# Html CSS代码规范

# 申明

本文档属于上海数凹科技有限公司，只限于内部使用。如果违反，公司将保留追究泄密的权利。

|  |  |  |
| --- | --- | --- |
| 日期 | 作者 | 备注 |
| 2016-11-03 | Andy.zhou | 创建文档 |
|  |  |  |

目录

[Html CSS代码规范 1](#_Toc465936541)

[申明 2](#_Toc465936542)

[1 文档概述 6](#_Toc465936543)

[1.1 文档目标 6](#_Toc465936544)

[1.2 编程风格和原则 6](#_Toc465936545)

[1.2.1 编程风格 6](#_Toc465936546)

[1.2.2 规范原则 6](#_Toc465936547)

[1.3 规范分类 7](#_Toc465936548)

[2 html 代码规范 8](#_Toc465936549)

[2.1 通用规则 8](#_Toc465936550)

[2.1.1 协议头 8](#_Toc465936551)

[2.2 通用格式规则 8](#_Toc465936552)

[2.2.1缩进 8](#_Toc465936553)

[2.2.2 大小写：只使用小写 9](#_Toc465936554)

[2.2.3 尾随空格 9](#_Toc465936555)

[2.2.4 注释 9](#_Toc465936556)

[2.2.5 TODO待定项 9](#_Toc465936557)

[2.3 meta 原型格式 10](#_Toc465936558)

[2.3.1文档类型：使用HTML5 10](#_Toc465936559)

[2.3.2 字符编码 10](#_Toc465936560)

[2.3.3 IE兼容模式 10](#_Toc465936561)

[2.4 html风格规则 10](#_Toc465936562)

[2.4.1 HTML 合法性验证 10](#_Toc465936563)

[2.4.2 语义性 11](#_Toc465936564)

[2.4.3 多媒体后备 11](#_Toc465936565)

[2.4.4 关注点分离 12](#_Toc465936566)

[2.4.5 可选Tags： 13](#_Toc465936567)

[2.4.6 实体引用 13](#_Toc465936568)

[2.4.7 Type属性：省略 13](#_Toc465936569)

[2.5 格式规则 14](#_Toc465936570)

[2.5.1 通用格式 14](#_Toc465936571)

[2.5.2 引号 15](#_Toc465936572)

[2.6 元素建议 15](#_Toc465936573)

[2.6.1 html5 元素 15](#_Toc465936574)

[2.6.2 结构性元素 16](#_Toc465936575)

[2.6.3 头部元素 17](#_Toc465936576)

[2.6.4 文本元素 17](#_Toc465936577)

[2.6.5 媒体元素 17](#_Toc465936578)

[2.6.6 列表元素 17](#_Toc465936579)

[2.6.7 表单元素 18](#_Toc465936580)

[2.7 属性建议 18](#_Toc465936581)

[2.7.1 顺序建议 18](#_Toc465936582)

[2.7.2 文件和目录命名约定 19](#_Toc465936583)

[2.7.3 空格/标签/注释 19](#_Toc465936584)

[2.7.4 id和class命名约定 19](#_Toc465936585)

[2.7.5 选择器 20](#_Toc465936586)

[2.7.6 z-index 20](#_Toc465936587)

[3 css代码规范 21](#_Toc465936588)

[参考资料 22](#_Toc465936589)

# 1 文档概述

## 1.1 文档目标

本文档是描述html和CSS的代码规范，规范分为两类，必须follow和建议follow。

## 1.2 编程风格和原则

### 1.2.1 编程风格

**所谓"编程风格"（programming style），指的是编写代码的样式规则。**不同的程序员，往往有不同的编程风格。

有人说，编译器的规范叫做"语法规则"（grammar），这是程序员必须遵守的；而编译器忽略的部分，就叫"编程风格"（programming style），这是程序员可以自由选择的。这种说法不完全正确，**程序员固然可以自由选择编程风格，但是好的编程风格有助于写出质量更高、错误更少、更易于维护的程序。**

所以，有一点应该明确，**"编程风格"的选择不应该基于个人爱好、熟悉程度、打字工作量等因素，而要考虑如何尽量使代码清晰易读、减少出错。你选择的，不是你喜欢的风格，而是一种能够清晰表达你的意图的风格。**这一点，对于html/css这种语法自由度很高、设计[不完全成熟](http://www.ruanyifeng.com/blog/2010/01/12_javascript_syntax_structures_you_should_not_use.html)的语言尤其重要。

本文的风格主要参考了google的规范和github开源的资料。

### 1.2.2 规范原则

规范的原则是为了团队更高效的开发和维护高质量的代码。

原则一：质量为先，容易维护；

原则二：逻辑结构清晰、优雅、简洁，易于阅读和维护；

原则三：风格一致，团队成员可以开心的码代码，维护代码，好心情好代码；

原则四：遵循必须的规则，其它的规则尽量follow；

## 1.3 规范分类

|  |  |  |
| --- | --- | --- |
| 词语 | 含义 | 说明 |
| M | Must | 必须遵守 |
| SS | Strong suggest | 强烈建议， |
| S | suggest | 建议，好的代码风格，不过可以考虑自身的习惯 |
| O | other | 其它的，能遵从最好，但更多是习惯问题 |

表1.3 规范分类

# 2 html 代码规范

## 2.1 通用规则

### 2.1.1 **协议头**

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

建议在指向图片或其他媒体文件、样式表和脚本的URL地址中省略http：https:协议部分，除非已知相应文件不能同时兼容2个协议。

<br>  
<!– Not recommended不推荐 –><br>  
<script src=”http://www.google.com/js/gweb/analytics/autotrack.js”></script><br>  
<!– Recommended 推荐–><br>  
<script src=”//www.google.com/js/gweb/analytics/autotrack.js”></script><br>  
/\* Not recommended不推荐 \*/<br>  
.example {<br>  
background: url(http://www.google.com/images/example);<br>  
}<br>  
/\* Recommended 推荐 \*/<br>  
.example {<br>  
background: url(//www.google.com/images/example);<br>  
}

## 2.2 通用格式规则

### 2.2.1缩进

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

不要使用tab制表符或制表符加空格的混合方式缩进

[?](http://www.mxria.com/html5/h201205111471.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | <ul><br>    <li>mxria.com<br>    <li>www.mxria.com<br>  </ul><br>  .example {<br>    color: blue;<br>  } |

### 2.2.2 大小写：**只使用小写**

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

所有的代码都应是小写的，包括元素名称、属性，属性值（除非text或CDATA的内容）、选择器、css属性、属性值（字符串除外）

[?](http://www.mxria.com/html5/h201205111471.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5 | <br>  <!-- Not recommended 不推荐--><br>  <A HREF="/">Home</A><br>  <!-- Recommended 推荐--><br>  <img src="mxria.png" alt="MXRIA"> |

### 2.2.3 尾随空格

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

尾随空格是不必要的，容易搞复杂diff文件。这个绝对是经验教训的总结！！！

[?](http://www.mxria.com/html5/h201205111471.htm)

|  |  |
| --- | --- |
| 1  2  3  4 | <!-- Not recommended --><br>  <p>What?\_<br>  <!-- Recommended --><br>  <p>Yes please.</p> |

### 2.2.4 **注释**

根据需要解释代码，这个就不多说了，团队开发这个非常重要，尽管很多时候大家不愿意遵守，但确实重要！！！

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

### 2.2.5 **TODO待定项**

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

尚未实现的或待定的内容一定要标识强调出来，利用TODO辨识，而非其他诸如@@来强调。  
在todo项中如果有必要列明联系人，比如负责人  
在TODO后追加一个冒号作为行动内容，例如 TODO：为美瑞网增加html5模板.

## 2.3 meta 原型格式

### 2.3.1**文档类型：使用HTML5**

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

使用<!DOCTYPE html>.HTML（text/html）类型文件相对XHTML（alication/xhtml+xml）文件，在浏览器及框架支持上和优化空间上都要好很多。

### 2.3.2 **字符编码**

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

通过明确声明字符编码，能够确保浏览器快速并容易的判断页面内容的渲染方式。这样做的好处是，可以避免在 HTML 中使用字符实体标记（character entity），从而全部与文档编码一致（约定一致采用UTF-8编码）。

<head>

<meta charset="UTF-8">

</head>

### 2.3.3 IE兼容模式

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

IE 支持通过特定的 <meta> 标签来确定绘制当前页面所应该采用的 IE 版本。除非有强烈的特殊需求，否则最好是设置为 edge mode，从而通知 IE 采用其所支持的最新的模式。

<meta http-equiv="X-UA-Compatible" content="IE=Edge">

## 2.4 html风格规则

### 2.4.1 **HTML 合法性验证**

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

合法的使用HTML，并利用w3c的工具（W3C HTML validator）进行检查。唯一例外就是因为性能原因需要压缩文件大小。  
原文如下：Use valid HTML code unless that is not possible due to otherwise unattainable performance goals  regarding file size. 但这个确实很难想象，省略标签节省的文件大小能有多少字节？但带来的问题可是风险居高哦！

[?](http://www.mxria.com/html5/h201205111471.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <br>  <!-- Not recommended 不推荐--><br>  <title>Test</title><br>  <article>This is only a test.<br>  <!-- Recommended 推荐--><br>  <!DOCTYPE html><br>  <meta charset="utf-8"><br>  <title>Test</title><br>  <article>This is only a test.</article> |

### 2.4.2 语义性

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

根据目的来合理使用HTML，这点对于HTML5而言尤为重要。下面例子可以对比，能实现同样的结果，但效率和可读性却有很大差别。

<!-- Not recommended -->

<div onclick="goToRecommendations();">All recommendations</div>

<!-- Recommended -->

<a href="recommendations/">All recommendations</a>

### 2.4.3 **多媒体后备**

**为多媒体提供备选内容**

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

For multimedia, such as images, videos, animated objects via canvas, make sure to offer alternative access. For images that means use of meaningful alternative text (alt) and for video and audio transcripts and captions, if available.

Providing alternative contents is important for accessibility reasons: A blind user has few cues to tell what an image is about without @alt, and other users may have no way of understanding what video or audio contents are about either.

(For images whose alt attributes would introduce redundancy, and for images whose purpose is purely decorative which you cannot immediately use CSS for, use no alternative text, as in alt="".)

<!-- Not recommended -->

<img src="spreadsheet.png">

<!-- Recommended -->

<img src="spreadsheet.png" alt="Spreadsheet screenshot.">

### 2.4.4 **关注点分离**

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

严格遵守将组织结构(markup)、表现样式(style)和行为动作(script)分开处理的原则，并且尽量使三者之间的关联度降到最小。这主要是基于维护性的考虑，通常，更新style文件或脚本文件比更改HTML文件的代价要小很多，试想一下，对于一个有超过10万页面的网站进行局部颜色调整，是每个html文件修改容易还是修改一个style文件容易？

<!-- Not recommended -->

<!DOCTYPE html>

<title>HTML sucks</title>

<link rel="stylesheet" href="base.css" media="screen">

<link rel="stylesheet" href="grid.css" media="screen">

<link rel="stylesheet" href="print.css" media="print">

<h1 style="font-size: 1em;">HTML sucks</h1>

<p>I’ve read about this on a few sites but now I’m sure:

<u>HTML is stupid!!1</u>

<center>I can’t believe there’s no way to control the styling of

my website without doing everything all over again!</center>

<!-- Recommended -->

<!DOCTYPE html>

<title>My first CSS-only redesign</title>

<link rel="stylesheet" href="default.css">

<h1>My first CSS-only redesign</h1>

<p>I’ve read about this on a few sites but today I’m actually

doing it: separating concerns and avoiding anything in the HTML of

my website that is presentational.

<p>It’s awesome!

### 2.4.5 **可选Tags：**

|  |  |  |
| --- | --- | --- |
| 规范类别 | S | 建议 |

**从文件大小的角度，我们应该省略可选的标签；从文档规范的角度，我们不应该省略可选的标签。**

我们选择规范的角度，所以，我们建议不要省略此类标签。

### 2.4.6 **实体引用**

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

假定开发团队内，文件和编辑器都是使用同样的编码格式（UTF-8），则没有必要使用实体引用的方式，例如&mdash;,&rdquo;, or &#x263a; 除非一些HTML中具有特定含义的字符，如”<“,或不可见字符如空格

<!-- Not recommended -->

The currency symbol for the Euro is &ldquo;&eur;&rdquo;.

<!-- Recommended -->

The currency symbol for the Euro is “€”.

### 2.4.7 **Type属性：省略**

|  |  |  |
| --- | --- | --- |
| 规范类别 | S | 建议 |

将样式表和脚本中的Type省略，除非你不是用的css或javascript，在HTML5中，该值默认是text/css和text/javascript

<!-- Not recommended -->

<link rel="stylesheet" href="//www.google.com/css/maia.css"

type="text/css">

<!-- Recommended -->

<link rel="stylesheet" href="//www.google.com/css/maia.css">

<!-- Not recommended -->

<script src="//www.google.com/js/gweb/analytics/autotrack.js"

type="text/javascript"></script>

<!-- Recommended -->

<script src="//www.google.com/js/gweb/analytics/autotrack.js"></script>

## 2.5 格式规则

### 2.5.1 通用格式

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

针对每个block，list或table元素另起一行，并在每个子元素前缩进。这样可读性好，例如：

Independent of the styling of an element (as CSS allows elements to assume a different role per display property), put every block, list, or table element on a new line.

Also, indent them if they are child elements of a block, list, or table element.

(If you run into issues around whitespace between list items it’s acceptable to put all li elements in one line. A linter is encouraged to throw a warning instead of an error.)

<blockquote>

<p><em>Space</em>, the final frontier.</p>

</blockquote>

<ul>

<li>Moe

<li>Larry

<li>Curly

</ul>

<table>

<thead>

<tr>

<th scope="col">Income

<th scope="col">Taxes

<tbody>

<tr>

<td>$ 5.00

<td>$ 4.50

</table>

### 2.5.2 引号

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

在属性中，使用双引号。

When quoting attributes values, use double quotation marks.

Use double ("") rather than single quotation marks ('') around attribute values.

<!-- Not recommended -->

<a class='maia-button maia-button-secondary'>Sign in</a>

<!-- Recommended -->

<a class="maia-button maia-button-secondary">Sign in</a>

## 2.6 元素建议

|  |  |  |
| --- | --- | --- |
| 规范类别 | S | 建议 |

文／熊猫猫超人（简书作者）  
原文链接：http://www.jianshu.com/p/8d291d823cc0  
著作权归作者所有，转载请联系作者获得授权，并标注“简书作者”。

### 2.6.1 html5 元素

 section 表示文档中的节、区段，可以和h1-h6一起来显示文档结构

 article 表示一块比较独立的内容或者主题内容，块级元素，比如blog的内容，报纸的文章

 aside 表示article以外的内容，而且应该和article有一定的关系，块级元素

 hgroup 表示一个文档、区段(section)的标题组合

 header 表示页眉,页头

 footer 表示页脚

 nav 表示导航内容

 figure 表示以相对独立的或外引的元素，如img video

 figcaption 表示 figure内容的标题

:::html

<!-- hgroup 示例 -->

<hgroup>

<h1>文档主标题</h1>

<h2>文档副标题</h2>

</hgroup>

<!-- figure 示例 -->

<figure>

<video src="ogg"></video>

<figcaption>Example</figcaption>

</figure>

<figure>

<img src="img" alt="Example image" />

<figcaption>Example image</figcaption>

</figure>

### 2.6.2 结构性元素

 p 表示段落。只能包含内联元素，不能包含块级元素;

 div 本身无特殊含义，可用于布局。几乎可以包含任何元素;

 br 表示换行符;

 hr 表示水平分割线;

 h1-h6 表示标题。其中 h1 用于表示当前页面最重要的内容的标题;

 blockquote 表示引用，可以包含多个段落。请勿纯粹为了缩进而使用 blockquote，大部分浏览器默认将 blockquote 渲染为带有左右缩进;

 pre 表示一段格式化好的文本;

### 2.6.3 头部元素

 title 每个页面必须有且仅有一个 title 元素;

 base 可用场景：首页、频道等大部分链接都为新窗口打开的页面;

 link link 用于引入 css 资源时，可省去 media(默认为all) 和 type(默认为text/css) 属性;

 style type 默认为 text/css，可以省去;

 script type 属性可以省去; 不赞成使用lang属性; 不要使用古老的<!– //–>这种hack脚本, 它用于阻止第一代浏览器（Netscape 1和Mosaic）将脚本显示成文字;

 noscript 在用户代理不支持 JavaScript 的情况下提供说明;

### 2.6.4 文本元素

 a a 存在 href 属性时表示链接，无 href 属性但有 name 属性表示锚点;

 em,strong em 表示句意强调，加与不加会引起语义变化，可用于表示不同的心情或语调; strong 表示重要性强调，可用于局部或全局，strong强调的是重要性，不会改变句意;

 abbr 表示缩写;

 sub,sup 主要用于数学和化学公式，sup还可用于脚注;

 span 本身无特殊含义;

 ins,del 分别表示从文档中增加(插入)和删除

### 2.6.5 媒体元素

* img 请勿将img元素作为定位布局的工具，不要用他显示空白图片; 必要时给img元素增加alt属性;
* object 可以用来插入Flash;

### 2.6.6 列表元素

 dl 表示关联列表，dd是对dt的解释; dt和dd的对应关系比较随意：一个dt对应多个dd、多个dt对应一个dd、多个dt对应多个dd，都合法; 可用于名词/单词解释、日程列表、站点目录;

 ul 表示无序列表;

 ol 表示有序列表, 可用于排行榜等;

 li 表示列表项，必须是ul/ol的子元素;

### 2.6.7 表单元素

* 推荐使用 button 代替 input，但必须声明 type;
* 表单元素的 name 不能设定为 action, enctype, method, novalidate, target, submit 会导致表单提交混乱

## 2.7 属性建议

|  |  |  |
| --- | --- | --- |
| 规范类别 | S | 建议 |

### 2.7.1 顺序建议

HTML 属性应当按照以下给出的顺序依次排列，确保代码的易读性。

* class
* id 、 name
* data-\*
* src、for、 type、 href
* title、alt
* aria-\*、 role

class用于标识高度可复用组件，因此应该排在首位。id 用于标识具体组件，应当谨慎使用（例如，页面内的书签），因此排在第二位。如：

:::html

<a class="..." id="..." data-modal="toggle" href="#">

Example link

</a>

<input class="form-control" type="text">

<img src="..." alt="...">

### 2.7.2 文件和目录命名约定

 *一律小写，必须是英文单词或者汉语拼音，以英语单词优先，多个单词以连字符 ( - ) 连接。* 只能出现小写引文字母，数字和连字符。

 很多浏览器会将含有这些词的作为广告拦截： ad、ads、adv、banner、sponsor、gg、guangg、guanggao等 页面中尽量避免采用以上词汇来命名。

 该命令规范适用于所有前端维护的内容和相关目录。(html, css, js, png, gif, jpg, ico)。

### 2.7.3 空格/标签/注释

 用两个空格来代替制表符（tab） —— 这是唯一能保证在所有环境下获得一致展现的方法；

 嵌套元素应当缩进一次 —— 即两个空格；

 对于属性的定义，确保全部使用双引号，绝不要使用单引号；

 不要省略可选的结束标签，如：</li>,</body>；

 合理书写注释，方便日后维护；

### 2.7.4 id和class命名约定

 id 和 class 的命名基本原则: *内容优先，表现为辅*。首先根据内容来命名，如:#header,#footer,.main-nav.如根据内容无法找到合适的命名，可以再结合表现进行命名，如：col-main, col-sub, col-extra,blue-box

 id 和 class 的名称一律小写，多个单词以连字符连接，如： main-wrap

 id 和 class 的名称只能出现，小写字母，数字和连字符( - )(js钩子除外)

 id 和 class 的名称*尽量使用英文单词命名*,如确实找不到合适的单词，可以使用拼音，如：zhidao-com

 在不影响语意的情况下，id 和 class 的名称 可以适当使用缩写，如: col, nav, hd, bd, fd( *缩写只用来表示结构，不允许写任何样式*)。不要自造缩写。

 class 对于选中命名.selected;对于hover，支持伪类使用:hover，不支持的使用 .hover，隐藏使用.hide 。

 id 和 class 的选择，如果只使用一次，使用id,如果使用多次使用class，如果需要和js交互，使用id,如果需要交互并且页面中有大量重复，请参见下一条。

 对于作为js钩子的 id 和 class 命名规则为以”J\_“开头(J,象形钩子的形状)，后面加上原应有的命名，在使用class的时候需要放在最前面。如:class="J\_tab-content some-mod-content"。（注意：钩子，不允许在css中定义任何的样式效果）

 很多浏览器会将含有这些词的作为广告拦截： ad、ads、adv、banner、sponsor、gg、guangg、guanggao等 页面中尽量避免采用以上词汇来命名。

### 2.7.5 选择器

1. 命名比较短的词，或者缩写的不允许直接定义样式，如：.title,.hd,.bd,.body.必须用上级节点进行限定,如：.recommend-mod .title
2. 当样式针对多个选择器时每个选择器占一行

/推荐的写法/  
a.btn,  
input.btn,  
input[type="button"] {  
......  
}

### 2.7.6 z-index

1. 自己写的z-index的值不能超过 100 (通用组的除外)
2. 页面中的元素内容的z-index不能超过10(popup poptip除外)，需要按照内容定义1 2 3 4不允许直接使用如1000，9999
3. popup poptip的z-index需要按照内容使用 99以下，10以上的值（11,12,13，也可以小于10），不允许直接使用1000，9999之类大值

现在通用z-index记录,使用时请避开和灵活使用

| **z-index** | **使用者** | **类型** |
| --- | --- | --- |
| <10 | page-content | 页面级别 |
| >10, <99 | page-popup | 页面级别 |
| 20 | usercard用户名片 | common通用 |
|  | MSG气泡消息 | common通用 |
|  | Dialog-Cover | common通用 |
|  | Dialog | common通用 |

# 3 css代码规范

## 3.1 css 风格规则

### 3.1.1 css validity

使用css 检验工具

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

Use valid CSS where possible.

Unless dealing with CSS validator bugs or requiring proprietary syntax, use valid CSS code.

Use tools such as the [W3C CSS validator](https://jigsaw.w3.org/css-validator/) to test.

Using valid CSS is a measurable baseline quality attribute that allows to spot CSS code that may not have any effect and can be removed, and that ensures proper CSS usage.

### 3.1.2 id 和class命名

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

这一条和之前得类似，主要原则是是用有意义或者通用得命名方式。命名需要反应元素得目的，取名尽量到位，减少改动的机会。

Instead of presentational or cryptic names, always use ID and class names that reflect the purpose of the element in question, or that are otherwise generic.

Names that are specific and reflect the purpose of the element should be preferred as these are most understandable and the least likely to change.

Generic names are simply a fallback for elements that have no particular or no meaning different from their siblings. They are typically needed as “helpers.”

Using functional or generic names reduces the probability of unnecessary document or template changes.

/\* Not recommended: meaningless \*/

#yee-1901 {}

/\* Not recommended: presentational \*/

.button-green {}

.clear {}

/\* Recommended: specific \*/

#gallery {}

#login {}

.video {}

/\* Recommended: generic \*/

.aux {}

.alt {}

### 3.1.3 id和class命名风格

尽可能短，但是必须要有意义。

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

Use ID and class names that are as short as possible but as long as necessary.

Try to convey what an ID or class is about while being as brief as possible.

Using ID and class names this way contributes to acceptable levels of understandability and code efficiency.

/\* Not recommended \*/

#navigation {}

.atr {}

/\* Recommended \*/

#nav {}

.author {}

### 3.1.4 类型选择器

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

尽量避免类型选择器，尽量避免同时使用元素名和id/class

void qualifying ID and class names with type selectors.

Unless necessary (for example with helper classes), do not use element names in conjunction with IDs or classes.

Avoiding unnecessary ancestor selectors is useful for [performance reasons](http://www.stevesouders.com/blog/2009/06/18/simplifying-css-selectors/).

/\* Not recommended \*/

ul#example {}

div.error {}

/\* Recommended \*/

#example {}

.error {}

### 3.1.5 简化属性

|  |  |  |
| --- | --- | --- |
| 规范类别 | S | 建议 |

Use shorthand properties where possible.

CSS offers a variety of [shorthand](https://www.w3.org/TR/CSS21/about.html#shorthand) properties (like font) that should be used whenever possible, even in cases where only one value is explicitly set.

Using shorthand properties is useful for code efficiency and understandability.

/\* Not recommended \*/

border-top-style: none;

font-family: palatino, georgia, serif;

font-size: 100%;

line-height: 1.6;

padding-bottom: 2em;

padding-left: 1em;

padding-right: 1em;

padding-top: 0;

/\* Recommended \*/

border-top: 0;

font: 100%/1.6 palatino, georgia, serif;

padding: 0 1em 2em;

### 3.1.6 为0省略单位

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

Omit unit specification after “0” values.

Do not use units after 0 values unless they are required.

margin: 0;

padding: 0;

### 3.1.7 省略前导0

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

Omit leading “0”s in values.

Do not use put 0s in front of values or lengths between -1 and 1.

font-size: .8em;

### 3.1.8 hex符号缩写

|  |  |  |
| --- | --- | --- |
| 规范类别 | S | 建议 |

Use 3 character hexadecimal notation where possible.

For color values that permit it, 3 character hexadecimal notation is shorter and more succinct.

/\* Not recommended \*/

color: #eebbcc;

/\* Recommended \*/

color: #ebc;

### 3.1.9 连接符

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

使用‘-’连字符，而不是‘\_’.

Separate words in ID and class names by a hyphen.

Do not concatenate words and abbreviations in selectors by any characters (including none at all) other than hyphens, in order to improve understanding and scannability.

/\* Not recommended: does not separate the words “demo” and “image” \*/

.demoimage {}

/\* Not recommended: uses underscore instead of hyphen \*/

.error\_status {}

/\* Recommended \*/

#video-id {}

.ads-sample {}

## 3.2 css 格式规则

### 3.2.1 元素顺序

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

有两种顺序，一种是按照含义分类，另外一种是按照字母，google建议按照字母。

我们采用按照含义分类的顺序。

1. 显示属性；
2. 元素位置；
3. 元素属性；
4. 元素内容属性；
5. css书写顺序：
6. 例子:
7. .header {
8. /\* 显示属性 \*/
9. display || visibility
10. list-style
11. position
12. top || right || bottom || left
13. z-index
14. clear
15. float
16. /\* 自身属性 \*/
17. width
18. max-width || min-width
19. height
20. max-height || min-height
21. overflow || clip
22. margin
23. padding
24. outline
25. border
26. background
27. /\* 文本属性 \*/
28. color
29. font
30. text-overflow
31. text-align
32. text-indent
33. line-height
34. white-space
35. vertical-align
36. cursor
37. content
38. }

下面是google的

Alphabetize declarations.

Put declarations in alphabetical order in order to achieve consistent code in a way that is easy to remember and maintain.

Ignore vendor-specific prefixes for sorting purposes. However, multiple vendor-specific prefixes for a certain CSS property should be kept sorted (e.g. -moz prefix comes before -webkit).

background: fuchsia;

border: 1px solid;

-moz-border-radius: 4px;

-webkit-border-radius: 4px;

border-radius: 4px;

color: black;

text-align: center;

text-indent: 2em;

### 3.2.2 缩进

内容需要缩进

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

Indent all block content.

Indent all [block content](https://www.w3.org/TR/CSS21/syndata.html#block), that is rules within rules as well as declarations, so to reflect hierarchy and improve understanding.

@media screen, projection {

html {

background: #fff;

color: #444;

}

}

### 3.2.3 句末分隔符

句末都需要；

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

Use a semicolon after every declaration.

End every declaration with a semicolon for consistency and extensibility reasons.

/\* Not recommended \*/

.test {

display: block;

height: 100px

}

/\* Recommended \*/

.test {

display: block;

height: 100px;

}

### 3.2.4 属性名分隔

属性名冒号后加空格

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

Use a space after a property name’s colon.

Always use a single space between property and value (but no space between property and colon) for consistency reasons.

/\* Not recommended \*/

h3 {

font-weight:bold;

}

/\* Recommended \*/

h3 {

font-weight: bold;

}

### 3.2.5 块申明

块和{之间需要一个空格

块名和{在同一行

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

Use a space between the last selector and the declaration block.

Always use a single space between the last selector and the opening brace that begins the [declaration block](https://www.w3.org/TR/CSS21/syndata.html#rule-sets).

The opening brace should be on the same line as the last selector in a given rule.

/\* Not recommended: missing space \*/

#video{

margin-top: 1em;

}

/\* Not recommended: unnecessary line break \*/

#video

{

margin-top: 1em;

}

/\* Recommended \*/

#video {

margin-top: 1em;

}

### 3.2.6选择器和申明都独立成行

选择器和申明都独立成行

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

Separate selectors and declarations by new lines.

Always start a new line for each selector and declaration.

/\* Not recommended \*/

a:focus, a:active {

position: relative; top: 1px;

}

/\* Recommended \*/

h1,

h2,

h3 {

font-weight: normal;

line-height: 1.2;

}

### 3.2.7 规则之间换行

|  |  |  |
| --- | --- | --- |
| 规范类别 | M | 必须遵守 |

Separate rules by new lines.

Always put a blank line (two line breaks) between rules.

html {

background: #fff;

}

body {

margin: auto;

width: 50%;

}

### 3.2.8 引号

CSS和html有点不同，建议使用单引号而非双引号

|  |  |  |
| --- | --- | --- |
| 规范类别 | SS | 强烈建议 |

Use single quotation marks for attribute selectors and property values.

Use single ('') rather than double ("") quotation marks for attribute selectors or property values. Do not use quotation marks in URI values (url()).

Exception: If you do need to use the @charset rule, use double quotation marks—[single quotation marks are not permitted](https://www.w3.org/TR/CSS21/syndata.html#charset).

/\* Not recommended \*/

@import url("//www.google.com/css/maia.css");

html {

font-family: "open sans", arial, sans-serif;

}

/\* Recommended \*/

@import url(//www.google.com/css/maia.css);

html {

font-family: 'open sans', arial, sans-serif;

}

## 3.3 Meta Rules

### 3.3.1 段落注释

建议段落加一定的注释

|  |  |  |
| --- | --- | --- |
| 规范类别 | S | 建议 |

Group sections by a section comment (optional).

If possible, group style sheet sections together by using comments. Separate sections with new lines.

/\* Header \*/

#adw-header {}

/\* Footer \*/

#adw-footer {}

/\* Gallery \*/

.adw-gallery {}

# 参考资料

<https://google.github.io/styleguide/htmlcssguide.xml>

<https://segmentfault.com/a/1190000007023192>

<http://www.jianshu.com/p/8d291d823cc0>

<https://github.com/fex-team/styleguide/blob/master/html.md>