

CSS 编码规范

对当前规范有更好的建议，请到 [issue](http://gitlab.alibaba-inc.com/specs/style-guide/issues) (<http://gitlab.alibaba-inc.com/specs/style-guide/issues>) 页面反馈给我们，谢谢！

基本规范

- 缩进使用两个空格

```
/* bad */
.mod-example {
    padding-left: 15px;
}

/* good */
.mod-example {
  padding-left: 15px;
}
```

- 在每个声明块的左花括号前添加一个空格

```
/* bad */
.mod-example{
  padding-left: 15px;
}

/* good */
.mod-example {
  padding-left: 15px;
}
```

- 声明块的右花括号应当单独成行

```
/* bad */
.mod-example {
  padding-left: 15px;}

/* good */
.mod-example {
  padding-left: 15px;
}
```

- 每条声明语句的 `:` 后应该插入一个空格，前面无空格

```
/* bad */
.mod-example {
  padding-left:15px;
}

/* good */
.mod-example {
  padding-left: 15px;
}
```

- 所有声明语句都以分号结尾，不能省略不写

```
/* bad */
.mod-example {
  padding-left: 15px
}

/* good */
.mod-example {
  padding-left: 15px;
}
```

选择器规范

- 为选择器分组时，将单独的选择器单独放在一行

```
/* bad */
.selector, .selector-secondary {
  padding-left: 15px;
}

/* good */
.selector,
.selector-secondary {
  padding-left: 15px;
}
```

- 为选择器中的属性添加双引号

```
/* bad */
.selector[type=text] {
  padding-left: 15px;
}

/* good */
.selector[type="text"] {
  padding-left: 15px;
}
```

- 建议选择器层级不要超过5级

```
/* bad */
.main .top .left .mod-a .content .detail {
  padding-left: 15px;
}

/* good */
.mod-a .content .detail {
  padding-left: 15px;
}
```

属性规范

属性顺序

建议相关的属性说明放在一组，并按照下面的顺序排列：

1. 定位 (position、left、right、top、bottom、z-index)
2. 盒子模型 (display、float、width、height、margin、padding、border、border-radius)
3. 排印 (font、color、background、line-height、text-align)

由于定位可以从正常的文档流中移除元素，并且还能覆盖盒子模型相关的样式，因此排在首位。而盒子模型决定了组件的尺寸和位置，所以排第二位。排印只是影响元素的细节样式变化，所以放第三位。

```
/* bad */
.mod-example {
  font: normal 13px "Helvetica Neue", sans-serif;
  display: block;
  position: absolute;
  z-index: 100;
  width: 100px;
  height: 100px;
  border: 1px solid #ccc;
  top: 0;
  bottom: 0;
  left: 0;
  right: 0;
  line-height: 1.5;
  text-align: center;
}

/* good */
.mod-example {
  /* 定位 */
  position: absolute;
  top: 0;
  right: 0;
  bottom: 0;
  left: 0;
  z-index: 100;
  /* 盒模型 */
  display: block;
  float: right;
  width: 100px;
  height: 100px;
  margin: 15px auto;
  padding: 10px 15px;
  border: 1px solid #ccc;
  /* 排印 */
  font: normal 13px "Helvetica Neue", sans-serif;
  line-height: 1.5;
  color: #333;
  background-color: #f5f5f5;
  text-align: center;
}
```

简写形式的属性声明

- 对于 background 和 font 这两个简写形式的属性声明，要么就显式声明所有的值，要么就不要使用简写形式。

```
/* bad */
.mod-example {
  font: normal 13px;
  background: red;
  background: url(image.jpg);
}

/* good */
.element {
  font-style: normal;
  font-size: 13px;
  background-color: red;
  background-image: url(image.jpg);
}
```

0 和 单位

- 省略“0”值后面的单位。不要在 0 值后面使用单位，除非有值。

```
/* bad */
.mod-example {
  padding-left: 0px;
}

/* good */
.mod-example {
  padding-left: 0;
}
```

颜色值十六进制表示法

- 在可能的情况下，使用 3 个字符的十六进制表示法，并始终使用小写的十六进制数字

```
/* bad */
.mod-example {
  color: #cccccc;
  background-color: #efefef;
}

/* good */
.mod-example {
  color: #ccc;
  background-color: #efefef;
}
```

- 应避免16进制表示法与rgb表示法混用的情况，并优先使用16进制表示法

```
/* bad */
.example-part1 {
  color: #efefef;
}
.example-part2 {
  color: rgb(252, 252, 252);
}

/* good */
.example-part1 {
  color: #efefef;
}
.example-part2 {
  color: #fcfcfc;
}
```

小数

- 对于使用到小数的情况，省略前边的 0

```
/* bad */
.mod-example {
  opacity: 0.5;
}

/* good */
.mod-example {
  opacity: .5;
}
```

引号

- 属性选择器或属性值用双引号 `""` 括起来，而 URI 值 `url()` 不要使用任何引号

```
/* bad */
body {
  font-family: open sans, arial, sans-serif;
  background-image: url('http://taobao.com/');
}

/* good */
body {
  font-family: "open sans", arial, sans-serif;
  background-image: url(http://taobao.com/);
}
```

自定义 font-family

- 对于自定义的 font-family 命名，必须使用业务域名前缀作为名字的开始，例如淘宝爱逛街的自定义字体：


```
/* bad */
@font-face {
  /* 爱逛街业务自定义字体 */
  font-family: icon-font;
  src: url(//at.alicdn.com/t/font_1426561436_1444210)
}

@font-face {
  /* 爱逛街业务自定义字体 */
  font-family: guang-iconfont;
  src: url(//at.alicdn.com/t/font_1426561436_1444210)
}
```

媒体查询（Media query）规范

- 媒体查询建议根据需要采用下面两种组织形式：
 - 将媒体查询放在尽可能相关规则的附近，不要放在文档底部，否则容易被后来维护的人遗忘
 - 媒体查询针对每一个种屏幕（大、中、小）的分别单独组织为一个文件

Example1:

```
.element {
}

.element-avatar {
}

.element-selected {
}

@media (min-width: 480px) {
  .element {
  }
  .element-avatar {
  }
  .element-selected {
  }
}
```

Example2:

```
/* base.css */

.element {
}

.element-avatar {
}

.element-selected {
}
```

```
/* base-media-small.css */

@media (min-width: 480px) {
  .element {
  }
  .element-avatar {
  }
  .element-selected {
  }
}
```

注释规范

代码注释

- 代码是由人来编写和维护的。保证你的代码是描述性的，包含好的注释，并且容易被他人理解。好的代码注释传达上下文和目标。不要简单地重申组件或者 `class` 名称。

```
/* bad */

/* Modal header */
.modal-header {
}

/* good */

/* Wrapping element for .modal-title and .modal-close */
.modal-header {
}
```

文件注释

- 文件注释，即声明在文件头部，描述文件的元信息，表明这个文件的作用是什么，如下例子：

```
/**
 * 这个文件的作用是什么，巴拉巴拉
 */
body {
  color: red;
}
```

命名规范

- 小写字母加连字符（不是下划线，也不是驼峰命名法）

```
/* bad */
.mod_example {
  padding-left: 15px;
}
.modExample {
  padding-left: 15px;
}

/* good */
.mod-example {
  padding-left: 15px;
}
```

- 需要在 javascript 中使用的类名以 J_ 开头，接“大驼峰”命名。例如 J_ExampleClass，并且这类的 class 不能出现在 CSS 文件中

```
<!-- Bad Html Class for Javascript Hook -->
<div class="mod-example"></div>

<!-- Good Html Class for Javascript Hook -->
<div class="J_ExampleClass">Just a Example</div>
```

参考资料

- Code Guide by @mdo (<http://mdo.github.io/code-guide/>)