命名规则

1. **项目命名**

全部采用小写方式， 以下划线分隔。

例：my\_project\_name

1. **目录命名**

参照项目命名规则；

有复数结构时，要采用复数命名法。

例：scripts, styles, images, data\_models

1. **JS文件命名**

参照项目命名规则。

例：account\_model.js

1. **CSS文件命名**

参照项目命名规则。

例：retina\_sprites.scss

1. **HTML文件命名**

参照项目命名规则。

例：error\_report.html

1. **HTML语法**

缩进使用soft tab（4个空格）；

嵌套的节点应该缩进；

在属性上，使用双引号，不要使用单引号；

属性名全小写，用中划线做分隔符；

不要在自动闭合标签结尾处使用斜线；

不要忽略可选的关闭标签，例：</li> 和 </body>。

<!DOCTYPE html><html>

<head>

<title>Page title</title>

</head>

<body>

<img src="images/company\_logo.png" alt="Company">

<h1 class="hello-world">Hello, world!</h1>

</body></html>

1. **HTML5 doctype**

在页面开头使用这个简单地doctype来启用标准模式，使其在每个浏览器中尽可能一致的展现；

doctype大写

<!DOCTYPE html><html>

...</html>

lang属性

html标签上加上lang属性。

<!DOCTYPE html><html lang="zh-cn">

...</html>

1. **字符编码**

指定为'UTF-8'。

<!DOCTYPE html><html>

<head>

<meta charset="UTF-8">

</head>

...</html>

1. **IE兼容模式**

用 <meta> 标签可以指定页面应该用什么版本的IE来渲染；

<!DOCTYPE html><html>

<head>

<meta http-equiv="X-UA-Compatible" content="IE=Edge">

</head>

...</html>

1. **引入CSS, JS**

在引入CSS和JS时不需要指明 type，因为 text/css 和text/javascript 分别是他们的默认值。

1. **属性顺序**

属性应该按照特定的顺序出现以保证易读性；

class, id, name, data-\*

src, for, type, href, value , max-length, max, min, pattern

placeholder, title, alt

aria-\*, role

required, readonly, disabled

class是为高可复用组件设计的，所以应处在第一位；

id更加具体且应该尽量少使用，所以将它放在第二位。

<a class="..." id="..." data-modal="toggle" href="#">Example link</a>

<input class="form-control" type="text">

<img src="..." alt="...">

1. **JS生成标签**

在JS文件中生成标签让内容变得更难查找，更难编辑，性能更差。应该尽量避免这种情况的出现。

减少标签数量

在编写HTML代码时，需要尽量避免多余的父节点；

1. **CSS缩进**

使用soft tab（4个空格）。

每个属性声明末尾都要加分号。

'}'后最好跟一个空行

1. **CSS注释**

注释统一用'/\* \*/'；

1. **CSS命名**

类名使用小写字母，以中划线分隔

id采用驼峰式命名

/\* class \*/

.element-content {

...}

/\* id \*/

#myDialog {

...}

1. **CSS颜色**

颜色16进制用小写字母；

颜色16进制尽量用简写。

1. **CSS属性简写**

常见的属性简写包括：

font

background

transition

animation

1. **CSS媒体查询**

尽量将媒体查询的规则靠近与他们相关的规则，不要将他们一起放到一个独立的样式文件中，或者丢在文档的最底部，这样做只会让大家以后更容易忘记他们。

.element {

...}

.element-avatar{

...}

@media (min-width: 480px) {

.element {

...

}

.element-avatar {

...

}}

1. **CSS杂项**

不允许有空的规则；

元素选择器用小写字母；

去掉小数点前面的0；

去掉数字中不必要的小数点和末尾的0；

属性值'0'后面不要加单位；

同个属性不同前缀的写法需要在垂直方向保持对齐，具体参照右边的写法；

无前缀的标准属性应该写在有前缀的属性后面；

不要在同个规则里出现重复的属性，如果重复的属性是连续的则没关系；

不要在一个文件里出现两个相同的规则；

用 border: 0; 代替 border: none;；

1. **JavaScript**

缩进使用soft tab（4个空格）。

单行长度不要超过80。

1. **以下几种情况后需加分号：**

变量声明

表达式

return

throw

break

continue

do-while

1. **空行**

以下几种情况需要空行：

变量声明后（当变量声明在代码块的最后一行时，则无需空行）

注释前（当注释在代码块的第一行时，则无需空行）

代码块后（在函数调用、数组、对象中则无需空行）

文件最后保留一个空行

1. **换行**

换行的地方，行末必须有','或者运算符；

以下几种情况不需要换行：

下列关键字后：else, catch, finally

代码块'{'前

以下几种情况需要换行：

代码块'{'后和'}'前

变量赋值后

1. **单行注释**

双斜线后，必须跟一个空格；

缩进与下一行代码保持一致；

可位于一个代码行的末尾，与代码间隔一个空格。

1. **多行注释**

最少三行, '\*'后跟一个空格，具体参照右边的写法；

建议在以下情况下使用：

难于理解的代码段

可能存在错误的代码段

浏览器特殊的HACK代码

业务逻辑强相关的代码

1. **文档注释**

在以下情况下使用：

所有常量

所有函数

所有类

1. **引号**

最外层统一使用单引号。

1. **变量命名**

标准变量采用驼峰式命名

'ID'在变量名中全大写

'URL'在变量名中全大写

'Android'在变量名中大写第一个字母

'iOS'在变量名中小写第一个，大写后两个字母

常量全大写，用下划线连接

构造函数，大写第一个字母

jquery对象必须以'$'开头命名

var thisIsMyName;

var goodID;

var reportURL;

var AndroidVersion;

var iOSVersion;

var MAX\_COUNT = 10;

function Person(name) {

this.name = name;}

var $body = $('body');

1. **变量声明**

一个函数作用域中所有的变量声明尽量提到函数首部，用一个var声明，不允许出现两个连续的var声明。

1. **函数**

无论是函数声明还是函数表达式，'('前不要空格，但'{'前一定要有空格；

函数调用括号前不需要空格；

立即执行函数外必须包一层括号；

参数之间用', '分隔，注意逗号后有一个空格。

1. **数组、对象**

对象属性名不需要加引号；

对象以缩进的形式书写，不要写在一行；

数组、对象最后不要有逗号。

变量不要先使用后声明；

不要使用未声明的变量

不要声明了变量却不使用；

不要在应该做比较的地方做赋值；

debugger不要出现在提交的代码里；

数组中不要存在空元素；

不要在循环内部声明函数；

1. **杂项**

不要混用tab和space；

不要在一处使用多个tab或space；

对上下文this的引用只能使用'\_this', 'that', 'self'其中一个来命名；

行尾不要有空白字符；

不允许有空的代码块。