

# CSS方法论及样式规范

## 一、CSS方法论

方法论是一个哲学术语，会对一系列具体的方法进行分析研究、系统总结并最终提出较为一般性的原则。CSS方法论是一种面向CSS、由个人和组织设计、已被诸多项目检验且公认有效的最佳实践。这些方法论都会涉及结构化的命名约定，并且在组织CSS时可提供相应的指南，从而提升代码的性能、可读性以及可维护性。(引自：[前端躬行记](#))

常见的CSS方法论有：

- OOCSS
- BEM
- SMACSS
- Atomic CSS

我们的实战项目主要采用 BEM + SMACSS 这两种方法论的结合命名约定，所以下面对这两种方法论进行简单介绍。

BEM (Block Element Modifier) 是指块级元素修饰符，BEM分为三部分：

1. 块 (Block) 是一个独立实体，最高级抽象，例如菜单、文本框等。
2. 元素 (Element) 是块的组成部分，被包含在块中，无法自成一体，例如菜单项、标题等。
3. 修饰符 (Modifier) 是块或元素的状态，可更改它们的外观或行为，例如高亮、选中等。

BEM中的块、元素和修饰符需要全部小写，名称中的单词用连字符 (-) 分隔，元素由双下划线 (\_\_) 分隔，修饰符由双连字符 (--) 分隔。注意，块和元素都既不能是HTML元素名或ID，也不依赖其它块或元素。

```
.setting-menu{}
.setting-menu--open{}
.setting-menu__head{}
.setting-menu__head--fixed{}
.setting-menu__content{}
```

上面代码中，`.setting-menu`表示一个独立实体，`.setting-menu__head`、`.setting-menu__content`表示独立实体的组成部分，`.setting-menu--open`是对配置菜单展开状态的一种行为描述，`.setting-menu__head--fixed`是对配置菜单头部固定状态的一种行为描述。

SMACSS (Scalable and Modular Architecture for CSS) 是指可伸缩及模块化的CSS架构，由Jonathan Snook在2011年雅虎工作时提出。他在OOCSS和BEM的基础上添加了五种类别的组件的处理规则，具体如下所列。

1. 基础 (Base) 是为HTML元素定义默认样式，可以包含属性、伪类等选择器。
2. 布局 (Layout) 会将页面分为几部分，可作为高级容器包含一个或多个模块，例如左右分栏、栅格系统等。
3. 模块 (Module) 又名对象或块，是可重用的模块化部分，例如导航栏、产品列表等。
4. 状态 (State) 描述的是任一模块或布局在特定状态下的外观，例如隐藏、激活等。
5. 主题 (Theme) 也就是换肤，描述了页面的外观，它可修改前面四个类别的样式，例如链接颜色、布局方式等。

通过相应的命名前缀来完成对类别的表示，`l-`用作布局的前缀，例如`l-inline`、`l-layout-grid`等；`m-`模块命名前缀，例如`m-profile`、`field`等；`is-`用作状态的前缀，例如`is-collapsed`、`is-active`等；`theme-`用作主题的前缀，例如`theme-a-background`、`theme-l-grid`等。

在实际工作中，不需要局限于某一个CSS方法论，很多时候可以结合使用，共享模块化CSS的规则。例如遵循BEM的命名约定，以及SMACSS的分类前缀，具体如下所列。

```
.m-setting-menu{}
.m-setting-menu--open{}
.m-setting-menu__head{}
.m-setting-menu__head--fixed{}
.m-setting-menu__content{}
```

任何软件设计都存在不完美的地方，CSS方法论也是一样。比如：命名过长不易维护；HTML不够简洁等。但是它带来的巨大优势也是非常明显的，比如：样式不容易冲突，尤其是复杂的样式结构；清晰的CSS架构，对大型CSS项目管理维护非常重要等。

## 二、样式规范

1. 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。
2. 命名应简约而不失语义：反对过长命名和没有语义的命名，如：`.m-navigation`、`.m-abc`，推荐`.m-nav`。
3. 相同语义的不同类命名：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)。

4. 防止污染和被污染：为了防止后代选择器污染，尽量采用 `>` 方式，例如：`-m-nav>li`，而不是`-m-nav li`。
5. 最后一个值也以分号结尾：通常在大括号结束前的值可以省略分号，但是这样做会对修改、添加和维护工作带来不必要的失误和麻烦。
6. 省略值为0时的单位：为节省不必要的字节同时也使阅读方便，我们将`0px`、`0em`、`0%`等值缩写为`0`。
7. 根据属性的重要性按顺序书写：先显示定位布局类属性，后盒模型等自身属性，最后是文本类及修饰类属性。例如：`.m-box{position:relative;width:600px;margin:0 auto 10px;text-align:center;color:#000;}`。
8. CSS需要进行格式化和对齐处理：推荐vsCode插件：`formate: CSS/LESS/SCSS formatter`。