编程规范 HTML + CSS

命名规范

> HTML + CSS 命名规范

```
1. 命名需要是具备语义性的单词，不能用 数字 拼音 符号
     正确示范 ： wrap description title  content
     错误示范 ： aaaa a1 $we 4tdds
2. 命名需要多个单词连接的情况下，标记语言中可以使用 _ - 进行连接 不能直接单词拼接 或者驼峰命名
注意：书写风格必须统一 不容许出现 _ -一起使用的情况  PS：更推荐使用 - 这样更清晰.
     正确示范 ： header-nav content-left slide-bar
     错误示范 ： headernav slideBar ContentLeft
3. 命名需要进行适当的缩写，单词连接层级不要超过4层
     正确示范 ： head-tit-ico
     错误示范 ： header-title-left-logo-icon
4. 不容许通过1、2、3等序号进行命名
     正确示范：content-product
     错误示范：content1
             content2

另：
1. 避免class与ID重名
2. id用于识别模块或一级结构区域 且必须唯一 不要改动线上项目ID名称
```

常用命名

头	header	内容	content	尾	footer	导航	nav
子导航	subnav	栏目	column	主体	main	新闻	news
版权	copyright	文章列表	list	加入	joinus	合作伙伴	partner
标志	logo	侧栏	sidebar	横幅	banner	状态	status
菜单	menu	子菜单	submenu	滚劢	scroll	搜索	search
标签页	tab	提示信息	msg	小技巧	tips	标题	title
指南	guild	服务	service	热点	hot	下载	download
注册	regsiter	登录条	loginbar	按钮	btn	投票	vote
注释	note	友情链接	friend-link	外套	wrap	面包屑	bread-crumb
当前	current	购物车	shop	图标	icon	文本	txt

的	current	购物车	shop	图标	icon	文本	txt
头	**header**	**内容**	**content**	**尾**	**footer**	**导航**	**nav**
容器	container	wrap					

img 标签四要素

> 为图像指定 height 和 width 属性是一个好习惯。如果设置了这些属性，就可以在页面加载时为图像预留空间。如果没有这些属性，浏览器就无法了解图像的尺寸，也就无法为图像保留合适的空间，因此当图像加载时，页面的布局就会发生变化。
> img标签四要素 width height
> 四要素的width和height只需要写数字 图片的原始宽度和高度 不需要添加单位, 爬虫需要的是原始图片数据
> alt不仅是图片替代文本而且起到让搜索引擎蜘蛛了解图片的内容,便于百度图片的收录以及优化。
> alt还可以在图片无法加载的时候 显示图片描述文字 让用户了解这个图是做什么的
> title不属于四要素 四要素有 src、width、height、alt

正确示范 : `<img src='xxx/xxx.png' width='125' height='125' alt='示范'>`

错误示范 : `<img src='xxx/xxx.png' width='125px' height='225px'>`

标签空行

> HTML标签内部原则上不容许出现无意义的空行 但如果为源码 可基于保证可读性的原则上 对大块分区之间进行空行或 补充注释说明, 禁止出现连续两行以及以上的空行行为

```
<div class="header">
 ......
</div>
<!-- content start -->
<div class="content">

</div>

<div class="footer">

</div>
```

转义符号

> 为了避免在多类型HTML标签编辑 查阅 传输中 将符号误认为结构元素或关键字 在使用符号的时候统一用转义符

正确示范 :`<p>3>2</p>`

错误示范 :`<p>3>2</p>`

标签书写与嵌套

> 合理遵守标签书写与嵌套规则 满足w3c规范的同时避免不必要的兼容问题

1. **所有标签书写必须在半角英文状态下小写输入**
2. **双标签必须闭合　单标签不写闭合符 (/)**
3. **子级标签相对父级标签进行一次TAB缩进 1tab = 2空格**
4. **属性值必须带有引号(单引号 or 双引号) 保证引号风格统一**
5. **ul>li ol>li dl>dt>dd 是成套　并且为绝对父子关系标签 不容许在ul直接子代中插入非li的标签**
6. **a标签内部不容许嵌套a标签**
7. **除了a之外的行内元素不容许嵌套块元素**
8. **p dt h内不能嵌套块元素**

图片文件命名

1. 图片后缀命名一律小写。
2. 使用间隔符 - 进行连接。
   一般背景图片以bg-开头，
   按钮图片以btn-开头，
   图标图片以icon-开头，
   聚合图以spr-开头，后跟英文单词，如果名称过长，适当使用缩写

CSS书写

注释

CSS规则是使用　　/*　需要注释的内容　*/　进行注释的，即在需要注释的内容前使用“/*”标记开始注释，在内容的结尾使用“*/”结束。

例如：

```
p {
  font-size: 14px;                  /* 所有的字体是14像素大小*/
}
```

文件顶部注释

```
/*
 * @description: xxxxx 中文说明
 * @author: zhifu.wang
 * @update: zhifu.wang (2012-10-17 18:32)
 */
```

模块注释

```
/* module: module1 by zhifu.wang */ ….
/* module: module2 by zhifu.wang */
模块注释必须单独写在一行
```

简单注释

```
单行注释：/* this is a short comment */

单行注释可以单独写在一行，也可以写在行尾

多行注释：
/*
 * this is comment line 1.
 * this is comment line 2.
*/

多行注释必须写在单独行
```

特殊注释

```
用于标注修改、待办等信息
/* TODO: xxxx by zhifu.wang 2012-10-18 18:32 */
/* BUGFIX: xxxx by zhifu.wang 2012-10-18 18:32 */
长度要求
注释中的每一行长度丌超过 40 个汉字，或者 80 个英文字符
```

空格规范

【强制】 选择器 与 { 之间必须包含空格。

示例： .selector { }

【强制】 属性名 与之后的 : 之间不允许包含空格， : 与 属性值 之间必须包含空格。

示例：

```
color: red;
```

选择器规范

【建议】 选择器的嵌套层级应不大于 3 级，位置靠后的限定条件应尽可能精确。

示例：

```
/* 规范 */
#username input {}
.comment .avatar {}

/* 不规范 */
.page .header .login #username input {}
.comment div * {}
```

【建议】 并集选择器在逗号后换行书写下一个选择器

示例：

```
/* 规范 */
body,
ul,
p {

}

/* 不规范 */
body, ul, p {}
```

属性规范

【强制】 属性定义必须另起一行。

示例：

```
/* 规范 */
.selector {
    margin: 0;
    padding: 0;
}

/* 不规范 */
.selector { margin: 0; padding: 0; }
```

【强制】 属性定义后必须以分号结尾。

示例：

```
/* 规范 */
.selector {
    margin: 0;
}

/* 不规范 */
.selector {
    margin: 0
}
```

大小写规定

```
css虽然不区分大小写 按照惯例和规定 : 所有css文件中的代码都小写,
```

```
/* 规范 */
#username input {
    text-align: center;
}


/* 不规范 */
#username input {
    TEXT-ALIGN: center;
}
```

CSS内引号

> CSS内所有需要用到引号的部分统一推荐使用 单引号

```
content: '.'
```

CSS3兼容前缀

> 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-
> 的顺序进行添加，标准属性写在最后，并且属性名称要对齐，例如：

```
div.animation-demo {
    -webkit-animation: mymove 5s infinite;
    -moz-animation: mymove 5s infinite;
    -o-animation: mymove 5s infinite;
    animation: mymove 5s infinite;
}
```

背景URL

> 背景图片 background-image: url() 括号内部地址不使用引号包裹

```
background-image: url(img/banner.jpg)
```

CSS书写顺序

> 良好的CSS书写顺序是前端工程师需要遵守并维护的重要规范

1. 位置属性 display position float overflow z-index list-style clear等可以决定元素渲染位置或层级 以及能够影响其他元素渲染位置或层级的属性
2. 自身属性 width height margin padding border background line-height 等可以影响盒子自身展示的属性
3. 文本属性 color font- text- word- 等作用于文本的样式属性
4. 其他与新增属性 cursor zoom transform box-shadow 等新增属性

> 原理

1. 浏览器解读HTML是从上之下单行解析，如果没有良好的书写顺序，例如先解析了 width height 那预渲染的时候 就会从默认位置（当前文档流左上角）进行渲染
2. 如果后续解析到了 位置属性 浏览器需要擦除之前 渲染好的模型 重新根据 渲染基准点(左上角)位置和层级 进行重绘，这样导致浏览器会重复解析同一个元素 造成不必要的重绘
3. 良好的书写顺序是 BAT等一线互联网企业都遵守的CSS书写规范 他可以帮助团队成员协作的时候 迅速排查和后期维护

错误示范

```
.selector {
    font-size: 12px;
    width: 150px;
    float: left;
}
```

正确示范

```
.selector {
    float: left;
    width: 150px;
    font-size: 12px;
}
```

链接伪类顺序

链接的样式请严格按照：a:link-> a:visited-> a:hover-> a:active（LoVeHAte）的顺序添加

CSS复合写法与单例写法

如果对目标样式的子属性需求小于3时 进行单例写法 避免复合写法造成的computed浪费
当子属性需求 大于等于3时 进行复合式写法 避免单例写法过于冗余和字节浪费
PS
margin和padding建议: 合并margin、padding、border的-left/-top/-right/-bottom的设置，尽量使用短名称。

错误示范

```
.selector {
    background: red;
    margin-top: 20px;
    margin-bottom: 20px;
    margin-left: 20px;
    margin-right: 20px;
}
```

正确示范

```
div {
    background-color: red;
    margin: 20px;
}
```

布局方式选择

> 文档流 > 盒子模型距离调整(margin+padding) > 浮动 > 定位

定位元素会脱标并且独立新开文档流层级，高度依赖定位会导致浏览器压力大，并且在后期维护中因为定位元素并不能跟随文档流进行流动，所以维护成本高。

实际开发中 能用文档流+盒子模型处理的布局 轻易不使用浮动 能用浮动处理的布局 不要使用定位 。 只有在最关键的时刻才使用定位进行布局调整。

定位z-index取值范围

> 合理规划z-index的取值范围 避免和他人组件层级冲突 需要组内协商

公共头部导航	**1999 - 2100**
banner与二维码等弹出层	999 - 1900
页面公共底部	1999 - 2100
页面公共组件	-1 - 999

css细节优化

> 适当缩写 合理规划 兼容调整

1. 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
2. HEX 16进制颜色代码 对 AABBCC格式缩写为 ABC 例: #336688 => #368
3. 如果没有边框时，不要写成border:0，应该写成border:none
4. 为了 SEO 和页面可用性，请使用 text-indent 来隐藏文本内容
5. 使用背景图标请使用精灵图(csssprites) 合并图片 制作时颜色相近的放到临近位置
6. 在H5 DTD规则下 省略 script link style 标签的 type属性