**1.行内元素**

　　行内元素最常使用的就是span，其他的只在特定功能下使用，修饰字体<b>和<i>标签，还有<sub>和<sup>这两个标签可以直接做出平方的效果，而不需要类似移动属性的帮助，很实用。

　　行内元素特征：(1)设置宽高无效

　　　　　　　　　(2)对margin仅设置左右方向有效，上下无效；padding设置上下左右都有效，即会撑大空间

　　　　　　　　　(3)不会自动进行换行

text-align 指定了text在水平方向上的位置

vertical-align:middle的意义：元素的垂直中心点和父元素基线上1/2高度对齐，也就是，基线提高了，所以span就上去了。至于什么是基线，就像英语本的那条红线一样

HTML中的 a 标签加上 溢出隐藏 怎么会换行

问：要做成这种样式，前面一句话 ，后面是日期 然后我的代码是 HTML是： di

答：你给a标签再加个左浮动和 white-space:nowrap; text-overflow:ellipsis; 如果能行记得采纳

如何让li中内容显示超过长度后以省略号显示

答：white-space: nowrap; （不换行，一行显示） text-overflow: ellipsis; （溢[li标签超出隐藏](https://louqun.com/miaosha1703497/)出用省略号） -o-text-overflow: ellipsis; overflow: hidden; （多出的文字，隐藏） }

# HTML中meta标签viewpoint的作用

2016年10月19日 23:24:08 [闲逛上海滩](https://me.csdn.net/oscar92420aaa) 阅读数：2343更多

个人分类： [JavaScript](https://blog.csdn.net/oscar92420aaa/article/category/6080237)

1.在web上浏览的时候，响应式布局是好用的，放大缩小页面都可以实现页面变更，但是在手机上调试的时候死活不能用

2.因为css中@media是根据window的宽度来控制css的，所以我尝试输出了一下在window变更的时候window的width，结果发现PC上使用时一切正常，而手机上输出的一直是980。

原因很简单，手机上的浏览器是全屏的，我手机实际宽度是320像素，而我手机分辨率是980宽度，所以手机上打出来的是980而不是320

3.增加

<meta name="viewport" content="width=device-width, initial-scale=1.0">

以后发现手机window的width变为320了，即预期效果  
viewpoint说明

width 控制viewpoint的宽度，可以是固定值，也可以是device-width，即设备的宽度  
height 高度  
initial-scale 控制初始化缩放比例，1.0表示不可以缩放  
maximum-scale 最大缩放比例  
minimum-scale 最小缩放比例

可见如果不定义viewpoint的话，页面宽度以屏幕分辨率为基准，而设置以后可以根据设备宽度来调整页面，达到适配终端大小的效果

让select随表单一起提交需要满足两个条件：

   1.select需要有name属性（作为json数据的key值）

   2.select必须要在form内部

             提交内容为：如果option有value属性，则提交当前选中value属性的值，如果没有则会提交当前选择内容的具体内容

---------------------

作者：JLVJOE

来源：CSDN

原文：https://blog.csdn.net/afdasfggasdf/article/details/78647557

版权声明：本文为博主原创文章，转载请附上博文链接！

1、首先我们在网页[body](http://www.divcss5.com/html/h356.shtml)内最上面添加一个<span id="top" name="top"></span>  
2、我们再到body内，需要出现点击后转到顶部位置添加，<a href="#top">回到顶部</a>

**使用语法：**  
1、需要转到地方添加，<a href="#命名">文字</a>，注意href值是#开头+英文字母[命名](http://www.divcss5.com/jiqiao/j4.shtml)  
2、需要在被转到位置添加，<span id="命名" name="命名"></span>，注意id和name值相同并且与a命名相同

# [js常用返回网页顶部几种方法](https://www.cnblogs.com/NatChen/p/6296744.html)

**一、使用锚标记**此方法最简单，只需在body下放个隐藏的锚点标记，内容如下：

|  |  |
| --- | --- |
| 代码如下 | 复制代码 |
| <a name="top" id="top"></a> | |

然后，在网页底部放一个访问链接即可：

|  |  |
| --- | --- |
| 代码如下 | 复制代码 |
| <a href="#top" target="\_self">返回顶部</a> | |

此方法效果是一次直接跳到顶部，而且URL地址栏会显示个#top，追求完美的可使用下面方法。

**二、使用JS scrollTo函数**[javascript](http://www.111cn.net/js_a/js.html) scroll函数（scrollBy scrollTo）是用来滚动页面到指定位置，格式定义如下：

■scrollBy(xnum,ynum)  
■scrollTo(xpos,ypos)  
其中：

■xnum，ynum分别指水平、垂直滚动多少像素，正值表示向右或向下滚动，负值表示向左或向上滚动  
■xpos，ypos分别指水平和垂直坐标  
使用示例：

|  |  |
| --- | --- |
| 代码如下 | 复制代码 |
| <a href="javascript:scrollTo(0,0);">返回顶部</a> | |

**三、scrollBy慢速滚动返回顶部**本方式使用上面提到的scrollBy函数，每次只滚动定量像素，整体看起来有点滚动效果，代码如下：

|  |  |
| --- | --- |
| 代码如下 | 复制代码 |
| var sdelay=0; function returnTop() {  window.scrollBy(0,-100);//Only for y vertical-axis  if(document.body.scrollTop>0) {    sdelay= setTimeout('returnTop()',50);  } } | |

scrollBy函数第二个参数我设了-100，越大（比如-10）滚动越慢，越小滚动越快，启动滚动只需在页面底部加个链接：

|  |  |
| --- | --- |
| 代码如下 | 复制代码 |
| <a href="javascript:returnTop();">返回顶部</a> | |

**JQuery实现返回顶部功能**

首先需要在顶部添加如下html元素：

<p id="back-to-top"><a href="#top"><span></span>返回顶部</a></p>  
其中a标签指向锚点top，可以在顶部防止一个<a name="top"></a>的锚点，这样在浏览器不支持js时也可以实现返回顶部的效果了。

要想让返回顶部的图片显示在右侧，还需要一些css样式，如下：

|  |  |
| --- | --- |
| 代码如下 | 复制代码 |
| /\*returnTop\*/ p#back-to-top{     position:fixed;     display:none;     bottom:100px;     right:80px; } p#back-to-top a{     text-align:center;     text-decoration:none;     color:#d1d1d1;     display:block;     width:64px;     /\*使用CSS3中的transition属性给跳转链接中的文字添加渐变效果\*/     -moz-transition:color 1s;     -webkit-transition:color 1s;     -o-transition:color 1s; } p#back-to-top a:hover{     color:#979797; } p#back-to-top a span{     background:transparent url(/static/imgs/sprite.png?1202) no-repeat -25px -290px;     border-radius:6px;     display:block;     height:64px;     width:56px;     margin-bottom:5px;     /\*使用CSS3中的transition属性给<span>标签背景颜色添加渐变效果\*/     -moz-transition:background 1s;     -webkit-transition:background 1s;     -o-transition:background 1s; } #back-to-top a:hover span{     background:transparent url(/static/imgs/sprite.png?1202) no-repeat -25px -290px; } | |

上面样式中的背景图片是雪碧图，下面提供两个单独的返回顶部图片方便朋友们使用：

有了html和样式，我们还需要用js控制返回顶部按钮，在页面滚动时渐隐渐现返回顶部按钮。

|  |  |
| --- | --- |
| 代码如下 | 复制代码 |
| <script src="http://ajax.microsoft.com/ajax/jQuery/[jquery](http://www.111cn.net/list-139/)-1.7.2.min.js"></script> <script> $(function(){         //当滚动条的位置处于距顶部100像素以下时，跳转链接出现，否则消失         $(function () {             $(window).scroll(function(){                 if ($(window).scrollTop()>100){                     $("#back-to-top").fadeIn(1500);                 }                 else                 {                     $("#back-to-top").fadeOut(1500);                 }             });              //当点击跳转链接后，回到页面顶部位置              $("#back-to-top").click(function(){                 $('body,html').animate({scrollTop:0},1000);                 return false;             });         });     }); </script> | |

[**好文要顶**](javascript:void(0);) [**关注我**](javascript:void(0);) [**收藏该文**](javascript:void(0);) **[https://common.cnblogs.com/images/icon_weibo_24.png](javascript:void(0);)** **[https://common.cnblogs.com/images/wechat.png](javascript:void(0);)**

# 前端人员必看CSS命名规范

2017年07月25日 11:35:49 [iyox](https://me.csdn.net/u012182639) 阅读数：10211

作为前端的设计人员，对于CSS的接触，就像吃饭一样。所以CSS命名规范 命名是 前端人员必看的。  
  
文章整理了Web前端开发中的各种CSS规范，包括文件规范、注释规范、命名规范、书写规范、测试规范等。  
  
一、文件规范1、文件均归档至约定的目录中。具体要求通过豆瓣的CSS规范进行讲解：所有的CSS分为两大类：通用类和业务类。通用的CSS文件，放在如下目录中：  
  
  
业务类的CSS是指和具体产品相关的文件，放在如下目录中：  
  
  
  
外联CSS文件适用于全站级和产品级通用的大文件。内联CSS文件适用于在一个或几个页面共用的CSS。另外一对具体的CSS进行文档化的整理。如：  
  
  
- util-01 reset /css/core/reset.css  
- util-02 通用模块容器 /css/core/mod.css  
- ui-01. 喜欢按钮 /css/core/fav\_btn.css  
- ui-02. 视频/相册列表项 /css/core/media\_item.css  
- ui-03. 评星 /css/core/rating.css  
- ui-04. 通用按钮 /css/core/common\_button.css  
- ui-05. 分页 /css/core/pagination.css  
- ui-06. 推荐按钮 /css/core/rec\_btn.css  
- ui-07. 老版对话框 /css/core/old\_dialog.css  
- ui-08. 老版Tab /css/core/old\_tab.css  
- ui-09. 老版成员列表 /css/core/old\_userlist.css  
- ui-10. 老版信息区 /css/core/notify.css  
- ui-11. 社区用户导航 /css/core/profile\_nav.css  
- ui-12. 当前大社区导航 /css/core/site\_nav.css  
- ui-13. 加载中 /css/lib/loading.css  
  
2、文件引入可通过外联或内联方式引入。  
  
  
- 外联方式：（类型声明type=”text/css”可以省略）  
- 内联方式： （类型声明type=”text/css”可以省略）  
link和style标签都应该放入head中，原则上，不允许在html上直接写样式。避免在CSS中使用@import，嵌套不要超过一层。  
  
  
3、文件名、文件编码及文件大小  
  
  
- 文件名必须由小写字母、数字、中划线组成  
- 文件必须用UTF-8编码，使用UTF-8（非BOM），在HTML中指定UTF-8编码，在CSS中则不需要特别指定因为默认就是UTF-8。  
- 单个CSS文件避免过大（建议少于300行）  
二、注释规范  
  
1、文件顶部注释（推荐使用）  
  
- /\*  
- \* @description: 中文说明  
- \* @author: name  
- \* @update: name (2013-04-13 18:32)  
- \*/  
复制代码2、模块注释  
  
- /\* module: module1 by 张三 \*/  
- …  
- /\* module: module2 by 张三 \*/  
复制代码模块注释必须单独写在一行  
  
  
3、 单行注释与多行注释  
  
  
/\* this is a short comment  
\*/单行注释可以写在单独一行，也可以写在行尾，注释中的每一行长度不超过40个汉字，或者80个英文字符。/\*  
\* this is comment line 1.  
\* this is comment line 2.  
\*/多行注释必须写在单独行内  
  
  
  
4、特殊注释  
  
  
/\* TODO: xxxx by name 2013-04-13 18:32 \*//\* BUGFIX: xxxx by name  
2012-04-13 18:32 \*/用于标注修改、待办等信息  
  
  
5、区块注释  
  
  
- /\* Header \*/  
- /\* Footer \*/  
- /\* Gallery \*/  
复制代码对一个代码区块注释（可选），将样式语句分区块并在新行中对其注释。  
  
三、命名规范  
  
使  
用有意义的或通用的ID和class命名：ID和class的命名应反映该元素的功能或使用通用名称，而不要用抽象的晦涩的命名。反映元素的使用目的是首  
选；使用通用名称代表该元素不表特定意义，与其同级元素无异，通常是用于辅助命名；使用功能性或通用的名称可以更适用于文档或模版变化的情况。  
  
- /\* 不推荐: 无意义 \*/ #yee-1901 {}  
- /\* 不推荐: 与样式相关 \*/ .button-green {}.clear {}  
- /\* 推荐: 特殊性 \*/ #gallery {}#login {}.video {}  
- /\* 推荐: 通用性 \*/ .aux {}.alt {}  
常 用命名（多记多查英文单词）：page、wrap、layout、header(head)、footer(foot、ft)、  
content(cont)、menu、nav、main、submain、sidebar(side)、logo、banner、  
title(tit)、popo(pop)、icon、note、btn、txt、iblock、window(win)、tips等  
  
ID和class命名越简短越好，只要足够表达涵义。这样既有助于理解，也能提高代码效率。  
  
- /\* 不推荐 \*/ #navigation {}.atr {}  
- /\* 推荐 \*/ #nav {}.author {}  
  
类型选择器避免同时使用标签、ID和class作为定位一个元素选择器；从性能上考虑也应尽量减少选择器的层级。  
  
  
- /\* 不推荐 \*/ul#example {}div.error {}  
- /\* 推荐 \*/#example {}.error {}  
命名时需要注意的点：  
  
- 规则命名中，一律采用小写加中划线的方式，不允许使用大写字母或 \_  
- 命名避免使用中文拼音，应该采用更简明有语义的英文单词进行组合  
- 命名注意缩写，但是不能盲目缩写，具体请参见常用的CSS命名规则  
- 不允许通过1、2、3等序号进行命名  
- 避免class与id重名  
- id用于标识模块或页面的某一个父容器区域，名称必须唯一，不要随意新建id  
- class用于标识某一个类型的对象，命名必须言简意赅。  
- 尽可能提高代码模块的复用，样式尽量用组合的方式  
-  
规则名称中不应该包含颜色（red/blue）、定位（left/right）等与具体显示效果相关的信息。应该用意义命名，而不是样式显示结果命名。  
1、常用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：regsiter  
- 搜索：search  
- 功能区：shop  
- 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：guild  
- 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、书写规范  
  
1、排版规范(1)使用4个空格，而不使用tab或者混用空格+tab作为缩进；(2)规则可以写成单行，或者多行，但是整个文件内的规则排版必须统一；单行形式书写风格的排版约束  
  
- 如果是在html中写内联的css，则必须写成单行；  
- 每一条规则的大括号 { 前后加空格 ；  
- 每一条规则结束的大括号 } 前加空格；  
- 属性名冒号之前不加空格，冒号之后加空格；  
- 每一个属性值后必须添加分号; 并且分号后空格；  
- 多个selector共用一个样式集，则多个selector必须写成多行形式 ；  
多行形式书写风格的排版约束  
  
- 每一条规则的大括号 { 前添加空格;  
- 多个selector共用一个样式集，则多个selector必须写成多行形式 ;  
- 每一条规则结束的大括号 } 必须与规则选择器的第一个字符对齐 ;  
- 属性名冒号之前不加空格，冒号之后加空格;  
- 属性值之后添加分号;  
  
2、属性编写顺序  
  
  
- 显示属性：display/list-style/position/float/clear …  
- 自身属性（盒模型）：width/height/margin/padding/border  
- 背景：background  
- 行高：line-height  
-  
文本属性：color/font/text-decoration/text-align/text-indent/vertical-align/white-space/content…  
- 其他：cursor/z-index/zoom/overflow  
- CSS3属性：transform/transition/animation/box-shadow/border-radius  
- 如果使用CSS3的属性，如果有必要加入浏览器前缀，则按照 -webkit- / -moz- / -ms- / -o- /  
std的顺序进行添加，标准属性写在最后。  
- 链接的样式请严格按照如下顺序添加： a:link -> a:visited -> a:hover -> a:active  
  
3、规则书写规范  
  
  
- 使用单引号，不允许使用双引号;  
- 每个声明结束都应该带一个分号，不管是不是最后一个声明;  
- 除16进制颜色和字体设置外，CSS文件中的所有的代码都应该小写;  
- 除了重置浏览器默认样式外，禁止直接为html tag添加css样式设置;  
- 每一条规则应该确保选择器唯一，禁止直接为全局.nav/.header/.body等类设置属性;  
  
4、代码性能优化  
  
  
- 合并margin、padding、border的-left/-top/-right/-bottom的设置，尽量使用短名称。  
- 选择器应该在满足功能的基础上尽量简短，减少选择器嵌套，查询消耗。但是一定要避免覆盖全局样式设置。  
- 注意选择器的性能，不要使用低性能的选择器。  
- 禁止在css中使用\*选择符。  
- 除非必须，否则，一般有class或id的，不需要再写上元素对应的tag。  
- 0后面不需要单位，比如0px可以省略成0，0.8px可以省略成.8px。  
- 如果是16进制表示颜色，则颜色取值应该大写。  
- 如果可以，颜色尽量用三位字符表示，例如#AABBCC写成#ABC 。  
- 如果没有边框时，不要写成border:0，应该写成border:none 。  
- 尽量避免使用AlphaImageLoader 。  
- 在保持代码解耦的前提下，尽量合并重复的样式。  
- background、font等可以缩写的属性，尽量使用缩写形式 。  
5、CSS Hack的使用  
  
请不用动不动就使用浏览器检测和CSS  
Hacks，先试试别的解决方法吧！考虑到代码高效率和易管理，虽然这两种方法能快速解决浏览器解析差异，但应被视为最后的手段。在长期的项目中，允许使用hack只会带来更多的hack，你越是使用它，你越是会依赖它！  
  
  
推荐使用下面的：  
  
  
  
<ignore\_js\_op>  
  
[/url]  
  
6、字体规则  
  
  
- 为了防止文件合并及编码转换时造成问题，建议将样式中文字体名字改成对应的英文名字，如：黑体(SimHei) 宋体(SimSun) 微软雅黑  
(Microsoft Yahei，几个单词中间有空格组成的必须加引号)  
- 字体粗细采用具体数值，粗体bold写为700，正常normal写为400  
-  
font-size必须以px或pt为单位，推荐用px（注：pt为打印版字体大小设置），不允许使用xx-small/x-small/small/medium/large/x-large/xx-large等值  
- 为了对font-family取值进行统一，更好的支持各个操作系统上各个浏览器的兼容性，font-family不允许在业务代码中随意设置  
五、其他规范  
  
- 不要轻易改动全站级CSS和通用CSS库。改动后，要经过全面测试。  
- 避免使用filter  
- 避免在CSS中使用expression  
- 避免过小的背景图片平铺。  
- 尽量不要在CSS中使用!important  
- 绝对不要在CSS中使用”\*”选择符  
-  
层级(z-index)必须清晰明确，页面弹窗、气泡为最高级（最高级为999），不同弹窗气泡之间可在三位数之间调整；普通区块为10-90内10的倍数；区块展开、弹出为当前父层级上个位增加，禁止层级间盲目攀比。  
- 背景图片请尽可能使用sprite技术, 减小http请求, 考虑到多人协作开发, sprite按照模块、业务、页面来划分均可。  
六、测试规范  
1、了解浏览器特效支持为了页面性能考虑，如果浏览器不支持CSS3相关属性的，则该浏览器的某些特效将不再支持，属性的支持情况如下表所示（Y为支持，N为不支持）：<ignore\_js\_op>  
  
2、 设定浏览器支持标准  
  
<ignore\_js\_op>  
  
  
  
- A级－交互和视觉完全符全设计的要求  
- B级－视觉上允许有所差异，但不破坏页面的整体效果  
- C级－可忽略设计上的细节，但不防碍使用  
3、常用样式测试工具  
W3C CSS validator：<http://jigsaw.w3.org/css-validator/>

**页面结构命名**

1. page：代表整个页面，用于最外层。
2. wrap：外套，将所有元素包在一起的一个外围包，用于最外层
3. wrapper：页面外围控制整体布局宽度，用于最外层
4. container：一个整体容器，用于最外层
5. head、header：页头区域，用于头部
6. nav：导航条
7. content：内容，网站中最重要的内容区域，用于网页中部主体
8. main：网站中的主要区域（表示最重要的一块位置），用于中部主体内容
9. column：栏目
10. sidebar：侧栏
11. foot、footer：页尾、页脚。网站一些附加信息放置区域，（或命名为copyright）用于底部

**导航命名**

1. nav、navbar、navigation、nav-wrapper：导航条或导航包，代表横向导航
2. topnav：顶部导航
3. mainbav：主导航
4. subnav：子导航
5. sidebar：边导航
6. leftsidebar 或 sidebar\_a ：左导航
7. rightsidebar 或 sidebar\_b：右导航
8. title：标题
9. summary：摘要/li>
10. menu：菜单。区域包含一般的链接和菜单
11. submenu：子菜单
12. drop：下拉
13. dorpmenu：下拉菜单
14. links：链接菜单

**功能命名**

1. logo：标记网站logo标志
2. banner：标语、广告条、顶部广告条
3. login：登陆，（例如登录表单：form-login）
4. loginbar：登录条
5. regsiter：注册
6. tool、toolbar：工具条
7. search：搜索
8. searchbar：搜索条
9. searchlnput：搜索输入框
10. shop：功能区，表示现在的
11. icon：小图标
12. label：商标
13. homepage：首页
14. subpage：二级页面子页面
15. hot：热门热点
16. list：文章列表，（例如新闻列表：list-news）
17. scroll:滚动
18. tab：标签
19. sitemap：网站地图
20. msg 或 message：提示信息
21. current：当前的
22. joinus：加入
23. status：状态
24. btn：按钮，（例如搜索按钮可写成：btn-search）
25. tips：小技巧
26. note：注释
27. guild：指南
28. arr、arrow：标记箭头
29. service：服务
30. breadcrumb：(即页面所处位置导航提示）
31. download：下载
32. vote：投票
33. siteinfo：网站信息
34. partner：合作伙伴
35. link、friendlink：友情链接
36. copyright：版权信息
37. siteinfoCredits：信誉
38. siteinfoLegal：法律信息

**CSS样式命名**

1. 对齐样式命名：left（左边内容）、center（中间内容）、right（右边内容）等；
2. 颜色英文命名：red（红色）、green（绿色）、yellow（黄色），又或者border\_red（红色边框）等；
3. 颜色代码命名：f00（红色）、ff0（黄色）、f90（橙色）等；
4. 文字大小命名：font12px（字体12像素）、font16px（字体16像素）等；
5. 页面线条命名：line\_x （横线）、line\_y （纵线）或 line\_red（红线）、line\_black（黑线）
6. 图片图标命名：pic\_1.jpg、pic\_2.jpg 或 ico\_1.gif、ico\_2.gif
7. 页面广告命名：ad\_01、ad\_02
8. 背景框架命名：nav\_bg（代表导航条的背景图片位置）、tool\_bg（代表工具栏的背景图片位置）

**CSS样式表命名**

1. index.css：单独为首页建立样式
2. head.css：头部样式，多个页面头部设计风格相同时使用。
3. base.css：共用样式。
4. style.css：独立页面所使用的样式文件。
5. global.css：页面样式基础，全局公用样式，页面中必须包含。
6. layout.css：布局、版面样式，公用类型较多时使用，一般用在首页级页面和产品类页面中
7. module.css：模块，用于产品类页，也可与其它样式配合使用。
8. master.css：主要的样式表
9. columns.css：专栏样式
10. themes.css：主体样式
11. forms.css：表单样式
12. mend.css：补丁，基于以上样式进行的私有化修补。