

CSS规范 - 命名规则

使用类选择器，放弃ID选择器

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。

特殊字符： "-" 连字符

"-"在本规范中并不表示连字符的含义。

她只表示两种含义：分类前缀分隔符、扩展分隔符，详见以下具体规则。

分类的命名方法：使用单个字母+"-"为前缀

布局 (grid) (.g-)；模块 (module) (.m-)；元件 (unit) (.u-)；功能 (function) (.f-)；皮肤 (skin) (.s-)；状态 (.z-)。

对以上的解释详情参见：分类方法中的“CSS内部的分类及其顺序”。

注：

在你样式中的选择器总是要以上面前五类开头，然后在里面使用后代选择器。

如果这五类不能满足你的需求，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。

特殊：j- 将被专用于JS获取节点，请勿使用j-定义样式。

后代选择器命名

- 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：.item 为m-list模块里的每一个项，.text为m-list模块里的文本部分：.m-list .item{}.m-list .text{}。
- 一个语义化的标签也可以是后代选择器，比如：.m-list li{}。
- 不允许单个字母的类选择器出现，原因详见下面的“模块和元件的后代选择器的扩展类”。

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。

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；

```
/* 这里的 .itm 和 .cnt 只在 .m-list 中有效 */  
.m-list {margin:0;padding:0;}  
.m-list .itm {margin:1px;padding:1px;}  
.m-list .cnt {margin-left:100px;}  
/* 这里的 .cnt 和 .num 只在 .m-page 中有效 */  
.m-page {height:20px;}  
.m-page .cnt {text-align:center;}  
.m-page .num {border:1px solid #ddd;}
```

命名应简约而不失语义

```
/* 反对：表现化的或没有语义的命名 */  
.m-abc .green2 {}  
.g-left2 {}  
/* 推荐：使用有语义的简短的命名 */  
.m-list .wrap2 {}  
.g-side2 {}
```

相同语义的不同类命名

方法：直接加数字或字母区分即可（如：.m-list、.m-list2、.m-list3等，都是列表模块，但是是完全不一样的模块）。

其他举例：.f-fw0、.f-fw1、.s-fc0、.s-fc1、.m-logo2、.m-logo3、u-btn、u-btn2等等。

模块和元件的扩展类的命名方法

当A、B、C、...它们类型相同且外形相似区别不大，那么就以它们中出现率最高的做成基类，其他做成基类的扩展。

方法：+ “-” + 数字或字母（如：.m-list的扩展类为.m-list-1、.m-list-2等）。

补充：基类自身可以独立使用（如：即可），扩展类必须基于基类使用（如：）。

如果你的扩展类是表示不同状态，那么你可以这样命名：u-btn-dis，u-btn-hov，m-box-sel，m-box-hov等等，然后像这样使用：。

如果你的网站可以不兼容IE6等浏览器，那么你标识状态的方法也可以采取独立状态分类（.z-）方法：.u-btn.z-dis，.m-box.z-sel，然后像这样使用：。

模块和元件的后代选择器的扩展类

有时候模块内会有些类似的东西，如果你没有把它们做成元件和扩展，那么也可以使用后代选择器和扩展。

后代选择器：`.m-login .btn{}`。

后代选择器扩展：`.m-login .btn-1{}`，`.m-login .btn-dis{}`。

同样也可以采取独立状态分类（.z-）方法：`.m-login .btn.z-dis{}`，然后像这样使用：。

注：此方法用于类选择器，直接使用标签做为选择器的则不需要使用此命名方法。

注：

为防止后代选择器的扩展类和大类命名规范冲突，后代选择器不允许使用单个字母。

比如：`.m-list .a{}`是不允许的，因为当这个.a需要扩展的时候就会变成.a-bb，这样就和大型的命名规范冲突。

分组选择器有时可以代替扩展方法

有时候虽然两个同类型的模块很相似，但是我希望他们之间不要有依赖关系，也就是说你不希望使用扩展的方法，那么你可以通过合并选择器来设置共性的样式。

使用本方法的前提是：相同类型、功能和外观都相似，写在同一片代码区域方便维护。

```
/* 两个元件共性的样式 */
.u-tip1,.u-tip2 {}
.u-tip1 .itm,.u-tip2 .itm {}
/* 在分别是两个元件各自的样式 */
/* tip1 */
.u-tip1 {}
.u-tip1 .itm {}
/* tip2 */
.u-tip2 {}
.u-tip2 .itm {}
```

防止污染和被污染

当模块或元件之间互相嵌套，且使用了相同的标签选择器或其他后代选择器，那么里面的选择器就会被外面相同的选择器所影响。

所以，如果你的模块或元件可能嵌套或被嵌套于其他模块或元件，那么要慎用标签选择器，必要时采用类选择器，并注意命名方式，可以采用`.m-layer .layerxxx`、`.m-list2 .list2xxx`的形式来降低后代选择器的污染性。

