|  |  |
| --- | --- |
| 编码规范 | 密级 |
|  | 机密 |
| 产品版本 |  |
| 1.0 |

编码规范

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制 |  | 日期 |  |
| 评审人 |  | 日期 |  |
| 批准 |  | 日期 |  |



雅座在线（北京）科技发展有限公司

修订记录

| 日期 | 修订 版本 | 编辑 章节 | 编辑描述 | 作者 |
| --- | --- | --- | --- | --- |
| 2012-02-08 | 1.0 |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

目录

[修订记录 2](#_Toc316472609)

[1、注释 4](#_Toc316472610)

[1.1. 文件注释 4](#_Toc316472611)

[1.2. 函数注释 4](#_Toc316472612)

[1.3. 类注释 5](#_Toc316472613)

[2、代码 5](#_Toc316472614)

[2.1. 缩进 5](#_Toc316472615)

[2.2. 每行长度 5](#_Toc316472616)

[2.3．变量声明 5](#_Toc316472617)

[2.4. 函数 5](#_Toc316472618)

[2.5. 命名 6](#_Toc316472619)

[2.6. 语句 6](#_Toc316472620)

[2.6.1. return 语句 7](#_Toc316472621)

[2.6.2. if 语句 7](#_Toc316472622)

[2.6.3. for 语句 7](#_Toc316472623)

[2.6.4. while 语句 8](#_Toc316472624)

[2.6.5. do 语句 8](#_Toc316472625)

[2.6.6. switch 语句 8](#_Toc316472626)

[2.6.7. try 语句 8](#_Toc316472627)

[2.6.8. continue 语句 9](#_Toc316472628)

[2.6.9. with 语句 9](#_Toc316472629)

[3. 几点要求和建议 9](#_Toc316472630)

# 1、注释

遵循JsDoc[[1]](#footnote-1)的注释风格

## 文件注释

/\*\*

\* @encode: utf-8

\* @version: 1.0

\* @date: 2011/2/24 10:54:38

\* @copyright (c) 2011, by 13

\*

\* @author: 13

\* @contact: QQ: 353815844 MSN:qqworld@hotmail.com

\*

\* @license This file is released under the

\* <a target="\_blank" href="http://www.gnu.org/licenses/gpl.html">GNU GPL 3.0</a>

\* license. It’s free for use in GPL and GPL compatible open source software,

\* but if you want to use the component in a commercial software (closed source),

\* you have to get a commercial license.

\*/

## 函数注释

/\*\*

\* 一段函数的简单说明

\*

\* @param {String} sName 名称

\* @param {Number} nValue 值

\* @return {Object} 返回生成对象

\*/

## 类注释

/\*\*

\* 对话框窗口

\* @class Window

\* @extends Panel

\* @param {Object} opts 配置选项

\*/

单行注释，以//开头。多行注释，用/\*\*/包裹。

# 2、代码

## 2.1. 缩进

缩进的单位为四个空格。允许使用 Tab 键来控制缩进。

## 2.2. 每行长度

避免每行超过 80 个字符。当一条语句一行写不下时，请考虑折行。在运算符号，最好是逗号后换行。在运算符后换行可以减少因为复制粘贴产生的错误被分号掩盖的几率。

## 2.3．变量声明

所有的变量必须在使用前进行声明。JavaScript 并不强制必须这么做，但这么做可以让程序易于阅读，且也容易发现那些没声明的变量(它们会被编译成全局变量)。

多变量声明推荐采用逗号分隔的方式，每个变量单独占据一行，如：

var a=1, //

b=2, //

c=3; //

## 2.4. 函数

所有的函数在使用前进行声明。一般采用如下两种声明方式：

function fn1(p1,p2){

//do something

}

或:

var fn1=function(p1,p2){

//do something

}; //注意以分号结尾

如果使用匿名函数，并在声明后立即执行，采用如下两种写法之一：

(function(){

//do something

}());

或:

(function(){

//do something

})();

不限制闭包的使用，但需清楚闭包的使用方式和目的，并且明白页面中存在过多的闭包对性能来讲不是一件好事。

## 2.5. 命名

变量名应由 26 个大小写字母（A..Z，a..z），10 个数字（0..9），和 “\_”（下划线）组成，使用驼峰式或下划线连接的方式。避免使用国际化字符（如中文），因为它们不是在任何地方都可以被方便的阅读和理解。

以“\_”（下划线）作为第一个字符的变量一般用来表示私有变量，但实际上 JavaScript 并没提供私有变量的功能，只是作为一种约定规则。

用作构造器（配合new关键字使用）的函数名首字母大写。

宏变量名全部大写。

## 2.6. 语句

简单语句每一行最多只包含一条语句。把 “;” （分号）放到每条简单语句的结尾处。注意一个函数赋值或对象赋值语句也是赋值语句，应该以分号结尾。  
 JavaScript 可以把任何表达式当作一条语句。这很容易隐藏一些错误，特别是误加分号的错误。只有在赋值和调用时，表达式才应被当作一条单独的语句。

复合语句复合语句是被包含在 “{ }”（大括号）的语句序列。

被括起的语句必须多缩进四个空格。“{”（左大括号）应在复合语句其实行的结尾处。“}”（右大括号）应与“{”（左大括号）的那一行的开头对齐。大括号应该在所有复合语句中使用，即使只有一条语句，当它们是控制结构的一部分时，比如一个 if 或者 for 语句。这样做可以避免以后添加语句时造成的错误。

### 2.6.1. return 语句

一条有返回值的 return 语句不要使用“( )”（括号）来括住返回值。如果返回表达式，则表达式应与 return 关键字在同一行，以避免误加分号错误。

### 2.6.2. if 语句

if 语句应如以下格式:

if (condition){

statements;

}

if (condition) {

statements;

} else {

statements;

}

if (condition) {

statements;

} else if (condition) {

statements;

} else {

statements;

}

### 2.6.3. for 语句

for 语句应如以下格式:

for (initialization;condition; update) {

statements;

}

for (variable in object) if (filter) {

statements;

}

第一种形式的循环用于已经知道相关参数的数组循环。

第二种形式应用于对象中。Object 原型中的成员将会被包含在迭代器中。通过预先定义 hasOwnProperty方法来区分真正的 object 成员是个不错方法：

for (variablein object) {

if (object.hasOwnProperty(variable)){

statements;

}

}

### 2.6.4. while 语句

while 语句应如以下格式:

while (condition){

statements;

}

### 2.6.5. do 语句

do 语句应如以下格式:

do {

statements;

} while (condition);

不像别的复合语句，do 语句总是以 “;”（分号）结尾。

### 2.6.6. switch 语句

switch 语句应如以下格式:

switch (expression){

case expression:

statements;

default:

statements;

}

每个 case 与 switch 对齐。这可避免过分缩进。

每一组 statements（除了default 应以 break，return，或者 throw 结尾），不要让它顺次往下执行。

### 2.6.7. try 语句

try 语句应如以下格式:

try {

statements;

} catch (variable){

statements;

}

try {

statements;

} catch (variable){

statements;

} finally {

statements;

}

### 2.6.8. continue 语句

避免使用 continue 语句。它很容易使得程序的逻辑过程晦涩难懂。

### 2.6.9. with 语句

避免使用 with 语句。

# 3. 几点要求和建议

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

2、只有长语句可以考虑断行，注意“.”或“＋”这类操作符放在行尾，新断行与首行间隔一个“tab”空格，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);

或:

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  
        }  
   });

3、避免额外的逗号。如：var arr = [1,2,3,];或var obj={a:’v1’,b:’v2’,c:’v3’,};长数组或对象可考虑断行，如：

var obj={

a:{

aa:”v”,

ab:”v”,

ac:”v”

},

b:[“v”],

c:”v”

};

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; }

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。

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;

}

7、不要使用with, void, evil。

8、使用严格的条件判断符。用===代替==，用!==代替!=。

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)。

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

11、构造器的首字母大写。如：

function Dialog (config) {

statement;

}

var dlg = new Dialog({...});

12、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 () {},

 ...

};

13、私有变量名用下划线开头。如："\_current", "\_defaultConfig"

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

15、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 |
| a | array | aArrayName |
| is, can, has | boolean | [Boolean name]ConditionName |
| event method | event attachment | [event type]\_MethodName |
| get | accessor method | getMethodName |
| set | accessor method | setMethodName |

1. [jsdoc-toolkit](http://code.google.com/p/jsdoc-toolkit/) [↑](#footnote-ref-1)