**一：css分类方法**

　　通常一个页面我们只引用一个css，但是对于较大的项目，我们需要把css文件进行分类。

　　按照css的性质和用途，我们可以将**css文件分成“公共型样式”、“特殊型样式”、“皮肤型样式”**，并以此为顺序引用。那么他们分别是什么呢？

* **公共型样式**是最为重要的部分，对于比较小的项目，我们只引入一个css，这个css的样式即公共型样式，一般包括：“标签的重置和设置默认值”（以消除不同浏览器之间的差异）、“统一调用背景图和清除浮动或其他需统一处理的长样式（这样就无需对每个进行分别的处理）”、“网站通用布局”、“通用模块和其扩展”、“元件和其扩展”、“功能类样式”、“皮肤类样式”。后面会具体介绍。
* **特殊型样式**即对某个维护率较高的栏目或者某个与网站整体差异较大的页面独立引入这样一个特殊型样式，方便我们维护。
* **皮肤型样式**即产品需要换肤功能，那么我们就需要将颜色、背景等抽离出来放在这里，方便管理。

　　css文件我们分为了公共型样式、特殊型样式、皮肤型样式，那么在css文件的内部我们又是怎么分类的呢？（**此部分为重点**）

* **重置和默认的css代码**。这是为了消除默认样式和浏览器的差异，并设置部分标签的初始样式，**以减少后面的重复劳动。**你可以根据自己的网站需求设置，也可以使用别人写好的一些初始化代码，比如：[雅虎工程师提供的css初始化代码](http://www.cnblogs.com/zhuzhenwei918/p/6053955.html)。这一部分代码放在css内部的最上面。
* **统一处理的css代码。**这里可以统一调用背景图和清除浮动（指通用性较高的布局、模块、原件内的清楚），实际上，雅虎工程师提供的css初始化代码中就有清除浮动的css代码。这一部分代码放在重置和默认的css代码下面。
* 布局（grid）: 我们将页面分割为几个大块，通常有头部、主体、主栏、侧栏、尾部等。常用！
* 模块（module）：即语义化的可以重复使用的较大的整体。如导航、登陆、注册、列表、评论、搜索等。常用！
* 元件（unit）：通常是一个**不可再分**的较为小巧的个体，被重复用于各种模块中，比如按钮、输入框、loading、图表等。常用！
* 功能（function）：为方便一些常用样式的使用，我们将这些使用率较高的样式剥离出来，按需使用，通常这些选择器具有固定样式表现，比如清除浮动。不常用，不可滥用！
* 皮肤（skin）：对于换肤型网站需要使用，将皮肤型的样式抽离出来，非换肤型网站不可滥用，不常用。
* 状态:即一些状态类样式。不常用。

### 二、css命名规则

**重要：使用类选择器，放弃ID选择器。因为ID在一个页面中的唯一性导致了如果以ID为选择器来写css，就无法重用，而class而优势在于复用性，而且私有度也并不高。因此，我一般情况下会选择在HTML中的ID用于JavaScript，但是在CSS里只用class，一个ID也不用。这样做实际上也是将表现和行为分开，而不是混在一起。**

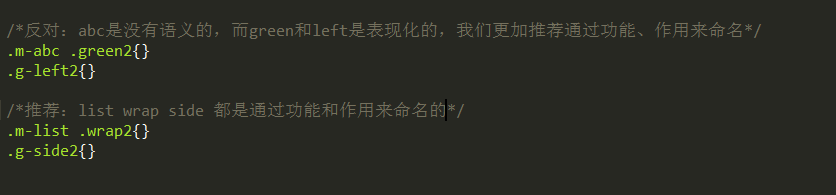
**在html文件中写class的方法：使用单个字母+“-”为前缀。**比如：对于header部分，我们可以这样写:

|  |  |
| --- | --- |
| 1 | <div class="g-header"></div> |

　　那么“单个字母”是什么呢? 实际上就是刚刚介绍的css内部分类方法中的布局（grid）的第一个字母，**好处是我们在css代码中可以通过首字母清楚地知道其作用是什么，即见名知意**。由此可知，对于模块化（module）的部分我们可以这样写：<div class="m-log"></div>。此div即为登陆框这个模块。 对于元件部分，即不可再分的个体，我们可以这样写<img class="u-img" src="../images/iam.png" alt="pic">。对于功能（function）部分，我们就可以这么写：<div class="f-clearFloat"></div>。对于皮肤（skin）部分，我们可以这么写：<div class="s-changeSkin">。 其中 g、m、u这三个首字母是我们用的最多的首字母，其他的首字母应该根据情况来使用。**注意1：在css中，样式的选择器总是要以上面的 .g-  .m- .u- .f- .s-这五类开头，然后再里面使用后代选择器。 注意2：这并不是说所有的className都需要加这些前缀，对于一些不属于这几种的元素，我们完全可以不加前缀，作为后代选择器使用。**

 　　那么后代选择器要怎么使用呢? 我们约定不以单个字母+“-”为前缀且长度大于等于2的类选择器为后代选择器。如：.g-date .u-rightArrow{  float: right;} 这个就是不合适的，我们直接写成 .u-rightArrow{  float: right;}即可。 且**一个语义化的标签也可以是后代选择器。比如.m-list li{}。** 上一句话也就是说不是语义化的标签作为后代选择器是不合适的，如：.m-list div{},这样的写法很有可能造成污染。

　　除此之外，我们应当注意：在命名时应当尽量简约而不失语义。如下所示：



 　　对于相同语义的不同类命名，我们可以直接加数字或字母区分即可（如.m-list1、.m-list2,都是列表模块，都通加数字即表示不同的列表模块）。

### 三：css代码格式

　　1.选择器、属性和值都是用小写。

        这一点非常关键：根据[xhtml规范](http://baike.baidu.com/view/15906.htm)，所有的标签属性和值都要使用小写形式，而我们知道xhtml更为标准，所以最好遵循之，这样，选择器必须小写就是十分必要的了。既然这样我们就不能使用驼峰式写法来写类名了，如class="u-leftArrow"实际上就是不规范的了，最好写成class="u-left\_arrow"，也可以表达相同的意思。

　　2.单行写完一个选择器定义。

　　　优点：**便于选择器的寻找和阅读，**也便于插入新的选择器和编辑，便于模块等的识别。**更重要的是可以去除多余空格，使代码紧凑减少换行。**试想，如果每一行只有一个属性名和属性值，那么对于一个大项目而言，就很难做到选择器的寻找和阅读了，而使用一行写完一个选择器，那么就有可能缩短为1/6到1/10，这还是非常客观的。

　　3.**最后一个值也要一分号结尾。**

实际上，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。比如，在最后添加一个属性，如果之前没有在末尾添加分号，那么你就要在新添加的属性前添加分号，否则就会出错，比如在我的[一篇博文](http://www.cnblogs.com/zhuzhenwei918/p/6098687.html#方框问题)为解决中文字体显示为方框添加JSON数据时就出现过此类问题。

　　 4.省略值为0的单位

　　　　优点：可以节省不必要的字节同时也为了方便阅读，我们将0px、0em、0%等都缩写为0.如下所示：

|  |  |
| --- | --- |
| 1 | .m-box{margin:0 10px; backgrond-position:50% 0;} |

　　  5.使用16进制表示颜色值，其中的字母使用大写形式，并尽量缩写。

　　　　除非在你需要透明度而使用rgba，否则都是用#FFFFFF这样的写法，并尽量缩写，如#FFF。使用大写形式，是因为这样更加具有易读性，且有利于压缩，而缩写为了减少不必要的字节。

　　　　(补充说明：博友@batsing 指出在PC端使用16进制表示颜色，IE8及以下不兼容，在此表示感谢，希望大家也可以注意这个问题。)

　　　6.根据属性的重要性顺序书写。

           在上面的第二点我们说到应当在单行写完一个选择器。如果我们不遵循一定的书写顺序，那么写出来的代码一定是混乱的，那么怎么才能写出优雅的css代码呢？这时就需要根据属性的重要性顺序书写。如下图所示：



　　注意：只遵循横向顺序即可，即**先书写定位布局类属性，在书写盒模型等自身属性，最后书写文本类及修饰类属性。**另外，如果属性间存在关联性，则不要隔开写，如下所示。

|  |  |
| --- | --- |
| 1 | .m-box{position:relative;height:20px;line-height:20px; padding:5px;color:#000;} |

　　其中的height和line-height具有关联性，我们尽量不要分开写。

　　　7.私有在前，标准在后。

　     先写带有浏览器私有标志的属性，后写W3C标准的属性。因为私有的属性，说明浏览器自身还没有规范化，标准属性是用不了的。若某一天该浏览器的属性规范化了，那么说明标准属性可以使用了，而如果我们先写W3C标准属性，最后写私有属性，就有可能导致私有属性覆盖标准属性。因此私有在前，标准在后的写法是考虑到了以后可能会出现的问题。

　　如：

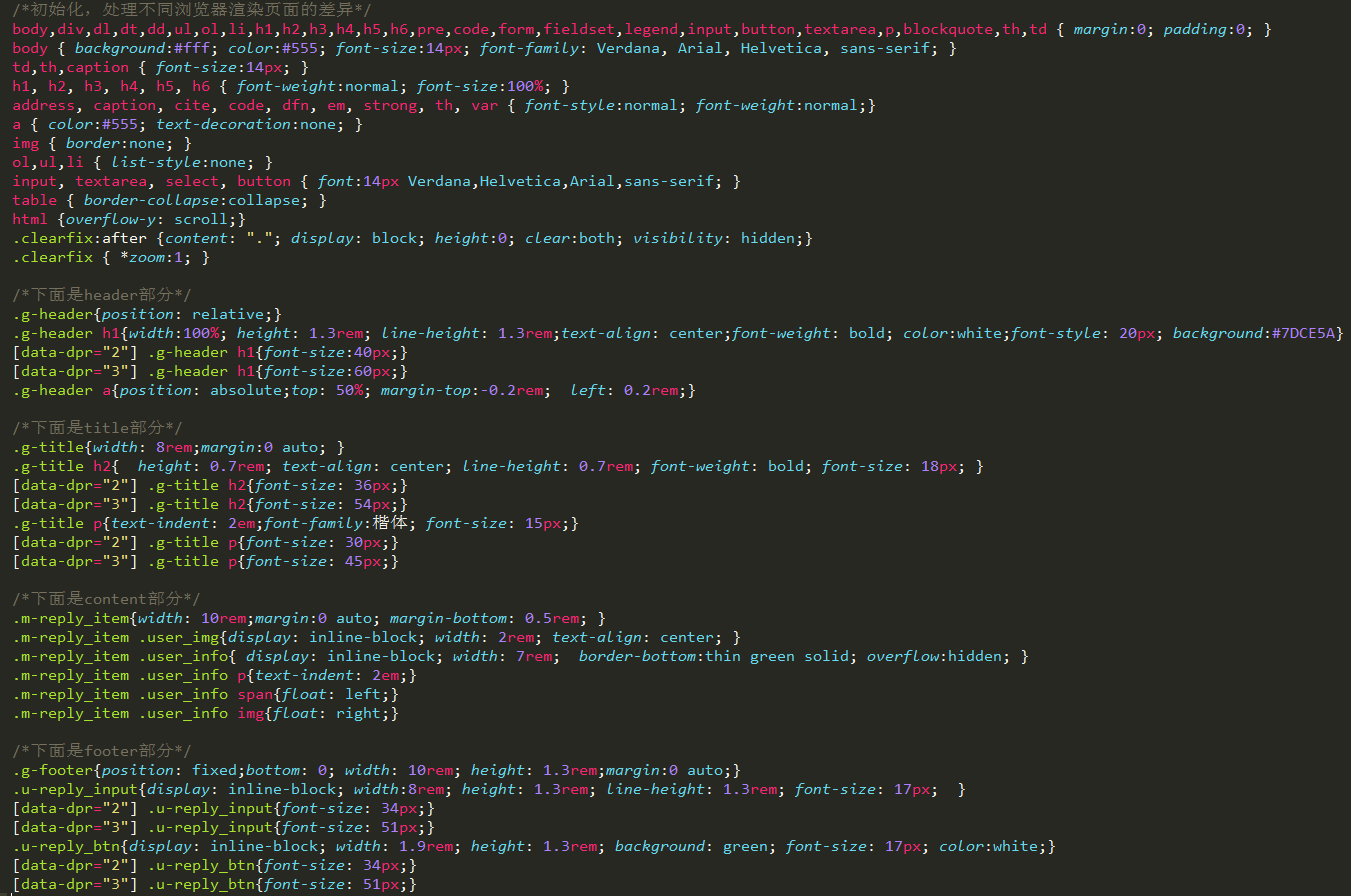
|  |  |
| --- | --- |
| 1 | .m-box{-webkit-box-shadow:0 0 0 #000;-moz-box-shadow:0 0 0 #000;box-shadow:0 0 0 #000;} |

      8.选择器等级

　　　!important>行内样式style>id选择器>类、伪类和属性选择器>标签选择器和伪元素选择器

**9.css内部的顺序**

**我认为，对于一个网页而言，我们在写css时，可以分为几个部分来写，比如header部分的css代码，main部分的css代码，部分之间通过空格隔开并给以响应的注释，这样更有利于我们的阅读和后期的维护，如下所示。**

****

　　10.优化方案：对于可以缩写的值我们尽量采用缩写形式，这样有利于减小css文件大小，并增加可读性和可维护性。且最好尽量减少没有实际作用的冗余的属性。有时我们会将定义相同的或者有大部分属性值相同的一系列的选择器组合到一起（采用逗号的方法）来统一定义，这样可以为你节省很多字节和宝贵时间。

### 四.类选择器的命名建议

　　在前面说到，命名className时，应当以其作用、功能来命名，而不要使用没有语义化或者以颜色、左右空间位置的文字来命名。

　1.　对于布局，即用.g-作为前缀，通常有以下推荐的写法。

　　头部： header或head

　　主体： body

　　尾部：footer或foot

　　主栏： main

　　侧栏：side

　　盒容器： wrap或box

　　主栏子容器：mainc

 　　侧栏子容器：sidec

　2.对于模块，即.m-作为前缀。元件，.u-作为前缀，通常有下面推荐的写法。

　　导航： nav

　　子导航：subnav

　　菜单：menu

　　选项卡：tab

　　标题区：head或title

　　内容区：body或content

　　列表：list

　　表格：table

　　表单：form

　　排行：top

　　热点：hot

　　登录：login

　　标志：logo

　　广告：adervertise

　　搜索：search

　　幻灯：slide

　　帮助：help

　　新闻：news

　　下载：download

　　注册：register或regist

　　投票：vote

　　版权:copyright

　　 结果：result

　　按钮：button

　　输入：input

　3.对于功能，即以.f-为前缀，通常推荐如下：

　　清除浮动：clearboth

　　向左浮动：floatleft

　　向右浮动: floatright

　　溢出隐藏：overflowhidden

　4.对于颜色，即以.s-为前缀，通常推荐如下：

　　字体颜色：fontcolor

　　背景：background

　　背景颜色：backgroundcolor

　　背景图片：backgroundimage

　　背景定位：backgroundposition

　　边框颜色：bordercolor

　5.对于状态，即以.z-为前缀，通常推荐如下：

　　选中:selected

　　当前：current

　　显示：show

　　隐藏：hide

　　打开：open

　　关闭:close

　　出错：error

　　不可用:disabled

### 五、使用css选择器常出现的错误。

* .class{}  不可用一个没有类别的样式作为主选择器，它只能作为后代选择器
* .m-xxx div{} 不可用没有语义的便签作为选择器,会造成大面积污染
* .g-zzz  .m-xxx{}  一般类别的选择器的后代选择中不应当包括类别选择器
* .m-xxx .main .mainc .nav li{} 不要将选择器写的过于冗长，因为选择器的结构越复杂，浏览器的消耗就越大，建议在3个长度之内写完。

补充内容：

<http://www.cnblogs.com/hustskyking/p/css-spec.html>