

样的单命名，因为说不定下面的命名就会与第三方 CSS 发生冲突：

```
.header {}      /* 不建议 */
.main {}        /* 不建议 */
.aside {}       /* 不建议 */

.warning {}     /* 不建议 */
.success {}     /* 不建议 */

.red {}         /* 不建议 */
.green {}       /* 不建议 */
```

正确的做法是加一个统一的前缀，使用组合命名的方式。你可以随意命名这个前缀，可以是项目代号的英文缩写，也可以是产品名称的拼音首字母，因为这个前缀的作用是避免冲突，它并不需要任何语义。但需要注意的是前缀最好不要超过 4 个字母，因为字母多了完全没有任何意义，只会徒增 CSS 文件的大小。例如，“CSS 选择器”的英文是 CSS Selector，我就可以取 CSS 的首字母 C 和 Selector 的首字母 S 作为本书所有选择器的前缀类名，于是有：

```
.cs-header {}
.cs-main {}
.cs-aside {}
...
```

如果你认真观察所有的开源 UI 框架，会发现其 CSS 样式一定都有一个一致的前缀，因为这样做会避免发生冲突，我们自己开发项目的时候也要秉承这个理念。

(3) 如果你的项目百分百是自主研发的，以后维护此项目的人也不会盗取别人的 CSS 来充数，则与网站公用结构、颜色相关的这些 CSS 可以使用单命名，例如：

```
.dark { color: #4c5161; }
.red { color: #f4615c; }
.gray { color: #a2a9b6; }
```

但对于非公用内容，如标题 (.title)、盒子 (.box) 等就不能使用单命名，因为颜色这类样式是贯穿于整个项目的，具有高度的一致性，而标题 (.title) 会在很多地方出现，且样式各不相同，如大标题、小标题、弹框标题、模块标题等，容易产生命名冲突。

对于网站 UI 组件，各个业务模块一定要采用多名称的组合命名方式，且最好都有一个统一的命名前缀。

(4) 如果你做的项目并不需要长期维护，也不需要多人合作，例如，只是一些运营活动，请务必添加统一的项目前缀，这都是过来人的忠告，因为这次活动的某些功能和效果日后会被复用，有了统一的前缀，日后直接复制代码就能使用，没有后顾之忧，大家都开心，例如：

```
.cs-title {}
.cs-text {}
.cs-box {}
```

但有一类基于 CSS 属性构建的单命名反而更安全，它们比颜色这些类名还要安全，即使项目会引入外部 CSS：