

3.CSS规范 - 代码格式

笔记本： 代码规范

创建时间： 2019/3/16 10:19

作者： 丁宇

更新时间： 2019/3/16 10:36

位置： 39°58'35 N 116°25'24 E

CSS规范 - 代码格式

选择器、属性和值都使用小写

在xhtml标准中规定了所有标签、属性和值都小写，CSS也是如此。

单行写完一个选择器定义

便于选择器的寻找和阅读，也便于插入新选择器和编辑，便于模块等的识别。去除多余空格，使代码紧凑减少换行。

如果有嵌套定义，可以采取内部单行的形式。

```
/* 单行定义一个选择器 */
.m-list li,.m-list h3{width:100px;padding:10px;border:1px solid #ddd;}
/* 这是一个有嵌套定义的选择器 */
@media all and (max-width:600px){
    .m-class1 .itm{height:17px;line-height:17px;font-size:12px;}
    .m-class2 .itm{width:100px;overflow:hidden;}
}
@-webkit-keyframes showitm{
    0%{height:0;opacity:0;}
    100%{height:100px;opacity:1;}
}
```

最后一个值也以分号结尾

通常在大括号结束前的值可以省略分号，但是这样做会对修改、添加和维护工作带来不必要的失误和麻烦。

省略值为0时的单位

为节省不必要的字节同时也使阅读方便，我们将0px、0em、0%等值缩写为0。

```
.m-box{margin:0 10px;background-position:50% 0;}
```

使用单引号

省略url引用中的引号，其他需要引号的地方使用单引号。

```
.m-box{background:url(bg.png);}
.m-box:after{content:'.';}
```

使用16进制表示颜色值

除非你需要透明度而使用rgba，否则都使用#f0f0f0这样的表示方法，并尽量缩写。

```
.m-box{color:#f00;background:rgba(0,0,0,0.5);}
```

根据属性的重要性按顺序书写

只遵循横向顺序即可，先显示定位布局类属性，后盒模型等自身属性，最后是文本类及修饰类属性。

显示属性	自身属性	文本属性和其他修饰
display	width	font
visibility	height	text-align
position	margin	text-decoration
float	padding	vertical-align
clear	border	white-space
list-style	overflow	color
top	min-width	background

```
.m-box{position:relative;width:600px;margin:0 auto 10px;text-align:center;color:#000;}
```

如果属性间存在关联性，则不要隔开写。

```
/* 这里的height和line-height有关联性 */  
.m-box{position:relative;height:20px;line-  
height:20px;padding:5px;color:#000;}
```

私有在前，标准在后

先写带有浏览器私有标志的，后写W3C标准的。

```
.m-box{-webkit-box-shadow:0 0 0 #000;-moz-box-shadow:0 0 0 #000;box-shadow:0  
0 0 #000;}
```

注释格式：/* 注释文字 */

对选择器的注释统一写在被注释对象的上一行，对属性及值的注释写于分号后。

注释内容两端需空格，已确保即使在编码错误的情况下也可以正确解析样式。

在必要的情况下，可以使用块状注释，块状注释保持统一的缩进对齐。

原则上每个系列的样式都需要有一个注释，言简意赅的表明名称、用途、注意事项等。

```
/* 块状注释文字  
 * 块状注释文字  
 * 块状注释文字  
 */  
.m-list{width:500px;}  
.m-list li{height:20px;line-height:20px;/* 这里是对line-height的一个注释  
*/overflow:hidden;}  
.m-list li a{color:#333;}  
/* 单行注释文字 */  
.m-list li em{color:#666;}
```

原则上不允许使用Hack

很多不兼容问题可以通过改变方法和思路来解决，并非一定需要Hack，根据经验你完全可以绕过某些兼容问题。

一种合理的结构和合理的样式，是极少会碰到兼容问题的。

由于浏览器自身缺陷，我们无法避开的时候，可以允许使用适当的Hack。

统一Hack方法

统一使用 “*” 和 “_” 分别对IE7和6进行Hack。如下代码所示：

```
/* IE7会显示灰色#888，IE6会显示白色#fff，其他浏览器显示黑色#000 */  
.m-list{color:#000;*color:#888;_color:#fff;}
```

建议并适当缩写值

“建议并适当” 是因为缩写总是会包含一系列的值，而有时候我们并不希望设置某一值，反而造成了麻烦，那么这时候你可以不缩写，而是分开写。

当然，在一切可以缩写的情况下，请务必缩写，它最大的好处就是节省了字节，便于维护，并使阅读更加一目了然。

缩写方法请查阅css手册。

选择器顺序

请综合考虑以下顺序依据：

- 从大到小（以选择器的范围为准）
- 从低到高（以等级上的高低为准）
- 从先到后（以结构上的先后为准）
- 从父到子（以结构上的嵌套为准）

以下仅为简单示范：

```
/* 从大到小 */  
.m-list p{margin:0;padding:0;}  
.m-list p.part{margin:1px;padding:1px;}  
/* 从低到高 */  
.m-logo a{color:#f00;}  
.m-logo a:hover{color:#fff;}  
/* 从先到后 */  
.g-hd{height:60px;}  
.g-bd{height:60px;}  
.g-ft{height:60px;}  
/* 从父到子 */  
.m-list{width:300px;}  
.m-list .itm{float:left;}
```

选择器等级

a = 行内样式style。

b = ID选择器的数量。

c = 类、伪类和属性选择器的数量。

d = 类型选择器和伪元素选择器的数量。

选择器	等级(a,b,c,d)
style=" "	1,0,0,0
#wrapper #content {}	0,2,0,0
#content .dateposted {}	0,1,1,0
div#content {}	0,1,0,1
#content p {}	0,1,0,1
#content {}	0,1,0,0
p.comment .dateposted {}	0,0,2,1
div.comment p {}	0,0,1,2
.comment p {}	0,0,1,1
p.comment {}	0,0,1,1
.comment {}	0,0,1,0
div p {}	0,0,0,2
p {}	0,0,0,1