**css style guide new**

[1 前言](#_1 前言)

[2 代码风格](#_2 代码风格)

[2.1 文件](#_2.1 文件)

[2.2 缩进](#_2.2 缩进)

[2.3 空格](#_2.3 空格)

[2.4 换行](#_2.4 换行)

[2.5 选择器](#_2.5 选择器)

[2.6 属性](#_2.6 属性)

[3 通用](#_3 通用)

[3.1 选择器](#_3.1 选择器)

[3.2 属性缩写](#_3.2 属性缩写)

[3.3 属性书写顺序](#_3.3 属性书写顺序)

[3.4 !important](#_3.4 !important)

[3.5 z-index](#_3.5 z-index)

[4 值与单位](#_4 值与单位)

[4.1 文本](#_4.1 文本)

[4.2 数值](#_4.2 数值)

[4.3 url()](#_4.3 url())

[4.4 长度](#_4.4 长度)

[4.5 颜色](#_4.5 颜色)

[4.6 2D位置](#_4.6 2D 位置)

[5 文本编排](#_5 文本编排)

[5.1 字体族](#_5.1 字体族)

[5.2 字号](#_5.2 字号)

[5.3 字体风格](#_5.3 字体风格)

[5.4 字重](#_5.4 字重)

[5.5 行高](#_5.5 行高)

[6 变换与动画](#_6 变换与动画)

[7 响应式](#_7 响应式)

[8 兼容性](#_8 兼容性)

## 1 前言

本文档定义了CSS的编写格式和风格规则。目的是提高代码质量，并使代码更容易维护。

易维护的代码意味着代码具有如下特征：

* 可读性好
* 具有一致性
* 可预见性好
* 看起来好像同一个人编写的
* 有文档

以下文档中，每一条规范都会有一个实施级别，分别是：

* 【强制】： 必须执行
* 【建议】： 推荐不强制使用

## 2 代码风格

### 2.1 文件

【建议】 CSS 文件使用无 BOM 的 UTF-8 编码。

让你的编辑器用没有字节顺序标记的UTF-8编码格式进行编写。

在HTML模板和文件中指定编码 <meta charset="utf-8”>。 不需要制定样式表的编码，它默认为UTF-8.

### 2.2 缩进

**【强制】** 使用 4 个空格做为一个缩进层级，不允许使用 2 个空格 或 tab 字符。

.selector {

margin: 0;

padding: 0;

}

### 2.3 空格

**【强制】** 选择器 与 { 之间必须包含空格。

例如：

.selector {

}

@page {

background-color: #fff;

}

**【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：

margin: 0;

**【强制】** 列表型属性值 书写在单行时，, 后必须跟一个空格。 例如：

font-family: Arial, sans-serif;

### 2.4 换行

**【强制】** 每行不得超过 120 个字符，除非单行不可分割。

**【强制】** 对于超长的样式，在样式值的 空格 处或 , 后换行，建议按逻辑分组。

示例：

/\* 不同属性值按逻辑分组 \*/

background:

transparent url(aVeryVeryVeryLongUrlIsPlacedHere)

no-repeat 0 0;

/\* 可重复多次的属性，每次重复一行 \*/

background-image:

url(aVeryVeryVeryLongUrlIsPlacedHere)

url(anotherVeryVeryVeryLongUrlIsPlacedHere);

/\* 类似函数的属性值可以根据函数调用的缩进进行 \*/

background-image: -webkit-gradient(

linear,

left bottom,

left top,

color-stop(0.04, rgb(88,94,124)),

color-stop(0.52, rgb(115,123,162))

);

### 2.5 选择器

**【强制】** 当一个 rule 包含多个 selector 时，每个选择器声明必须独占一行。

示例：

/\* good \*/

.post,

.page,

.comment {

line-height: 1.5;

}

/\* bad \*/

.post, .page, .comment {

line-height: 1.5;

}

**【强制】** >、+、~ 选择器的两边各保留一个空格。

示例：

/\* good \*/

main > nav {

padding: 10px;

}

label + input {

margin-left: 5px;

}

input:checked ~ button {

background-color: #69C;

}

/\* bad \*/

main>nav {

padding: 10px;

}

label+input {

margin-left: 5px;

}

input:checked~button {

background-color: #69C;

}

**【强制】** 属性选择器中的值必须用双引号包围。

解释：

不允许使用单引号，不允许不使用引号。

示例：

/\* good \*/

article[character="juliet"] {

voice-family: "Vivien Leigh", victoria, female;

}

/\* bad \*/

article[character='juliet'] {

voice-family: "Vivien Leigh", victoria, female;

}

### 2.6 属性

**【强制】** 属性定义必须另起一行。

示例：

/\* good \*/

.selector {

margin: 0;

padding: 0;

}

/\* bad \*/

.selector { margin: 0; padding: 0; }

**【强制】** 属性定义后必须以分号结尾。

示例：

/\* good \*/

.selector {

margin: 0;

}

/\* bad \*/

.selector {

margin: 0

}

## 3 通用

### 3.1 选择器

**【强制】** 如无必要，不得为 id、class 选择器添加类型选择器进行限定。

解释：

在性能和维护性上，都有一定的影响。

示例：

/\* good \*/

#error,

.danger-message {

font-color: #c00;

}

/\* bad \*/

dialog#error,

p.danger-message {

font-color: #c00;

}

示例：

/\* good \*/

.post {

font: 12px/1.5 arial, sans-serif;

}

/\* bad \*/

.post {

font-family: arial, sans-serif;

font-size: 12px;

line-height: 1.5;

}

【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：

/\* good \*/

#username input {}

.comment .avatar {}

/\* bad \*/

.page .header .login #username input {}

.comment div \* {}

### 3.2 属性缩写

【建议】 在可以使用缩写的情况下，尽量使用属性缩写。

示例：

/\* good \*/

.post {

font: 12px/1.5 arial, sans-serif;

}

/\* bad \*/

.post {

font-family: arial, sans-serif;

font-size: 12px;

line-height: 1.5;

}

### 3.3 属性书写顺序

【建议】 同一 rule set 下的属性在书写时，应按功能进行分组，并以 Formatting Model（布局方式、位置） > Box Model（尺寸） > Typographic（文本相关） > Visual（视觉效果） 的顺序书写，以提高代码的可读性。

解释：

* Formatting Model 相关属性包括：position / top / right / bottom / left / float / display / overflow 等
* Box Model 相关属性包括：border / margin / padding / width / height 等
* Typographic 相关属性包括：font / line-height / text-align / word-wrap 等
* Visual 相关属性包括：background / color / transition / list-style 等

另外，如果包含 content 属性，应放在最前面。

示例：

.sidebar {

/\* formatting model: positioning schemes / offsets / z-indexes / display / ... \*/

position: absolute;

top: 50px;

left: 0;

overflow-x: hidden;

/\* box model: sizes / margins / paddings / borders / ... \*/

width: 200px;

padding: 5px;

border: 1px solid #ddd;

/\* typographic: font / aligns / text styles / ... \*/

font-size: 14px;

line-height: 20px;

/\* visual: colors / shadows / gradients / ... \*/

background: #f5f5f5;

color: #333;

-webkit-transition: color 1s;

-moz-transition: color 1s;

transition: color 1s;

}

### 3.4 !important

【建议】 除第三方引用的css外尽量不使用 !important 声明。

!important的作用是来平衡层叠级别，因此，除非用于强制覆盖用户样式，否则，不允许使用!important。

### 3.5 z-index

【建议】 在可控的情况下（不包含三方css），z-index不要设置过大，如，在没有层级是98的情况下，不允许使用99。

对于希望展示在最上层的元素，可以设置z-index为 1000000。

## 4 值与单位

### 4.1 文本

**【强制】** 文本内容必须用双引号包围。

解释：

文本类型的内容可能在选择器、属性值等内容中。

示例：

/\* good \*/

html[lang|="zh"] q:before {

font-family: "Microsoft YaHei", sans-serif;

content: "“";

}

html[lang|="zh"] q:after {

font-family: "Microsoft YaHei", sans-serif;

content: "”";

}

/\* bad \*/

html[lang|=zh] q:before {

font-family: 'Microsoft YaHei', sans-serif;

content: '“';

}

html[lang|=zh] q:after {

font-family: "Microsoft YaHei", sans-serif;

content: "”";

}

### 4.2 数值

**【强制】** 当数值为 0 - 1 之间的小数时，省略整数部分的 0。

示例：

/\* good \*/

panel {

opacity: .8;

}

/\* bad \*/

panel {

opacity: 0.8;

}

### 4.3 url()

**【强制】** url() 函数中的路径不加引号。

示例：

body {

background: url(bg.png);

}

**【强制】** url() 函数中的绝对路径可省去协议名。

示例：

body {

background: url(//baidu.com/img/bg.png) no-repeat 0 0;

}

### 4.4 长度

**【强制】** 长度为 0 时须省略单位。 (也只有长度单位可省)

示例：

/\* good \*/

body {

padding: 0 5px;

}

/\* bad \*/

body {

padding: 0px 5px;

}

### 4.5 颜色

**【强制】** RGB颜色值必须使用十六进制记号形式 #rrggbb。不允许使用 rgb()。

解释：

带有alpha的颜色信息可以使用 rgba()。使用 rgba() 时每个逗号后必须保留一个空格。

示例：

/\* good \*/

.success {

box-shadow: 0 0 2px rgba(0, 128, 0, .3);

border-color: #008000;

}

/\* bad \*/

.success {

box-shadow: 0 0 2px rgba(0,128,0,.3);

border-color: rgb(0, 128, 0);

}

**【强制】** 颜色值可以缩写时，必须使用缩写形式。

示例：

/\* good \*/

.success {

background-color: #aca;

}

/\* bad \*/

.success {

background-color: #aaccaa;

}

**【强制】** 颜色值不允许使用命名色值。

示例：

/\* good \*/

.success {

color: #90ee90;

}

/\* bad \*/

.success {

color: lightgreen;

}

【建议】 颜色值中的英文字符采用小写。如不用小写也需要保证同一项目内保持大小写一致。

示例：

/\* good \*/

.success {

background-color: #aca;

color: #90ee90;

}

/\* good \*/

.success {

background-color: #ACA;

color: #90EE90;

}

/\* bad \*/

.success {

background-color: #ACA;

color: #90ee90;

}

### 4.6 2D 位置

**【强制】** 必须同时给出水平和垂直方向的位置。

解释：

2D 位置初始值为 0% 0%，但在只有一个方向的值时，另一个方向的值会被解析为 center。为避免理解上的困扰，应同时给出两个方向的值。

示例：

/\* good \*/

body {

background-position: center top; /\* 50% 0% \*/

}

/\* bad \*/

body {

background-position: top; /\* 50% 0% \*/

}

## 5 文本编排

### 5.1 字体族

**【强制】** font-family 属性中的字体族名称应使用字体的英文 Family Name，其中如有空格，须放置在引号中。

示例：

h1 {

font-family: "Microsoft YaHei";

}

**【强制】** font-family 按「西文字体在前、中文字体在后」、「效果佳 (质量高/更能满足需求) 的字体在前、效果一般的字体在后」的顺序编写，最后必须指定一个通用字体族( serif / sans-serif )。

示例：

/\* Display according to platform \*/

.article {

font-family: Arial, sans-serif;

}

/\* Specific for most platforms \*/

h1 {

font-family: "Helvetica Neue", Arial, "Hiragino Sans GB", "WenQuanYi Micro Hei", "Microsoft YaHei", sans-serif;

}

**【强制】** font-family 不区分大小写，但在同一个项目中，同样的 Family Name 大小写必须统一。

示例：

/\* good \*/

body {

font-family: Arial, sans-serif;

}

h1 {

font-family: Arial, "Microsoft YaHei", sans-serif;

}

/\* bad \*/

body {

font-family: arial, sans-serif;

}

h1 {

font-family: Arial, "Microsoft YaHei", sans-serif;

}

### 5.2 字号

**【强制】** 需要在 Windows 平台显示的中文内容，其字号应不小于 12px。

解释：

由于 Windows 的字体渲染机制，小于 12px 的文字显示效果极差、难以辨认。

### 5.3 字体风格

【建议】 需要在 Windows 平台显示的中文内容，不要使用除 normal 外的 font-style。其他平台也应慎用。

解释：

由于中文字体没有 italic 风格的实现，所有浏览器下都会 fallback 到 obilique 实现 (自动拟合为斜体)，小字号下 (特别是 Windows 下会在小字号下使用点阵字体的情况下) 显示效果差，造成阅读困难。

### 5.4 字重

**【强制】** font-weight 属性必须使用数值方式描述。

解释：

CSS 的字重分 100 – 900 共九档，但目前受字体本身质量和浏览器的限制，实际上支持 400 和 700 两档，分别等价于关键词normal 和 bold。 浏览器本身使用一系列启发式规则来进行匹配，在 =700 时匹配 Bold 字重。

示例：

/\* good \*/

h1 {

font-weight: 700;

}

/\* bad \*/

h1 {

font-weight: bold;

}

### 5.5 行高

【建议】 line-height 在定义文本段落时，应使用数值。

解释： 将 line-height 设置为数值，浏览器会基于当前元素设置的 font-size 进行再次计算。在不同字号的文本段落组合中，能达到较为舒适的行间间隔效果，避免在每个设置了 font-size 都需要设置 line-height。 当 line-height 用于控制垂直居中时，还是应该设置成与容器高度一致。

示例：

.container {

line-height: 1.5;

}

## 6 变换与动画

**【强制】** 使用 transition 时应指定 transition-property。

示例：

/\* good \*/

.box {

transition: color 1s, border-color 1s;

}

/\* bad \*/

.box {

transition: all 1s;

}

## 7 响应式

**【强制】** Media Query 不得单独编排，必须与相关的规则一起定义。

示例：

/\* Good \*/

/\* header styles \*/

@media (...) {

/\* header styles \*/

}

/\* main styles \*/

@media (...) {

/\* main styles \*/

}

/\* footer styles \*/

@media (...) {

/\* footer styles \*/

}

/\* Bad \*/

/\* header styles \*/

/\* main styles \*/

/\* footer styles \*/

@media (...) {

/\* header styles \*/

/\* main styles \*/

/\* footer styles \*/

}

**【强制】** Media Query 如果有多个逗号分隔的条件时，应将每个条件放在单独一行中。

示例：

@media

(-webkit-min-device-pixel-ratio: 2), /\* Webkit-based browsers \*/

(min--moz-device-pixel-ratio: 2),/\* Older Firefox browsers (prior to Firefox 16) \*/

(min-resolution: 2dppx), /\* The standard way \*/

(min-resolution: 192dpi) { /\* dppx fallback \*/

/\* Retina-specific stuff here \*/

}

## 8 兼容性

**【强制】** 禁止使用 Expression。