豆瓣Javascript代码风格规范

**Douban Javascript Core Style Guideline**

**1. Javascript代码应符合Douban-JSLint检验标准**

1-1. 语句必须都有分号结尾，除了for, function, if, switch, try, while

1-2. 只有长语句可以考虑断行，如：

                                 TEMPL\_SONGLIST.replace('{TABLE}', da['results'])  
                                               .replace('{PREV\_NUM}', prev)  
                                               .replace('{NEXT\_NUM}', next)  
                                               .replace('{CURRENT\_NUM}', current)  
                                               .replace('{TOTAL\_NUM}', da.page\_total);

为了避免和JSLint的检验机制冲突，“.”或“＋”这类操作符放在行尾，上面代码应改为：

                                 TEMPL\_SONGLIST.replace('{TABLE}', da['results']).  
                                               replace('{PREV\_NUM}', prev).  
                                               replace('{NEXT\_NUM}', next).  
                                               replace('{CURRENT\_NUM}', current).  
                                               replace('{TOTAL\_NUM}', da.page\_total);

1-3. 避免额外的逗号。如：var arr = [1,2,3,];

1-4. 所有的循环体和判断体都需要用"{}"括起来。如：

错：

                                 if (condition)  
                                     statement;  
                                 或  
                                  if (condition) statement;

对:

                                 if (condition) {  
                                     statement;  
                                 }  
                                 或  
                                 if (condition) { statement; }

1-5. for-in循环体中必须用hasOwnProperty方法检查成员是否为自身成员。避免来自原型链上的污染。

1-6. 变量声明。变量声明应放在function的最上面。避免使用未声明的变量。

错：

                                 if (n > 0) {  
                                   var isvalid = true;  
                                 }

对：

                                 var isvalid;  
                                 if (n > 0) {  
                                   isvalid = true;  
                                 }

1-7. 不要使用with, void, eval。

1-8. 使用严格的条件判断符。用===代替==，用!==代替!=。

1-9. 下面类型的对象不建议用new构造：new Number, new String, new Boolean, new Object(用{}代替), new Array(用[]代替)。

1-10. 引用对象成员用obj.prop1代替obj[“prop1”]，除非属性名是变量。

注：Douban-JSLint是定制过的JSLint

注：如果模块代码中，使用其它全局变量想跳过JSLint的检查，可以在该文件中加入/\*global\*/声明，如： /\*global alert: true, console: true, top: true, setTimeout: true \*/

**2. Javascript命名规则**

2-1. 构造器的首字母大写。如：

                         function Dialog (config) {  
                           statement;  
                         }  
  
                         var dlg = new Dialog({...});

2-2. 对象的属性或方法名采用小驼峰式(lower camel-case)，如"init", "bindEvent", "updatePosition"：

                         Dialog.prototype = {  
                           init: function () {},  
                           bindEvent: function () {},  
                           updatePosition: function () {}  
                         ...  
                         };

2-3. 私有变量名用下划线开头。如："\_current", "\_defaultConfig"

2-4. 常量名全部大写，单词间用下划线分隔。如：“CSS\_BTN\_CLOSE”, "TXT\_LOADING"

2-5. 变量名的前缀：

|  |  |  |
| --- | --- | --- |
| Prefix | Element | Example |
| n | integer | nVariableName |
| i,j,k,m,n, etc. \* | integer as counter/iterator | (for i=0; i<=oArray.length; i++) |
| s | string | sVariableName |
| o | object | oObjectName |
| is, can, has | boolean | [Boolean name]ConditionName |
| event method | event attachment | [event type]\_MethodName |
| get | accessor method | getMethodName |
| set | accessor method | setMethodName |

Note: Only a counter/iterator should use a single-letter designation.

**3. 代码格式化要求**

3-1. 语句中的必要空格和缩进

3-1-1. 用来包含语句的"()"前后需要跟空格，诸如： if / for / while / switch ( statements ) { … } 等

3-1-2. "="前后需要跟空格

3-1-3. 数组成员间的","后面需要跟空格

不好：

             for (t in selected) { if (!hash[t]) deselect(t) }

好：

             for ( t in selected ) {   
               if ( !hash[t] ) {  
                deselect(t);   
               }  
             }

3-2. 长语句采用断行:

不好：

             TEMPL\_SONGLIST.replace('{TABLE}', da['results']).replace('{PREV\_NUM}', prev).replace('{NEXT\_NUM}', next).replace('{CURRENT\_NUM}', current).replace('{TOTAL\_NUM}', da.page\_total);

好：

             TEMPL\_SONGLIST.replace('{TABLE}', da['results']).  
                           replace('{PREV\_NUM}', prev).  
                           replace('{NEXT\_NUM}', next).  
                           replace('{CURRENT\_NUM}', current).  
                           replace('{TOTAL\_NUM}', da.page\_total);

3-3. 格式化对象参数：

不好：

              embedSWF(id, { url: '/swf/player30792.swf?url=' + el.href, width: 261, height: 30, params: { wmode:'transparent' }, attributes: { id: "player-sample" + i, name: "player-sample" + i }});

好：

              embedSWF(id, {  
                 url: '/swf/player30792.swf?url=' + el.href,   
                 width: 261,  
                 height: 30,  
                 params: { wmode:'transparent' },  
                 attributes: {  
                   id: "player-sample" + i,  
                   name: "player-sample" + i  
                 }  
               });

**相关工具**

[Douban-JSLint](https://www.google.com/url?q=http://www.google.com/url?q%3Dhttps%253A%252F%252Fgithub.com%252Fkejun%252FDouban-JSLint%26sa%3DD%26sntz%3D1%26usg%3DAFQjCNHHzDJeKiUjkNsUO3fwS-ciZoGUeA&sa=D&ust=1520244780522000&usg=AFQjCNEi8I6vErlTfKR6xM9q9B4hy-LhFw)

[Closure Linter](https://www.google.com/url?q=http://code.google.com/closure/&sa=D&ust=1520244780523000&usg=AFQjCNELX8dT4xZxw6UKli6girQ-ujlSig)

**相关推荐**

[Douglas Crockford: Code Conventions for the JavaScript Programming Language](https://www.google.com/url?q=http://www.google.com/url?q%3Dhttp%253A%252F%252Fjavascript.crockford.com%252Fcode.html%26sa%3DD%26sntz%3D1%26usg%3DAFQjCNHNaLUgg7CIaeAUyX2tnmwE99uRSQ&sa=D&ust=1520244780523000&usg=AFQjCNFvwkk96qkPdWchZrqOh4saH8dtww)

[Google JavaScript Style Guide](https://www.google.com/url?q=http://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml&sa=D&ust=1520244780524000&usg=AFQjCNFHmIhs0Q8uvVdfbNd0lpP9j7ZkvQ)

[JQuery Core Style Guidelines](https://www.google.com/url?q=http://www.google.com/url?q%3Dhttp%253A%252F%252Fdocs.jquery.com%252FJQuery_Core_Style_Guidelines%26sa%3DD%26sntz%3D1%26usg%3DAFQjCNFSg86bKzfIRRtO6HhDPrOEqEtN1g&sa=D&ust=1520244780524000&usg=AFQjCNEQcoftGHfMILYE5oD3wB5ASaUtzA)

更新记录：

2011/2/25 - 添加Douban-JSLint

2010/11/17 - 发布

豆瓣CSS开发规范

# Douban CSS Code Guideline

## 1. CSS浏览器支持标准

|  |  |  |  |
| --- | --- | --- | --- |
|  | WinXP | Win7 | OS X |
| IE9 | B | B |  |
| IE8 | A | A |  |
| IE7 | B | B |  |
| IE6 | B | B |  |
| Chrome16 | A | A | A |
| Chrome14 | A | A | A |
| Firefox10 | A | A | A |
| Firefox9 | A | A | A |
| Safari | B | B | B |
| Opera | C | C | C |

（注：根据2012年4月数据整理）

* A级－交互和视觉完全符全设计的要求
* B级－视觉上允许有所差异，但不破坏页面的整体效果
* C级－可忽略设计上的细节，但不防碍使用

## 2. 重用已有的样式库。

* 2-1. 主站全局CSS文件 - douban.css (注意：新项目中不推荐使用)
* 2-2. 兼容主站的全局CSS文件 - /css/core/\_init\_.css
* 2-3. 当前使用的CSS库（持续更新）
* util-01 reset /css/core/reset.css
* util-02 通用模块容器 /css/core/mod.css
* ui-01. 喜欢按钮 /css/core/fav\_btn.css
* ui-02. 视频/相册列表项 /css/core/media\_item.css
* ui-03. 评星 /css/core/rating.css
* ui-04. 通用按钮 /css/core/common\_button.css
* ui-05. 分页 /css/core/pagination.css
* ui-06. 推荐按钮 /css/core/rec\_btn.css
* ui-07. 老版对话框 /css/core/old\_dialog.css
* ui-08. 老版Tab /css/core/old\_tab.css
* ui-09. 老版成员列表 /css/core/old\_userlist.css
* ui-10. 老版信息区 /css/core/notify.css
* ui-11. 社区用户导航 /css/core/profile\_nav.css
* ui-12. 当前大社区导航 /css/core/site\_nav.css
* ui-13. 加载中 /css/lib/loading.css

## 3. CSS文件的目录管理

所有的CSS分为两大类：通用类和业务类。

* 3-1. 通用的CSS文件，放在如下目录中：

|  |  |
| --- | --- |
| 基本样式库 | /css/core |
| 通用UI元素样式库 | /css/lib |
| JS组件相关样式库 | /css/ui |

* 3-2. 业务类的CSS是指和具体产品（如音乐、读书等）相关，放在如下目录中：

|  |  |
| --- | --- |
| 读书 | /css/book/ |
| 电影 | /css/movie/ |
| 音乐 | /css/music/ |
| 社区 | /css/sns/ |
| 小站 | /css/site/ |
| 同城 | /css/location/ |
| 电台 | /css/radio/ |
| 九点 | /css/newnine/ |
| 商务 | /css/biz/ |
| ... | /css/产品名称 |

* 3-3. 外联CSS文件适用于全站级和产品级通用的大文件。
* 3-4. 内联CSS文件适用于在一个或几个页面共用的CSS。利用模板系统支持的istatic方法引用。
* 3-5. 模块依赖的CSS和模块放在一起。利用模板系统支持的collect\_css过滤器实现。

## 4. CSS的模块化组织

当前静态文件管理系统已支持SCSS和导入语法。开发中，应按功能分解大文件，实现模块化的文件组织。

* 使用和完善现有CSS库（见2-4）。
* 单个CSS文件避免过大（建议少于300行）。
* CSS文件嵌套不要超过一层。

## 5. 不要轻易改动全站级CSS和通用CSS库。改动后，要经过全面测试。

## 6. 单条CSS规则的书写格式要求

* 6-1. 单行形式适用于直接写在页面中和长文件的情况。声明写在一行。需要在“{"和"}”前后加空格。

（注：在很长的文件中，单行形式有利于检索选择器）

.selector { property:value;property:value; }

简短规则声明（1或2个）也适用单行形式。

.selector { property:value; }

* 6-2. 格式化书写形式。适用于不是很长的模块文件或CSS3语法。冒号后加空格。

        .selector {   
          property: value;  
          property: value;   
        }

CSS3兼容书写形式和对齐方式：

.selector {   
 -webkit-box-shadow: 0 0 5px rgba(200, 200,200, 0.8);  
    -moz-box-shadow: 0 0 5px rgba(200, 200,200, 0.8);  
             box-shadow: 0 0 5px rgba(200, 200,200, 0.8);  
}

CSS3中逗号分隔的长属性值：

.selector {  
   box-shadow:  
       1px 1px 1px #000,  
       2px 2px 1px 1px #ccc inset;  
   background-image:  
       linear-gradient(#fff, #ccc),  
       linear-gradient(#f3c, #4ec);  
 }

* 6-3. 多个(>2)selector每个占一行：

       .selector1,  
      .selector2,  
      .selector3 { ... }

* 6-4. 规则声明的顺序：定位、盒模型（width/height/padding/border/margin）、行高、字体/字号/颜色、背景、CSS3效果等
* 6-5. 兼容多个浏览器时，将标准规则声明写在后面，如：

         -webkit-border-radius: 4px;  
           -moz-border-radius: 4px;  
                    border-radius: 4px;

## 7. 注释书写形式。

* 7-1. 注释内容单行控制在40个中文或80个英文字符宽。注释的格式：

      /\* mod: doulist   
      \* 一二三四五六七八九十一二三四五六七八九十一二三四五六七八九十一二三四五六七八九十  
      \* abcdefghijabcdefghijabcdefghijabcdefghijabcdefghijabcdefghijabcdefghijabcdefghij  
      \*/

     SCSS中支持单行注释

// mod: doulist

* 7-2. 规则分类放在一起。通用规则在具体业务规则的前面。如：

         /\* layout \*/  
         ...  
  
       /\* mod \*/  
       ...  
  
      /\* nav \*/  
      ...  
  
      /\* mod: events album \*/  
      ...

## 8. ID和Class命名。命名不要用缩写(除一些公认的缩写，见8-4)，单词间用"-"做为连接符。

* 8-1. ID是用来标识具体模块，命名必须具体且唯一，由前缀和名字组成。不要滥用ID。如： #db-video-list、#group-member-list等。
* 8-2. Class是用来标识某一类型的对象，命名简洁表意清楚。如：.list。
* 8-3. 命名示例：

坏：

#rec  
.gray-link  
.broadsmr  
.pl

好：

#db-nav-main  
.infobox  
.item

* 8-4. 推荐使用的class名：

|  |  |
| --- | --- |
| 表示状态 | .on, .active, .selected, .hili |
| 表示位置 | .first, .last, .main, .side |
| 表示结构 | .hd, .bd, .ft, .col, .section |
| 通用元素 | .tb, .frm, .nav, .list, .item, .tag, .pic, .info |

## 9. 避免滥用CSS Hack。

推荐使用下面的：

区别属性：

|  |  |
| --- | --- |
| IE6 | \_property:value |
| IE6/7 | \*property:value |
| IE6/7/8/9 | property:value\9 |

区别规则：

|  |  |
| --- | --- |
| IE6 | \* html selector { ... } |
| IE7 | \*:first-child+html selector { ... } |
| 非IE6 | html>body selector { ... } |
| firefox only | @-moz-document url-prefix() { ... } |
| saf3+/chrome1+ | @media all and (-webkit-min-device-pixel-ratio:0) { ... } |
| opera only | @media all and (-webkit-min-device-pixel-ratio:10000),not all and (-webkit-min-device-pixel-ratio:0) { ... } |
| iPhone/mobile webkit | @media screen and (max-device-width: 480px) { ... } |

注意：SCSS会自动转换一些不标准的CSS写法，会破坏CSSHack。

## 10. 使用after或overflow的方式清浮动，不要在html里增加多余的标签。

## 11. CSS必须放在head里。

## 12. 避免使用低效的选择器。

如：

body > \* {...}  
ul > li > a {...}  
#footer > h3 {...}  
ul#top\_blue\_nav {...}  
#searbar span.submit a { ... }  
.target #target-node { ... }

## 13. 避免使用filter

## 14. 避免直接定义标签的样式。如： div { ... }

## 15. 避免在标签上直接写样式。如：<div style="margin-bottom:30px;">

## 16. 避免在CSS中使用expression

## 17. 避免在CSS中使用@import

## 18. 尽量不要在CSS中使用!important

## 19. 绝对不要在CSS中使用"\*"选择符

最后更新日期：2012年6月12日