0.5);

}

.dpr3 & {

transform: scale(0.3);

}

}

## 我们的项目选择方案二

## iPhone安全区

iPhone有四个安全区

safe-area-inset-top

safe-area-inset-bottom

safe-area-inset-left

safe-area-inset-right

基本对于我们来说，横屏下需要考虑的是bottom和left。设置如下

.safearea-padding(@direction, @value) {

padding-@{direction}: @value;

@safe-const: ~'constant(safe-area-inset-@{direction})';

@safe-env: ~'env(safe-area-inset-@{direction})';

/\* stylelint-disable-next-line function-no-unknown \*/

padding-@{direction}: calc(@value + @safe-const);

padding-@{direction}: calc(@value + @safe-env);

}

## 其他

### 4.1 图片长按出现保存

可以通过给img标签使用div包裹，设置div的宽高为图片展示的高度，图片宽高为100%，对于点击事件，绑定到包裹img的div元素上，img全局设置

img {

user-select: none;

pointer-events: none;

}

对于需要该功能图片自己再设置回去

### 4.2 长按选中文本

html, body {

user-select: none;

}