**引子：**

　　女孩子都喜欢买衣服，而我也不例外，奈何钱包太瘦，买不起高大上的定制，只能买撞衫率极高的休闲衣，不过对于我来说，我还是开心的，毕竟买衣服买的不仅是衣服，更是一种心情。在web前端的世界里，css里也充满了生活的艺术，都说，艺术源于生活，却高于生活，是对生活的一种升华，那么在css里，是个性化的定制好，还是模块化的量产好呢？这就要看各个项目了，这是一种权衡，一切离不开一个度，这世上没有最好的，只有最适合自己的，在web前端的世界里亦是如此。

　　假如你现在只是做自己的一个小博客，那么就需要你有自己的个性，因为博客相当于你个人的代言，那么很多都是个性化的，而且需要你独自一人慢慢打造出适合自己的风格。不过，大多数的时候是在工作中，对于工作中，就要考虑两个字--合作，面对一个需要大家合作才能尽快完成的项目，这就需要模块化的思想了，大家需要一个共同的标准，有一个共享库，不在是个性化的代码，而是便于团队成员阅读和维护的优雅代码。由于css模块化涉及方方面面，所以，我就一点点揭开它的神秘面纱吧！这一篇就先聊聊命名好了。

**正篇：**

**什么是css模块化思想？(what)**

为了理解css模块化思想，我们首先了解下，什么是模块化，在百度百科上的解释是，在系统的结构中，模块是可组合、分解和更换的单元。模块化是一种处理复杂系统分解成为更好的可管理模块的方式。它可以通过在不同组件设定不同的功能，把一个问题分解成多个小的独立、互相作用的组件，来处理复杂、大型的软件。看完模块化，是不是有种拼图的即视感，可以把大图分成各个小图，然后把小图拼成大图，分与合的艺术感。那么css模块化思想，也就是在css编写环境中，用上模块化的思想，把一个大的项目，分解成独立的组件，不同的组件负责不同的功能，最后把模块组装，就成了我们要完成的项目了。

**为什么需要css模块化?(why)**

当做一个大项目，几个人团队合作开发，结果看不懂彼此的代码，怎么办，当面对前人已经写好代码，需要修改，可是无处下手，怎么办.当代码耦合，修改费时费力，怎么办，当需要迭代，面对庞大的代码，牵一发动全身的悲催时刻，怎么办，这个时候，模块化思想就是救星了。css写法特别的灵活，也因为灵活，所以容易耦合在一起，这时候就需要进行模块化的分离。那么css模块化的好处多多，列举了一些如下：

　　　　提高代码重用率

　　　　提高开发效率、减少沟通成本

　　　　提高页面容错

　　　　降低耦合

　　　　降低发布风险

　　　　减少Bug定位时间和Fix成本

　　　　更好的实现快速迭代

　　　　便于代码维护

　　　　。。。。。。

**怎样实现css模块化？（how）**

　　　　要实现css模块化需要很多步，今天我们就说说第一步，命名。

　　　　看别人的代码，就好像进入对方的军事基地一样，要想里面的武器用的得心应手，首先得知道每个武器的名字，假如你的武器都是取的不三不四，那么就算来到了基地，估计也不敢动你的武器了，更别提战斗了，所以，要规范我们在css中的命名，尽量做到团队成员能一看就懂。

　　　　css命名上的优化可以分为css文件名的命名，以及css选择器的命名。

**css样式文件名：**

1）

主要的 master.css   
布局，版面 layout.css   
专栏 columns.css   
文字 font.css   
打印样式 print.css   
主题 themes.css

2）

reset.css // 对浏览器的默认样式进行重设   
layout.css // 管理页面的布局   
typeset.css // 图文的编排  
color.css // 统一管理颜色的搭配   
print.css // 打印效果样式   
ie.css // 把对ie的hack单独分开

3）

reset.css   
header.css // 头部的所有样式   
container.css // 除头部/底部外的中间区域样式   
footer.css // 底部样式   
print.css   
ie.css

4)

reset.css /\*页面样式充值\*/  
header.css /\*全站头部样式\*/  
footer.css /\*全站尾部样式\*/  
public.css /\*全站公共模块样式\*/  
index.css /\*首页特有样式\*/  
container.css /\*二级及以下主体样式\*/  
print.css /\*打印样式\*/  
ie.css /\*IE hack\*/

等等

还是那句话，没有最好的命名方式，只有最合适的命名方式。主要就是从html结构和bug的快速定位和便于其他成员的修改上去考虑。

假如你的网站是新闻类的门户网站，然后html的结构就主要是布局，版面 layout.css ，专栏 columns.css ，文字 font.css ，打印样式 print.css 就比较适合，因为跟html很一致，也很契合。

假如是企业官网的话，一般html结构都是头和尾都有标准的，所以与html结构最契合的就是第三种了，同时我们可以优化第三种，在container里面再分类，分成common,layout，就方便进一步的修改，至于common里就可以放入font,color一些公用的样式。

没有完美的生活，也没有完美的人，所以也没有完美的文件命名，只有尽量做到与html布局契合，然后重在从每次的项目中反省，总结，条条大路通罗马，能抓住老鼠的就是好猫。

**css选择器的命名：**

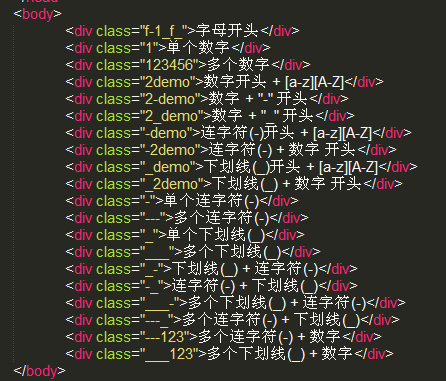
规范css选择器的命名，是个老生常谈的问题，对它的规范，不仅是模块化实现的一环，便于团队开发，同时也规避了一些兼容性的麻烦。因为各个浏览器，IE，火狐，谷歌，safari

，每个浏览器,IE产品，火狐，safari,coogle Chrome，都会因为命名不规范会产生不同样式..

所有的CSS语法是不区分大小写的ASCII范围（即，i.e., [a-z] and [A-Z]是等价的）中，除了部分不属于CSS的控制之下。例如，在HTML中值的区分大小写属性“ID”的字体名称和“class”，和URI就在于本规范的范围之内。请特别注意，元素名称是不区分大小写的HTML，但区分大小写的XML。  
在CSS中，标识符（包括元素名称，类别，并在选择的ID）只能包含字符[A-ZA-Z0-9]和ISO 10646字符U + 00A0高，加上连字符（ - ）和下划线（ \_）;他们不能以数字开头，两个连字符，或一个连字符后跟数字。标识符也可以包含转义字符和任何ISO 10646字符作为一个数字代码（见下项）。例如，识别符“B&W?”可被写为“B \＆W \？”或“B \ 26W¯¯\ 3F”。请参考W3C CSS2.1的4.1.3节,链接地址：http://www.w3.org/TR/CSS21/syndata.html#characters

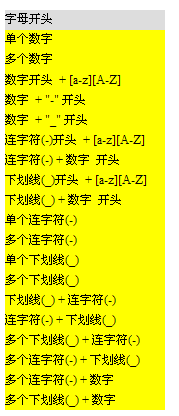
通过如下代码：



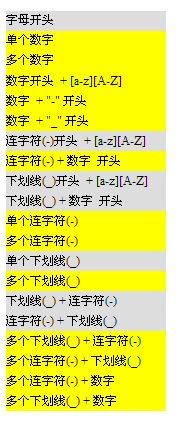


在各浏览器的结果如下：

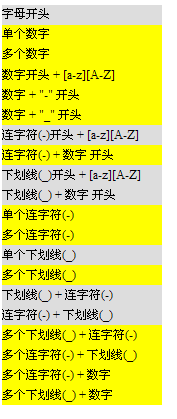
ie6:



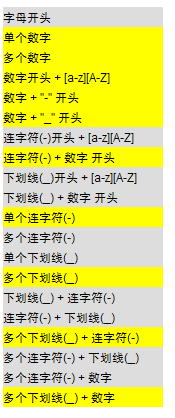
ie7



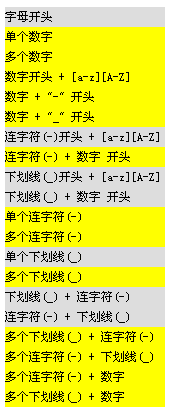
ie8



Firefox38.0.1



360结果，Google Chrome，safari,opera结果一致如下：



从上面直观的了解到选择器的命名在各浏览器下的支持情况有所不同。因此，如果选择器的命名不规范，将影响各浏览器下的渲染效果。为了兼容以上的浏览器，省麻烦，我们尽量命名以字母开头。可以保证都兼容。

**常用的css命名规则**

头：header

内容：content/container

尾：footer

导航：nav

侧栏：sidebar

栏目：column

页面外围控制整体布局宽度：wrapper

左右中：left right center

登录条：loginbar

标志：logo

广告：banner

页面主体：main

热点：hot

新闻：news

下载：download

子导航：subnav

菜单：menu

子菜单：submenu

搜索：search

友情链接：friendlink

页脚：footer

版权：copyright

滚动：scroll

内容：content

标签页：tab

文章列表：list

提示信息：msg

小技巧：tips

栏目标题：title

加入：joinus

指南：guide

服务：service

注册：register

状态：status

投票：vote

合作伙伴：partner

(二)注释的写法:

/\* Header \*/

内容区

/\* End Header \*/

(三)id的命名:

(1)页面结构

容器: container

页头：header

内容：content/container

页面主体：main

页尾：footer

导航：nav

侧栏：sidebar

栏目：column

页面外围控制整体布局宽度：wrapper

左右中：left right center

(2)导航

导航：nav

主导航：mainbav

子导航：subnav

顶导航：topnav

边导航：sidebar

左导航：leftsidebar

右导航：rightsidebar

菜单：menu

子菜单：submenu

标题: title

摘要: summary

(3)功能

标志：logo

广告：banner

登陆：login

登录条：loginbar

注册：register

搜索：search

功能区：shop（如购物车）

标题：title

加入：joinus

状态：status

按钮：btn

滚动：scroll

标签页：tab

文章列表：list

提示信息：msg

当前的: current

小技巧：tips

图标: icon

注释：note

指南：guide

服务：service

热点：hot

新闻：news

下载：download

投票：vote

合作伙伴：partner

友情链接：link

版权：copyright

(四)class的命名:

(1)颜色:使用颜色的名称或者16进制代码,如

.red { color: red; }

.f60 { color: #f60; }

.ff8600 { color: #ff8600; }

（2）字体大小,直接使用”font+字体大小”作为名称,如

.font10px { font-size: 10px; }

.font6pt {font-size: 6pt; }

(3)对齐样式,使用对齐目标的英文名称,如

.left { float:left; }

.bottom { float:bottom; }

(4)标题栏样式,使用”类别+功能”的方式命名,如

.barnews { }

.barproduct { }

注意事项::

1. 一律小写;
2. 尽量用英文;
3. 不加中杠和下划线;
4. 尽量不缩写；

我比较喜欢bootstrap的简洁优雅的风格。所以css命名也可以参考一下。

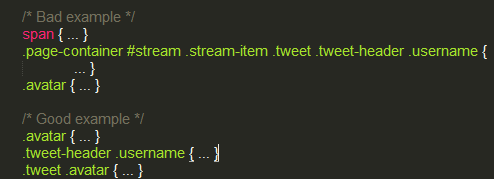
**特别是class命名：**

* class 名称中只能出现小写字符和连字符（不是下划线，也不是驼峰命名法）。连字符应当用于相关 class 的命名（类似于命名空间）（例如，.btn 和 .btn-danger）。
* 避免过度任意的简写。.btn 代表 *button*，但是 .s 不能表达任何意思。
* class 名称应当尽可能短，并且意义明确。
* 使用有意义的名称。使用有组织的或目的明确的名称，不要使用表现形式（presentational）的名称。
* 基于最近的父 class 或基本（base） class 作为新 class 的前缀。
* 使用 .js-\* class 来标识行为（与样式相对），并且不要将这些 class 包含到 CSS 文件中。



**选择器**

* 对于通用元素使用 class ，这样利于渲染性能的优化。
* 对于经常出现的组件，避免使用属性选择器（例如，[class^="..."]）。浏览器的性能会受到这些因素的影响。
* 选择器要尽可能短，并且尽量限制组成选择器的元素个数，建议不要超过 3 。
* **只有**在必要的时候才将 class 限制在最近的父元素内（也就是后代选择器）（例如，不使用带前缀的 class 时 -- 前缀类似于命名空间）。



css编码规范可以参考http://codeguide.bootcss.com/

非常高兴能跟大家一起分享，一起学习，今天就写到这里了，因为本人水平有限，有错误的地方，欢迎指正，谢谢啦！---妙瞳。