**智能中心开发指南与规范**

**(CSS代码规范篇)**

**V0.1**

2018年6月

**上海电气风电集团有限公司**

目录

文档类别使用对象 3

**1** **引言** 4

1.1 目的 4

1.2 范围 4

1.3 术语定义 4

1.4 参考资料 4

1.5 版序控制记录 4

1.6 版本更新记录 5

2 CSS代码规范 6

.格式化代码 6

1.1文件 6

1.2缩进 6

1.3空格 6

1.4行长度 6

1.5选择器 7

2.选择器与属性缩写 8

2.1选择器 8

2.2 属性缩写 9

2.3 每个规则集之间保留一个空行 10

3.值与单位 11

3.1 文本 11

3.2 数值 11

4.3 url() 12

3.3 长度 12

3.4 颜色 12

3.5 2D 位置 14

4.文本编排 14

4.1 字体族 14

4.2 字号 16

4.3 字体风格 16

6 变换与动画 16

7 响应式 17

CSS注释 18

普通注释 18

区块注释 18

CSS命名规范 18

命名组成 18

命名前缀 18

命名单词 19

编辑器配置 19

文档类别使用对象

*文档类别*

该文档是为上海电气风电集团智能中心软件开发代码与指南的规范性文件。(Note.JS部分)

*使用对象*

该文档使用对象为智能中心软件的设计、开发和外包软件开发人员以及智能中心的部门经理等管理人员。

## **引言**

## 目的

本文档是为规范公司智能中心软件开发设计、编码而制定的。

## 范围

本文档为智能中心软件开发人员、产品经理以及外包人员提供有关软件开发设计规范的相关内容，包括：

* JavaScript代码规范

## 术语定义

## 参考资料

## 版序控制记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版序状态 | 拟稿 | 审核 | 批准 | 发布日期 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## 版本更新记录

\***A** - 增加 **M** - 修改 **D** - 删除

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本/修订版 | 修改页码 | 修改记录 | 修改人 | 日期 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## CSS代码规范

## .格式化代码

### 1.1文件

#### [建议]：CSS文件使用无BOM的UTF-8编码

### 1.2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;

}

### 1.3空格

#### [强制]：选择器 与 { 之间必须包含空格。

.selector {

}

#### [强制]：选择器 与 { 之间必须包含空格。

margin: 0;

#### [强制]：列表性属性书在单行时，,后必须跟一个空格

font-family: Aria, sans-serif;

### 1.4行长度

#### [强制]： 每行不得超过 120 个字符，除非单行不可分割。

[建议]： 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。

/\* 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))

);

### 1.5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。

/\* 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。

css

/\* good \*/

article[character="juliet"] {

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

}

/\* bad \*/

article[character='juliet'] {

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

}

## 2.选择器与属性缩写

### 2.1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：

css

/\* good \*/

#error,

.danger-message {

font-color: #c00;

}

/\* bad \*/

dialog#error,

p.danger-message {

font-color: #c00;

}

#### [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 \* {}

### 2.2 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;

}

#### [建议] 使用 border / margin / padding 等缩写时，应注意隐含值对实际数值的影响，确实需要设置多个方向的值时才使用缩写。

解释：border / margin / padding 等缩写会同时设置多个属性的值，容易覆盖不需要覆盖的设定。如某些方向需要继承其他声明的值，则应该分开设置。

示例：

/\* centering <article class="page"> horizontally and highlight featured ones \*/

article {

margin: 5px;

border: 1px solid #999;

}

/\* good \*/

.page {

margin-right: auto;

margin-left: auto;

}

.featured {

border-color: #69c;

}

/\* bad \*/

.page {

margin: 5px auto; /\* introducing redundancy \*/

}

.featured {

border: 1px solid #69c; /\* introducing redundancy \*/

}

### 2.3 每个规则集之间保留一个空行

/\* good \*/

.selector1 {

display: block;

width: 100px;

}

.selector2 {

padding: 10px;

margin: 10px auto;

}

/\* bad \*/

.selector1 {

display: block;

width: 100px;

}

.selector2 {

padding: 10px;

margin: 10px auto;

}

## 3.值与单位

### 3.1 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: "”";

}

### 3.2 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);

}

### 3.3 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;

}

### 3.4 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()。

/\* 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;

}

### 3.5 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。[background-position属性值的定义](http://www.w3.org/TR/CSS21/colors.html#propdef-background-position)

示例：

/\* good \*/

body {

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

}

/\* bad \*/

body {

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

}

## 4.文本编排

### 4.1 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。

解释：  
所谓英文 Family Name，为字体文件的一个元数据，常见名称如下：

| **字体** | **操作系统** | **Family Name** |
| --- | --- | --- |
| 宋体 (中易宋体) | Windows | SimSun |
| 黑体 (中易黑体) | Windows | SimHei |
| 微软雅黑 | Windows | Microsoft YaHei |
| 微软正黑 | Windows | Microsoft JhengHei |
| 华文黑体 | Mac/iOS | STHeiti |
| 冬青黑体 | Mac/iOS | Hiragino Sans GB |
| 文泉驿正黑 | Linux | WenQuanYi Zen Hei |
| 文泉驿微米黑 | Linux | WenQuanYi Micro Hei |

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 )

解释：  
更详细说明可参考[本文](http://www.zhihu.com/question/19911793/answer/13329819)。

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;

}

### 4.2 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。

### 4.3 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。

## 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;

}

#### [建议] 尽可能在浏览器能高效实现的属性上添加过渡和动画。

解释：  
见[本文](http://www.html5rocks.com/en/tutorials/speed/high-performance-animations/" \t "_blank)，在可能的情况下应选择这样四种变换：

* transform: translate(npx, npx);
* transform: scale(n);
* transform: rotate(ndeg);
* opacity: 0..1;

典型的，可以使用 translate 来代替 left 作为动画属性。  
示例：

/\* good \*/

.box {

transition: transform 1s;

}

.box:hover {

transform: translate(20px); /\* move right for 20px \*/

}

/\* bad \*/

.box {

left: 0;

transition: left 1s;

}

.box:hover {

left: 20px; /\* move right for 20px \*/

}

## 7 响应式

#### [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 \*/

}

# CSS注释

### 普通注释

/\* 普通注释 \*/

每个class前必须加注释。

### 区块注释

/\*\*

\* 模块：m-detail

\* author: xxx

\* edit: 2016.5.02

\*/

### 文件头注释

/\*\*

\* 模块：m-detail

\* 履历: xxxxx 2016.5.02

\*/

# CSS命名规范

### 命名组成

* 命名必须由单词，中划线组成。例如:.info,.news-list
* 不推荐使用拼音来作为样式名，尤其是缩写的拼音、拼音与英文的混合
* 所有命名都使用小写,使用中划线 “-” 作为连接字符，而不是下划线 “\_“

### 命名前缀

| **前缀** | **说明** | **示例** |
| --- | --- | --- |
| g- | 全局通用样式命名 | g-mod |
| m- | 模块命名方式 | m-detail |
| ui- | 组件命名方式 | ui-selector |
| j- | 所有用于纯交互的命名，不涉及任何样式规则。 | J-switch |

不允许出现以类似：.info, .current, .news 开头的选择器，比如：

.info{sRules;}   
因为这样将给我们带来不可预知的管理麻烦以及沉重的历史包袱。你永远也不会知道哪些样式名已经被用掉了，如果你是一个新人，你可能会遭遇，你每定义个样式名，都有同名的样式已存在，然后你只能是换样式名或者覆盖规则。所以我们推荐这样写：

.m-xxx .info{sRules;}  
所有的选择器必须是以 g-, m-, ui- 等有前缀的选择符开头的，意思就是说所有的规则都必须在某个相对的作用域下才生效，尽可能减少全局污染。 J- 这种级别的className完全交由JSer自定义，但是命名的规则也可以保持跟重构一致，比如说不能使用拼音之类的

### 命名单词

* 不以表现来命名，而是根据内容来命名。比如：left, right, center, red, black这种以表现来定命名，不允许出现；
* 推荐使用功能和内容相关词汇的命名，如:

套系:package

相册:photo-album

作品:works

攻略:raiders

普通用户:normal-user

达人:talent-user

摄影师:photographer

用户昵称:user-alias

头像:head

地区:area

关注数:follow

粉丝数:followers

互相注意:attention

标签:label

发表时间:publish-date,publish-time

标题:title

信息:info

内容:content

关于我:about

简介内容:intro-content

评论:review

服务:service

封面:cover

流行:popular

收藏:collect

查看:view

预约:reservation

促销:sale-promotion

(待补充)

# 编辑器配置

将你的编辑器按照下面的配置进行设置，以避免常见的代码不一致和差异：

* 使用4个空格缩进
* 使用Unix风格换行符(LF)
* 保存文件时，删除尾部的空白符。
* 设置文件编码为 UTF-8。
* 在文件结尾添加一个空白行。

参照文档并将这些配置信息添加到项目的 .editorconfig文件中。

// 示例

# css-creating coding style

root = true

# 为所有文件设置风格

[\*]

charset = utf-8

indent\_style = space

indent\_size = 4

end\_of\_line = lf

trim\_trailing\_whitespace = true

insert\_final\_newline = true

# 为 Markdown 文件保留行尾空格

[\*.md]

trim\_trailing\_whitespace = false