

一、前端尺寸单位

vw、vh、rem、em

em

em 是一个相对长度单位。其相对于当前对象内文本的字体尺寸，如当前对行内文本的字体尺寸未被人为设置，则相对于浏览器的默认字体尺寸。

- em的值并不是固定的
- em会继承父级元素的字体大小

rem

rem全名root em，简写rem，故其也是一个相对长度单位，但只相对于根元素，可以简单的通过更改根元素大小，从而调整所有字体大小。

- 只相对于根元素 (html)
- 通过修改根元素可成比例的调整页面字体大小
- 其适配方案通过js脚本设置像素点来实现

vw、vh

vw (Viewport Width)、vh(Viewport Height)是基于视图窗口的单位，是css3的一部分，基于视图窗口的单位。

- vw:1vw 等于视口宽度的1%
- Vh:1vh 等于视口高度的1%

二、动态计算rem

1. js根据屏幕宽度和设计图动态设置html的font-size

```
1 function setRem() {  
2     var cwidth = document.documentElement.clientWidth;  
3     // 计算当前宽度和设计图的比例,  
4     var scale = cwidth / 375;  
5     // 设置页面 1rem的大小  
6     document.documentElement.style.fontSize = 100 * scale + 'px';  
7 }  
8 setRem()  
9 // 每次窗口改变都重新设置一下rem  
10 window.onresize = setRem;
```

2. scss中使用vw计算rem

```
1 // 设计图 750px下 100px === 1rem,比例为 100: 1  
2 // 1vw = 750 / 100 = 7.5px  
3 // 100px = 100vw / (750 / 100) = 13.3333333vw  
4 // html {  
5     // font-size: 100vw / (750 / 100)  
6 // }  
7  
8 // 标准设计图  
9 $design_size: 375;  
10 // html默认的font-size  
11 $base_font: 75;  
12 // 定义方法把设计图的px专成rem  
13 @function rem($px) {  
14     @return $px / $base_font * 1rem;  
15 }  
16  
17 html {  
18     // 设置页面1rem的值
```

```
19     font-size: 100vw / ($design_size / $base_font);
20 }
21 // 设置320px以下都是用320px尺寸展示
22 @media screen and (max-width: 320px) {
23     html {
24         font-size: 64px;
25     }
26 }
27 // 设置540px以上都是用540px的尺寸展示
28 @media screen and (min-width: 540px) {
29     html {
30         font-size: 108px;
31     }
32 }
33 // 把页面所有的字体默认大小改成14px
34 body {
35     font-size: 14px;
36 }
37 h2 {
38     // 375尺寸的手机下为45px，其他设备下等比例缩放
39     height: rem(45);
40     background: red;
41 }
```

3. 使用flexible.js