

基于vw等viewport视区单位配合rem响应式排版和布局

这篇文章发布于 2016年08月8日, 星期一, 23:49, 归类于 [CSS相关](#)。阅读 50802 次, 今日 34 次 [56 条评论](#)

by zhangxinxu from <http://www.zhangxinxu.com/wordpress/?p=5583>

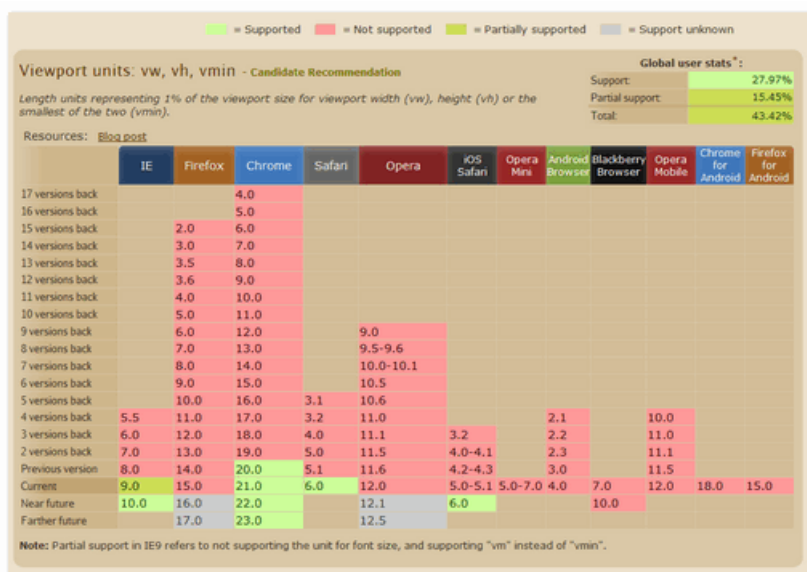
本文全文转载需购买版权(500¥), 摘要引流则免费, 具体参见[这里](#)

一、什么是视区相对单位?

就是相对于浏览器viewport尺寸的单位, 具体包括下面4个:

- vw – 视区宽度百分值
- vh – 视区高度百分值
- vmin – vw或vh, 取小的那个
- vmax – vw或vh, 取大的那个

至于更具体的内容, 我就不展开了, 因为我发现已经有一个很厉害的人对这些单位作为很详细的介绍了, 是超详细, 这个人超能写。而且一看发表日期, 2012年9月, 我勒个擦, 这不是上届伦敦奥运会的时候嘛, 足足有4年了呀, 也太前沿了吧, 那个时候的浏览器兼容性还只是这幅鸟样子:



好东西呢是要分享滴, 这篇文章就是: [“视区相关单位vw, vh, vmin简介以及可实际应用场景”](#), 速速围观, 就算不学知识, 也看感受下这篇文章的气场。

大家有没有发现,, 这个很厉害的人原来就是我本人..... 😊



4年时光, 物是人非, [vw](#) 头上的草叶越来越绿了, 现在的兼容性则是:

Viewport units: vw, vh, vmin, vmax - CR

Length units representing 1% of the viewport size for viewport width (vw), height (vh), the smaller of the two (vmin), or the larger of the two (vmax).

Current aligned		Usage relative		Show all					
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android *	Chrome for Android
			29						
			49					4.3	
			50					4.4	
8	13	47	51			9.2		4.4.4	
11	14	48	52	9.1	39	9.3	all	51	51
		49	53	10	40				
		50	54	TP	41				
		51	55						

确实绿了很多，Android 4.4开始支持，前段时间给老爹买的华为机子默认都是5.*版本，看上去，vw 在移动端的春天要来哈~

CSS3中还有另外一个单位rem，估计有不少小伙伴耳闻或者用过，为何rem 要比vw 火呢，因为rem 从Android 2.1版本就开始支持了，兼容性好。一旦兼容性不是问题，vw 也会火起来的，就跟薛之谦一样。

二、基于vw的响应式排版和布局

一般而言，屏幕越大，我们希望文字大小也越大，元素的尺寸也能等比例放大，尤其在现在的网页设计风格中。

如果来实现这种弹性自适应效果，目前主流的实现是通过设定根元素的font-size 大小，具体元素或模块使用rem 或em 单位来实现。

一种是直接设定一个临界点字体大小，如：

```
html {
  font-size: 16px;
}
@media screen and (min-width: 600px) {
  html {
    font-size: 18px;
  }
}
@media screen and (min-width: 1000px) {
  html {
    font-size: 22px;
  }
}
```

还有一种就是使用JS在resize或者屏幕旋转的时候，动态修改root的font-size 大小。

前面一种基于 @media 的CSS实现问题在于，内容的弹性自适应只会在临界点的时候，“Duang”变化下，于是，我们浏览器尺寸拉伸的时候，会感受到类似“噔噔噔”卡壳的效果，感觉就像是吃了含笑半步癫：



使用JS的问题在于他是JS，要保证加载体验，需要头部内联，为了保证实时性，需要多个浏览器变化事件监测，用我的口头禅讲就是“显得啰嗦”。

那有没有两全其美的方法呢？

有，那就是本文要隆重请出场的 vw，配合CSS3 calc 计算实现动态字体大小效果。

例如，我们希望浏览器宽度在600px~1000px变化的时候，html根元素的 font-size 大小是18px~22px之间对应变化的，则可以：

```
html { font-size: calc(18px + 4 * (100vw - 600px) / 400); }
```

当视区宽度是 600px 的时候， 100vw 就等于 600px，于是：

```
18px + 4 * (100vw - 600px) / 400
↓
18px + 4 * (600px - 600px) / 400
↓
18px
```

当视区宽度是 1000px 的时候， 100vw 就等于 1000px，于是：

```
18px + 4 * (100vw - 600px) / 400
↓
18px + 4 * (1000px - 600px) / 400
↓
18px + 4px
↓
22px
```

于是，理论上，一个18px~22px字体大小动态匹配的布局基础就建好了。

一般出现“理论上”这三个字，就说明我要“放坑”了，大家闪开~~

此“坑”就是苹果系统的Safari浏览器不认识上面的属性值（如下截图）：

就Safari不行，window下的所有浏览器，包括IE，以及mac下的Chrome等浏览器都是可以正确解析的。

我这篇文章的前一篇文章“[Safari 3D transform变换z-index层级渲染异常的研究](#)”就是吐槽“Safari是新时代的IE6”来着，没想到一日不见，这“新时代IE6”的名声是坐的越来越夯实了哈！

好在，我们还是有机会曲线救国的。那就是，我们基础字体大小不使用像素单位，使用百分比单位即可

，如下：

```
html { font-size: calc(112.5% + 4 * (100vw - 600px) / 400); }
```

这个包括Safari在内的浏览器，都能够按照我们的预期计算 `font-size` 大小了，于是，我们一番倒腾，就有了如下的响应式设置：

```
html {  
    /* iPhone6的375px尺寸作为16px基准，600px正好18px大小 */  
    font-size: calc(100% + 2 * (100vw - 375px) / 225);  
}  
@media screen and (min-width: 600px) {  
    html {  
        /* 600px-1000px每100像素宽字体增加1px(18px-22px) */  
        font-size: calc(112.5% + 4 * (100vw - 600px) / 400);  
    }  
}  
@media screen and (min-width: 1000px) {  
    html {  
        /* 1000px往后是每100像素0.5px增加 */  
        font-size: calc(137.5% + 5 * (100vw - 1000px) / 1000);  
    }  
}
```

有个动态的根字体大小，我们就可以使用 `rem` 或 `em` 这些相对单位，来让我们的页面排版和布局更富有弹性。

您可以狠狠地点击这里：[基于vw的弹性布局小演示小demo](#)

结果在大尺寸下：



小尺寸下：



对比会发现，文字的大小和图片的大小都明显不一样了，但是，最终给人的排版的感觉却没有有什么不舒服的，因为，你仔细看，就会发现，虽然字号差别很大，但是文字的换行位置居然都是一模一样的，这活脱脱等比例缩放的即视感。你说，这么洒脱的效果，什么手机各种分辨率适配还是问题吗？是问题吗？

对吧，直接变成了毛毛雨，几行CSS，全部搞定！如果你足够懒，甚至可以本文的CSS代码直接copy到自己项目中用，注释都不需要改，也是可以的！

这么低的成本，这么好的体验，还不赶快在自己的项目里试试？😁

更新于2017年2月8日

经过大型项目实践，下面这段CSS是最好的基于 `rem` 和 `vm` 和 `calc` 实践代码：

```
html {  
    font-size: 16px;  
}  
  
@media screen and (min-width: 375px) {  
    html {  
        /* iPhone6的375px尺寸作为16px基准，414px正好18px大小，600 20px */  
        font-size: calc(100% + 2 * (100vw - 375px) / 39);  
    }  
}
```

```

        font-size: calc(16px + 2 * (100vw - 375px) / 39);
    }
}
@media screen and (min-width: 414px) {
    html {
        /* 414px-1000px每100像素宽字体增加1px(18px-22px) */
        font-size: calc(112.5% + 4 * (100vw - 414px) / 586);
        font-size: calc(18px + 4 * (100vw - 414px) / 586);
    }
}
@media screen and (min-width: 600px) {
    html {
        /* 600px-1000px每100像素宽字体增加1px(20px-24px) */
        font-size: calc(125% + 4 * (100vw - 600px) / 400);
        font-size: calc(20px + 4 * (100vw - 600px) / 400);
    }
}
@media screen and (min-width: 1000px) {
    html {
        /* 1000px往后是每100像素0.5px增加 */
        font-size: calc(137.5% + 6 * (100vw - 1000px) / 1000);
        font-size: calc(22px + 6 * (100vw - 1000px) / 1000);
    }
}
}

```

然后，就可以愉快地使用 `rem` 单位用来排版和布局啦！

三、结束语

对于大厂，Android4.4之前的版本，目前怕是还是要兼顾的，怎么办？我觉得没有必要为了一个老奴而放弃整个后宫，对于绝大部分设备，我们还是 `vm` + `calc` 哗啦啦地走起，老的设备，我们可以使用以前JS的策略，打个脚本补丁，估计十来行代码，效果都是一样的，CSS代码也完全复用，一点都没必要特异处理，因为老机子不认识 `vw`，所以完全没必要担心新老冲突这样的问题。

当然，我自己目前也没在正式的to C项目中实践过，所以，或许还有其他坑什么的，Android设备那么多，谁知道哪个机子的浏览器突然就抽风了呢！

但是，这并不妨碍我大胆预测：再过1~2年，基于 `vm` 布局的各种文章、各种事件案例就会雨后春笋般出现……

好了，好像没有什么其他要说的了。

感谢阅读，欢迎交流！



《CSS世界》签名版独家发售，包邮，可指定寄语，点击显示购买码

（本篇完） // 想要打赏？[点击这里](#)。有话要说？[点击这里](#)。



猜你喜欢

- 理解SVG viewport,viewBox,preserveAspectRatio缩放
- 基于screen.width的伪响应式开发
- 小tip:CSS vw让overflow:auto页面滚动条出现时不跳动
- 博闻强识: 了解CSS中的@ AT规则
- 5分钟快速了解下CSS4 color-adjust属性
- 视区相关单位vw, vh..简介以及可实际应用场景
- 等宽字体在web布局中应用以及CSS3 ch单位嘿嘿
- CSS百分比padding实现比例固定图片自适应布局
- jQuery之replace字符串替换实现不同尺寸图片切换
- 翻译 - CSS继承详解
- CSS前景背景自动配色技术简介

分享到:         { 0 }

标签: @media, calc, font-size, rem, viewport, vw, 响应式布局, 相对单位

发表评论（目前56条评论）

<input type="text"/>	名称 (必须)
<input type="text"/>	邮件地址(不会被公开) (必须)
<input type="text"/>	网站
<div></div>	
<input type="button" value="提交评论"/>	

1. onepixel说道:

2018年09月3日 17:00

大神,我发先另一个办法,不需要用 media query,只需下面一句话就可以实现rem自适应方案:

```
html {  
font-size: calc(20 / 375 * 100vw);  
}
```

这里的375 和 20 指的是, 在 iphone6 的设计稿上, 1rem=20px (@2x图)

[回复](#)



2. 海说道:

2018年05月23日 10:26

每个元素的rem值,是怎么算出来的..

设计稿是750px; 有一个75px的div, 那他的rem应该是多少?



[回复](#)

3. 海说道：

2018年03月23日 15:20

还行, 马马虎虎, 联合vw!这么用rem, 那你至flex布局于何地?

[回复](#)



4. html基数设定说道：

2018年01月9日 23:00

你的demo中设定html{font-size:16px}，那么我要是设定12px呢，里面的媒体查询应该怎么办呢？基数是12，又该怎么处理呢？

[回复](#)



5. 似水沐雨说道：

2017年09月14日 09:50

张大大，为什么有的说，calc()最好不要用，效率不高呢。

[回复](#)



张鑫旭说道：

2017年09月15日 23:10

我们通常的页面开发CSS的性能都是可以忽略不计的。

[回复](#)



6. 爱踩键盘的猫说道：

2017年08月12日 13:25

鑫爷，这篇文章可以转载到我的博客吗~

[回复](#)



7. 忌白说道：

2017年05月6日 13:27

font-size: calc(16px + 2 * (100vw - 375px) / 39);

font-size: calc(18px + 4 * (100vw - 414px) / 586);

这两个除数39 586有点不太明白。。。怎么得来的

[回复](#)



nicholasurey说道：

2017年05月10日 13:12

375+39=?

414+586=?

[回复](#)



abc说道：

2018年11月16日 11:40

375到414刚好是39个，(100vw - 375px) / 39的意思就是，当前宽度比375多的部分占总差值的比值，这个总差值39刚好对应过渡需要2px，所以再乘以2便是需要增加的像素数。超过414的往后接下一个媒体查询，414-1000刚好是586的差值，刚好需要分配4px，道理是一样的。

[回复](#)



8.

Juven说道:

2017年04月18日 14:28

用过62.5%, 也用过16px, 用过100px, 不过这个方法的弹性更佳。

现在有个疑问, 想咨询一下, 如果使用此方法后, 发现各个布局都使用了rem单位了, 如何去考虑将px换算成rem? 使用sass, 有针对此方法的换算没?

回复

高龙芝说道:

2017年05月24日 10:38

hbuilder里面有自动换算的 需要设置一下: 工具->选项->Hbuilder->代码助手设置.里面有个px自动转rem设置, 按自己的实际情况设置就可以在每次输入px的时候有提示了, sublime也有类似插件。

回复
9.

gemingcao说道:

2017年04月10日 17:21

rem可以设计成100px, 这样可以方便计算rem。

回复

xiaodou说道:

2018年01月10日 10:16

那你rem可以设计成100px, 那你font-size的值设定多少呢? 媒体查询里面的值又该怎么设定呢? font-size: calc(100% + 2 * (100vw - 375px) / 39); 这个39、586又是怎么来的呢?

回复
10.

点点小鱼干说道:

2017年04月5日 14:27

感觉如果浏览器可以支持vw的话, 直接写一个html{font-size: 3vw;}就可以实现等比缩放的效果额。

回复
11.

abaozhi说道:

2017年03月22日 10:10

感动啊

回复
12.

阿威说道:

2017年03月6日 23:11

如果是640的设计稿, 请问图片怎么处理? 设计图/32 rem?

回复
13.

px怎么转换为rem说道:

2017年02月18日 17:06

我拿到的设计图是750宽的, 写静态页的时候px怎么转换为rem

回复

nz说道:

2018年05月4日 21:39

我也是啊, 正在纠结, 请问你时怎么换算的。我的想法是: ip6物理像素375, 正好375px的屏幕宽设置的根元素是16px, 换算成rem就是: (设计图/16) * 1 rem; 实际像素是750的话, 就是2倍, 于是就是 (设计图/32) * 1 rem。不知道这样对不对

[回复](#)

14. **coderunner**说道:

2017年01月9日 17:38

大神, 我问一个问题, calc用多了会影响移动端性能是吗?

[回复](#)



张 鑫旭说道:

2017年01月12日 22:37

基本上, 大多数情况下CSS侧的渲染都是可以忽略不计的。

[回复](#)



15. **cyh41**说道:

2016年11月5日 15:10

font-size因为点阵的关系不是最好要偶数, 这种字号不是会影响阅读么?

[回复](#)



16. **龙二君**说道:

2016年11月5日 10:13

已经在用vw vh 布局了, 用了之后爱不释手, 话说我是在微信网页开发者提问里面看到有人问微信支不支持这种单位, 自己试了一下, 支持。

[回复](#)



17. **carryme15**说道:

2016年10月19日 10:27

$600/18=33 \rightarrow$ 600px的屏幕能容纳33个字符

$1000/22=45 \rightarrow$ 1000px的屏幕能容纳45个字符

如果使用上述的公式, 必然会导致页面布局发生变化。如果直接指定一个屏幕能容纳的字符数就会解决布局发生变化的问题 (font-size: 4vw)

中文字本来就是等宽字体, 天然的优势->例子: <http://jsbin.com/fofako/edit?html,css>

[回复](#)



18. **james**说道:

2016年10月12日 15:09

如果把CSS放到JS中, 可以直接使用JS来动态计算vw, vh, calc等。

可以看下这个类库: <https://github.com/cssobj/cssobj>

[回复](#)



19. **那小爷**说道:

2016年09月20日 13:21

旭小哥, 看你文章收益杆杆。超6超赞

[回复](#)



20. **死兔比**说道:

2016年09月12日 19:58

看了好久好久 你的文章 好像是几个月前 就开始看 只有一个感慨: 溜溜溜溜溜! 🐇

[回复](#)



21. **zzz87**说道:
2016年09月1日 12:35
- 最近发现这样处理非常方便
- ```
html{
font-size:5vw;
}
```
- 以此为基准，全局使用rem布局
- 5vw 在 320（iphone4/iphone5） 屏幕上面 刚好是16px，也就是1rem=16px
- 于是
- 在iphone4/iphone5 1rem=16px
- 在iphone6 1rem = 18.75px
- 在iphone6 plus 1rem = 20.7px
- 安卓上面也一样
- 基本上各种顺畅，一个设备屏幕布局好，其他设备 按比例自动伸缩。

[回复](#)

**Suscc**说道:  
2016年10月11日 00:22

那为何不直接拿vw当做单位 还要借助rem~

[回复](#)

**mss**说道:  
2016年11月8日 12:48

兼容不支持的

[回复](#)

**堂**说道:  
2018年07月12日 16:34

pc端试了吗，感觉挺暴力的

[回复](#)

22. **无敌小钢豆**说道:  
2016年08月26日 15:52
- 你的文章我可以当段子看了，真是边娱乐边学习啊

[回复](#)

23. **有没有尝试从viewport的角度出发**说道:  
2016年08月17日 11:13
- 如果通过设置viewport的做法呢
- 这样写了之后，我的页面完全按照640的宽度写死（当然750也可以，随意），字体大小根据设计稿是多少就写多少就完了，不需要什么rem，也不需要vm，那么在不同的屏幕下，会自动缩放啊，根本不需要关心 css层面的东西

[回复](#)

**迷途小前端**说道:  
2016年08月29日 17:06

握个手。。

[回复](#)

**迷途小前端**说道:  
2016年08月29日 17:07

meta name="viewport" content="width=640, initial-scale=1.0,user-scalable=no"，加尖括号竟然给过滤掉了。。

[回复](#)

jserb说道:

2016年10月25日 14:50

对是的,但是目前发现这个方法,某些奇葩的安卓机型会不兼容.viewport设置失效



bestRenecton说道:

2017年01月22日 14:51

这种方法你有遇到过,有些机子 字体突然失效,字体变很大的情况吗

[回复](#)

张 鑫旭说道:

2017年01月22日 16:34

@bestRenecton 建议不要使用viewport缩放,内联文本元素是无法缩放的。

[回复](#)



Coreyxu说道:

2017年03月31日 18:39

我是使用你说的方式,JS改变viewport的width,小于360固定为320;小于375固定360;小于440以下固定为375.原理是用JS重写viewport的meta.但屏幕旋转后多数浏览器不会重新渲染,需要JS来控制刷新,如果是表单页面会带来灾难性的后果.

[回复](#)



firstBlood说道:

2017年09月1日 11:51

其实,我们这边一直也是640这个尺寸在做,我也是按照640来写死的,现在微信端口有个比较明显的问题是 在iphone上 用户旋转手机,比如竖屏转到横屏 再转回竖屏,整个页面尺寸被放大了.放大的原理我估计是 viewport 转回来后 content里面的width是device-width而不是640,但是页面元素的尺寸还是640....

[回复](#)



24. SJY说道:

2016年08月12日 23:54

在天朝浏览器市场占有率第一的UC浏览器不支持vw, calc。

[回复](#)



saber说道:

2016年08月20日 17:49

手机uc就是移动端的IE6

[回复](#)



25. 刘东奇说道:

2016年08月11日 08:06

我有一点不明白,为什么字体大小要和屏幕大小成正比呢?只要人眼对屏幕的距离不变,那么人眼对屏幕的识别情况是不变的,对他最舒服的字体大小也应该是不变的,屏幕变大,增多显示内容就可以了,字体为什么要放大呢?5寸屏幕应该比3寸屏幕显示更多内容而不是更大内容,对吗

[回复](#)



前端半吊子说道:

2017年10月17日 19:19

我也有这个疑惑,增加更多内容不是用户体验更好吗

[回复](#)



海说道：  
2018年05月23日 10:24  
你可以不转换字体为rem  
[回复](#)



26. **George**说道：

2016年08月10日 20:13

这种自适应方式很不错，但css夹杂计算是否会影响效率呢

[回复](#)



27. **杰**说道：

2016年08月10日 14:41

张老师不能实现你说的,那个在不同屏幕下,字都是现实4行?这是为什么呢

[回复](#)



28. **peixin**说道：

2016年08月9日 23:07

一年多前在桌面端项目中用过vw、vh，一开始用着没问题，结果后来测试的时候发现少部分浏览器不支持，不过现在桌面端应该好多了，哈哈~

[回复](#)



29. **middle.Lin**说道：

2016年08月9日 20:40

你真是6 立贴为证 2年后见 必定又是一片绿地

[回复](#)



30. **轩辕Rowboat**说道：

2016年08月9日 19:16

真是特别能写！2012年就写了vw,vh，现在16年，预测1-2年后，vm火起来

[回复](#)



31. **杨**说道：

2016年08月9日 16:45

一两年后见

[回复](#)



32. **倾译**说道：

2016年08月9日 10:42

我想问一下，font-size: calc(100% + 2 \* (100vw - 375px) / 225);，除号后边的225，400，这样的数是怎么出来的呢？

[回复](#)



**A\_BetterMan**说道：  
2016年08月10日 00:07

字体范围\*（（当前屏幕大小-最小适应大小）/（最大适应-最小适应））

[回复](#)



33. 王鹏说道:  
2016年08月9日 09:23



张大大，你好！请问：

```
html {
/* iPhone6的375px尺寸作为16px基准，600px正好18px大小 */
font-size: calc(100% + 2 * (100vw - 375px) / 225);
}
@media screen and (min-width: 600px) {
html {
/* 600px-1000px每100像素宽字体增加1px(18px-22px) */
font-size: calc(112.5% + 4 * (100vw - 600px) / 400);
}
}
@media screen and (min-width: 1000px) {
html {
/* 1000px往后是每100像素0.5px增加 */
font-size: calc(137.5% + 5 * (100vw - 1000px) / 1000);
}
}
```

上面代码中的 2\*~~ / 225 4\*~~ / 400 5\*~~ / 1000 这些变化的数值是根据什么的呢？

[回复](#)

张鑫旭说道:  
2016年08月9日 10:04



依据自己设定的断点的font-size大小决定的，不同项目值会有所出入~

[回复](#)

新新前端说道:  
2016年10月9日 11:03



请问大神JS脚本判断浏览器版本或者是判断不支持某个CSS属性控制加载不同的CSS是怎么做到的？有相关的文章吗？CSS是不是本身就有个能力检测？

[回复](#)

## 最新文章

- » [常见的CSS图形绘制合集](#)
- » [粉丝群第1期CSS小测点评与答疑](#)
- » [分享三个纯CSS实现26个英文字母的案例](#)
- » [小tips: 纯CSS实现打字动画效果](#)
- » [CSS/CSS3 box-decoration-break属性简介](#)
- » [CSS :placeholder-shown伪类实现Material Design占位符交互效果](#)
- » [从天猫某活动视频不必要的3次请求说起](#)
- » [CSS vector-effect与SVG stroke描边缩放](#)
- » [CSS ::backdrop伪元素是干嘛用的？](#)
- » [周知：CSS -webkit-伪元素选择器不再导致整行无效](#)

## 今日热门

- » [常见的CSS图形绘制合集](#) (190)
- » [未来必热：SVG Sprite技术介绍](#) (119)
- » [粉丝群第1期CSS小测点评与答疑](#) (115)
- » [HTML5终极备忘大全（图片版+文字版）](#) (93)
- » [让所有浏览器支持HTML5 video视频标签](#) (86)
- » [Selectivizr-让IE6~8支持CSS3伪类和属性选择器](#) (82)
- » [CSS3下的147个颜色名称及对应颜色值](#) (79)
- » [视区相关单位vw, vh..简介以及可实际应用场景](#) (76)
- » [写给自己看的display: flex布局教程](#) (76)
- » [小tips: 纯CSS实现打字动画效果](#) (76)



#### 今年热议

- » 《CSS世界》女主角诚寻靠谱一起奋斗之人 <sup>(76)</sup>
- » 不借助Echarts等图形框架原生JS快速实现折线图效果 <sup>(64)</sup>
- » 看, for..in和for..of在那里吵架! <sup>(60)</sup>
- » 是时候好好安利下LuLu UI框架了! <sup>(47)</sup>
- » 原来浏览器原生支持JS Base64编码解码 <sup>(35)</sup>
- » 妙法攻略: 渐变虚框及边框滚动动画的纯CSS实现 <sup>(33)</sup>
- » 炫酷H5中序列图片视频化播放的高性能实现 <sup>(31)</sup>
- » CSS scroll-behavior和JS scrollIntoView让页面滚动平滑 <sup>(30)</sup>
- » windows系统下批量删除OS X系统.DS\_Store文件 <sup>(26)</sup>
- » 写给自己看的display: flex布局教程 <sup>(26)</sup>

#### 猜你喜欢

- 理解SVG viewport,viewBox,preserveAspectRatio缩放
- 基于screen.width的伪响应式开发
- 小tip:CSS vw让overflow:auto页面滚动条出现时不跳动
- 博闻强识: 了解CSS中的@ AT规则
- 5分钟快速了解下CSS4 color-adjust属性
- 视区相关单位vw, vh..简介以及可实际应用场景
- 等宽字体在web布局中应用以及CSS3 ch单位嘿嘿
- CSS百分比padding实现比例固定图片自适应布局
- jQuery之replace字符串替换实现不同尺寸图片切换
- 翻译 - CSS继承详解
- CSS前景背景自动配色技术简介