

第 20 章 CSS3 前缀和 rem

学习要点：

- 1.CSS3 前缀
- 2.长度单位 rem

主讲教师：李炎恢

本课程由 北风网 和 瓢城 Web 俱乐部 提供

本章主要探讨 HTML5 中 CSS 在发展中实行标准化的一些问题，重点探讨 CSS3 中新属性前缀问题和新的单位 rem。

一. CSS3 前缀

在 CSS3 的很多新属性推出时，这些属性还处在不太稳定的阶段，随时可能被剔除。而此时的浏览器厂商为了实现这些属性，采用前缀方法。各大厂商前缀列表如下：

浏览器	厂商前缀
Chrome、Safari	-webkit-
Opera	-o-
Firefox	-moz-
Internet Explorer	-ms-

我们之前学习过几个 CSS3 的新属性，比如：box-shadow、border-radius、opacity 等。这几个属性我们在前面的使用中，并没有添加所谓的浏览器厂商前缀。那是因为，这些属性已经在主流浏览器或版本成为了标准。具体进化步骤如下：

- 1.属性尚未提出，这个属性所有浏览器不可用；
- 2.属性被提出，但未列入标准，浏览器厂商通过私有前缀来支持该属性；
- 3.属性被列入标准，可以使用前缀或不使用前缀来实现属性特性；
- 4.属性不需要再使用前缀，所有浏览器都稳定支持。

我们就拿 border-radius 举例，它是 CSS3 的标准属性。早期的时候处于实验阶段，尚未列入标准时，需要使用厂商前缀。具体浏览器支持度如下：

属性	浏览器	带前缀版本	不带前缀版本	标准/实验
border-radius	IE	不支持	9.0+	标准
	Firefox	3.0 需带-moz-	4.0+	
	Safari	3.1 需带-webkit-	5.1+	
	Chrome	4.0	5.0+	
	Opera	不支持	10.5+	

如果是手机等移动端一般采用的是 IOS 或安卓系统，那么基本上它的引擎是 **webkit**，直接参考 **-webkit-** 即可。

在 CSS3 手册上，**Chrome** 支持 **border-radius** 的版本为 **13.0**，而部分教材和文章上写到只要 **5.0**。当然，这里可能表达的含义可能不同。而截至到 **2015 年 4 月份** 最新的 **Chrome** 版本已经到 **41.0** 了，所以，不管是 **5.0** 还是 **13.0** 都是老古董了，没必要深究。**Opera** 支持 **border-radius** 版本为 **11.5**，而目前的版本是 **28.0**，也无伤大雅了。

而被列入标准的 **box-shadow** 和 **opacity** 基本与 **border-radius** 前缀版本一致。

//因为目前处在标准阶段，没必要写前缀了

```
div {  
    border-radius: 50px;  
}
```

//实验阶段的写法

```
div {  
    -webkit-border-radius: 50px;  
    -moz-border-radius: 50px;  
    border-radius: 50px;  
}
```

实验阶段的写法有三句，分别解释一下：**-webkit-**表示 **Chrome** 浏览器的私有属性前缀、**-moz-**表示 **Firefox** 私有属性前缀，如果是处于实验阶段的旧版本浏览器，那么不支持 **border-radius**，从而通过厂商前缀的方式来支持。如果是新版浏览器，已经是处于标准阶段，那么又有两种说法：**1.**如果新版浏览器废弃了前缀，那么前缀写法就不支持了，仅支持标准写法；**2.**如果新版浏览器没有废弃前缀，那么两种写法都可以，但要注意顺序，且属性值要保持一致。

如果同时出现四个前缀+一个标准写法，四个前缀是当实验阶段时让四种引擎的浏览器厂商支持自己的私有属性，一个标准写法表示当这个属性列入标准后，使用新版浏览器运行时直接执行这个标准属性。

//前缀写法写在标准后面，且值不一样，就会出现問題

```
div {  
    border-radius: 50px;  
    -webkit-border-radius: 100px;  
}
```

特别注意：**1.**IE 的前缀 **-ms-**，和 **Opera** 的前缀 **-o-**，在 **border-radius** 中不存在；**2.**现在的 **Opera** 浏览器也支持 **-webkit-** 前缀，**-webkit-border-radius** 就能支持；**3.****Safari for Windows** 已被苹果公司在 **2012 年** 放弃，遗留版本为 **5.1.7**。

最后说明：**W3C** 官方的立场表示实验阶段的属性仅为了测试，未来有可能修改或删除。而大量的开发者也认为在实际项目中不应该使用前缀，而使用一种优雅降级保证一定的用户

体验，而通过渐进增减的方式让新版高级浏览器保证最高的效果。

二. 长度单位 rem

CSS3 引入了一些新的尺寸单位，这里重点推荐一个：**rem** 或者成为(根 **em**)。目前主流的现代浏览器都很稳定的支持。它和 **em**、百分比不同的是，它不是与父元素挂钩，而是相对于根元素<html>的文本大小来计算的，这样能更好的统一整体页面的风格。

```
//首先，来一段 HTML
<h1>标题<em>小标题</em></h1>
<p>我是一个段落，我是一段<code>代码</code></p>

//其次来一段 CSS
html {
    font-size: 62.5%;
}
h1 {
    font-size: 3em;
}
p {
    font-size: 1.4em;
}
```

这里做几个解释，我们在之前的 Web 设计中大量使用了 **px** 单位进行布局。因为，早期的固定布局使用 **px** 较为方便，逐渐养成了这种习惯。而使用 **em** 单位其实更加灵活，尤其是在修改样式时，只需要修改一下挂钩元素的那个大小即可，无须每个元素一个个修改。

但就算是 **em**，还是有一定问题。网页默认的字号大小为 **16px**，然后通过<html>设置 **62.5%**，将网页基准设置为 **10px**。而<h1>设置为 **3em**，就是自身大小的 3 倍；<p>设置为 **1.4em**，就是 **10px** 的 1.4 倍，即 **14px**。

现在问题来了，<code>里面的文本想设置 **11px**，怎么办呢？设置 **1.1em** 吗？不对，因为它挂钩的父元素不是<html>而是<p>变成了 **14px** 的 1.1 倍了，而想设置 **11px**，则需要设置 **0.786** 倍才行。但是，这样的计算量太大了。所以，W3C 推出了直接基于根元素单位：**rem**。

```
//直接基于<html>的单位
code {
    font-size: 1.1rem;
}
```

浏览器	rem 单位
Opera	11.6+
Firefox	3.6+
Safari	5.0+
Chrome	6.0+
IE	9.0+

感谢收看本次教程！

本课程是由北风网(ibeifeng.com)

瓢城 Web 俱乐部(ycku.com)联合提供：

本次主讲老师：李炎恢

谢谢大家，再见！