

CSS 编码规范

一、命名规则

Rule 1：分类命名使用单个字母+ `-` 为前缀。

规则说明：

- 为了统一基础语义的理解和命名，在实际编码过程中，请参考以下示例表格为样式命名。
- 如果以下四类不能满足需求，你可以另外定义一个或多个大类，但必须符合单个字母+“-”为前缀的命名规则，即 `.x-` 的格式，并更新至示例表格中。
- 特殊：`.j-`将被专用于JS获取节点，请勿使用`.j-`定义样式。

注：在你样式中的选择器总是要以下面这四类开头，然后在里面使用后代选择器。

示例：

| 分类名称 | 命名 | 语义 | 命名 |
|------------|-----|-------|--------|
| 布局（grid） | .g- | 文档 | doc |
| | | 主体 | bd |
| | | 尾部 | ft |
| | | 主栏 | main |
| | | 主栏子容器 | mainc |
| | | 侧栏 | side |
| | | 侧栏子容器 | sidec |
| | | 盒容器 | wrap |
| 模块（module） | .m- | 导航 | nav |
| | | 子导航 | subnav |
| | | 面包屑 | crumb |
| | | 菜单 | menu |
| | | 选项卡 | tab |
| | | 标题区 | header |

| | | | |
|---------------|-----|------|-----------|
| | | 内容区 | content |
| | | 列表 | list |
| | | 表格 | tbl |
| | | 表单 | form |
| | | 热点 | hot |
| | | 排行 | top |
| | | 登录 | login |
| | | 标志 | logo |
| | | 广告 | ad |
| | | 搜索 | search |
| | | 幻灯 | slide |
| | | 提示 | tips |
| | | 帮助 | help |
| | | 新闻 | news |
| | | 下载 | dl |
| | | 注册 | reg |
| | | 投票 | vote |
| | | 版权 | copyright |
| | | 结果 | result |
| | | 标题 | title |
| | | 按钮 | btn |
| 功能 (function) | .f- | 浮动清除 | clear |
| | | 向左浮动 | fl |
| | | 向右浮动 | fr |
| | | 文本居中 | center |
| | | 文本居右 | right |
| | | 文本居左 | left |
| | | 垂直居中 | middle |

| | | | |
|----|-----|------|----------|
| | | 溢出隐藏 | hidden |
| 状态 | .z- | 选中 | selected |
| | | 当前 | current |
| | | 显示 | show |
| | | 隐藏 | hide |
| | | 打开 | open |
| | | 关闭 | close |
| | | 出错 | err |
| | | 不可用 | disabled |

Rule2：后代选择器不以单个字母出现或以单个字母+”-“为前缀出现。

规则说明：

- 由于使用单个字母的后代选择器的命名会与分类命名重复而导致冲突，因此在使用后代选择器时不得使用单字母。
- 通过使用后代选择器的方法，你不需要考虑他的命名是否已被使用，因为他只在当前模块或元件中生效，同样的样式名可以在不同的模块或元件中重复使用，互不干扰；在多人协作或者分模块协作的时候效果尤为明显！
- 后代选择器不需要完整表现结构树层级，尽量能短则短，但不失语义。

注：后代选择器不要在页面布局中使用，因为污染的可能性较大。

示例：

推荐：

```
.m-list .item {  
  margin: 1px;  
  padding: 1px;  
}
```

不推荐：

```
.g-abc .a {  
  margin: 1px;  
  padding: 1px;  
}
```

二、编码规范

Rule 3: 选择器、属性和值都使用小写

示例：

推荐：

```
.m-list .item {  
  color: #fff;  
}
```

不推荐：

```
.m-list .item {  
  color: #FFFFFF;  
}
```

Rule 4: 省略值为0时的单位

示例：

推荐：

```
.m-list .item {  
  left: 0;  
}
```

不推荐：

```
.m-list .item {  
  padding: 0rem 10rem;  
}
```

Rule 5: 使用单引号

省略url引用中的引号，其他需要引号的地方使用单引号。

示例：

推荐：

```
.m-box {
```

```
background: url(bg.png);
}
.m-box:after {
  content: '.';
}
```

Rule 6: 根据属性的重要性按顺序书写

遵循横向顺序，先书写定位布局类属性，后盒模型等自身属性，最后是文本类及修饰类属性。

| → | 显示属性 | 自身属性 | 文本属性和其他修饰 |
|---|----------|----------|------------|
| | display | width | font |
| | position | margin | text-align |
| | float | border | color |
| | clear | overflow | background |

示例：

推荐：

```
.m-box {
  position: relative;
  width: 600px;
  margin: 0 auto 10px;
  text-align: center;
  color: #000;
}
.m-box {
  position: relative;
  height: 20px;
  line-height: 20px; /* 关联属性推荐写在一起 */
  padding: 5px;
  color: #000;
}
```

Rule 7: 私有在前，标准在后

先写带有浏览器私有标志的，后写W3C标准的。

示例：

推荐：

```
.m-box {
```

```
-webkit-box-shadow: 0 0 0 #000;  
-moz-box-shadow: 0 0 0 #000;  
box-shadow: 0 0 0 #000;  
}
```

Rule 8: 避免耗性能的属性

先写带有浏览器私有标志的，后写W3C标准的。

示例：

不推荐：

```
/* expression */  
.m-panel {  
  width: expression(this.width>100?'100px':'auto');  
}  
/* filter */  
.m-panel {  
  filter: alpha(opacity=50);  
}
```

Rule 9: 尽量减少选择器的长度

不要将选择器写的过于冗长，这会额外增加文件大小并且限制了太小范围的选择器，使树形结构过于严格应用范围过于局限，建议3-4个长度之内写完。

选择器并不需要完整反映结构嵌套顺序，相反，能简则简。

Rule 10: 缩写属性和值

CSS提供了各种缩写属性（如 font 字体）和值（如color颜色）应该尽可能使用，即使在只设置一个值的情况下。

使用缩写属性和值对于代码效率和可读性是有很有用的。

示例：

不推荐：

```
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;
```

```
color: #ffffff;
```

推荐：

```
border-top: 0;  
padding: 0 1em 2em;  
font: 100%/1.6 palatino, georgia, serif;  
color: #fff;
```

Rule 10：属性名结束

属性名的冒号后使用一个空格。出于一致性的原因，属性和值（但属性和冒号之间没有空格）的之间始终使用一个空格。

示例：

不推荐：

```
h3{  
  font-weight:bold;  
}
```

推荐：

```
h3 {  
  font-weight: bold;  
}
```

Rule 11：选择器和声明分离

每个选择器和属性声明总是使用新的一行。

示例：

不推荐：

```
a:focus, a:active {  
  position: relative; top: 1px;  
}
```

推荐：

```
h1,  
h2,
```

```
h3 {  
  font-weight: normal;  
  line-height: 1.2;  
}
```

Rule 12: 规则分隔

规则之间始终有一个空行（双换行符）分隔。

示例：

推荐：

```
html {  
  background: #fff;  
}  
  
body {  
  margin: auto;  
  width: 50%;  
}
```