CSS的高效的写法推荐

2011-04-10

当浏览器解析html的时候，它构造了一个文档树来展现所有被显示的元素。 它在特定的样式表中去匹配元素，根据标准的css的层叠，继承和顺序规则， 在mozilla的实现中（可能其他的也是这样），对于每一个元素，css引擎通过规则去寻找匹配，css引擎评估每一个规则是从右到左的，从最右的selector开始， 也称之为key selector，直到找到匹配为止。

根据这个规则，越少的规则使用，css引擎将评估的越好。因此，移除没有用的css是改善页面性能的重要一步。之后，对于页面包含很多的元素的css规则， 优化这些规则也有利于提高页面的性能。优化页面性能最重要的一点在于使用明确的，避免不需要冗余的，使引擎系统快速的匹配到元素而不需要花费太多的时间。

**下面的这些规则被认为是低效的**

1. 使用了后代选择器

使用全局选择器作为key

body \* {...}

.hide-scrollbars \* {...}

使用了标签选择器作为key

ul li a {...}

#footer h3 {...}

1. 使用子选择器和相邻选择器

使用全局选择器作为key

body > \* {...}

.hide-scrollbars > \* {...}

使用标签选择器作为key

ul > li > a {...}

#footer > h3 {...}

子选择器和相邻选择器是低效的是因为对于每一个元素的匹配，浏览器不得不评估其他的一些元素，它需要双倍的时间耗费在匹配上。同时，越少的精确的 key，越多的时间花费在匹配元素上面。尽管如此，即使是不高效的，他们比后代选择器的使用的效果要好一点。

1. 过渡的限定选择器

id选择器是唯一的，包含了tag或者class的限定仅仅是增加了一些无用的信息（浏览器去评估）

ul#top\_blue\_nav {...}

form#UserLogin {...}

1. 使用：hover在一些没有：hover的元素上面

在非标准模式下面。ie7，ie8将忽略这些规则。在标准模式下面，可能会导致一些性能的退化

h3:hover {...}

.foo:hover {...}

**推荐的方式**

1. 避免使用全局样式

允许一个元素去继承它的祖先，或者使用一个class去应用复杂的元素。

1. 将规则写的越精确越好

偏向使用class ，id，少使用tag

1. 移除一些无用的限定

下面的这些限定是多余的：1.id选择器被class 或者tag选择器限定；2.class被tag选择器进行限定（如果一个class只被用于一个tag，这也是很好的实践）。

1. 避免使用后代选择器，特别是包含了一些无用的祖先元素

eg：body ul li a{...} 制定了一个无用的body限定，因为所有的元素都是在body中。

1. 使用class选择器取代后代选择器

eg：如果你需要两个不同的样式（一个无序列表，一个有序列表），不要使用下面的样式

ul li {color:blue}

ol li {color:red}

应该这样的使用

.unordered-list-item {color: blue;}

.ordered-list-item {color: red;}

如果你一定要用后代选择器，建议你使用子选择器

1. 避免使用:hover在一些没有连接的元素上面（对于ie序列的）

如果你使用：hover在一些无连接的元素上面，ie7,8会认为他们是无用的。建议使用事件onmouseover 来模拟。