# 编码规范

## javascript

### 1.1 命名规范

#### 1.1.1 公有变量、方法名称采用驼峰命名

例如 sProductName , fGetUserPassword

#### 1.1.2 私有变量、方法名称加下划线前缀

例如 \_sName, \_fGetUser

#### 1.1.3 普通变量前缀

以变量类型的第一个字母为前缀，类名、属性及没有限制类型的变量无需前缀，某些作用域很小的变量，如for循环中的变量i，允许无前缀：

1. 加强代码的可阅读性；
2. 拓展命名空间，如获取sHtml，转化为aHtml，就不需要另取名称

|  |  |  |  |
| --- | --- | --- | --- |
| 类型 | 标志 | 前缀 | 示例 |
| 布尔值 | {boolean} | b | bSample |
| 数值 | {number} | n | nSample |
| 字符串 | {string} | s | sSample |
| 数组 | {array} | a | aSample |
| 对象 | {object} | o | oSample |
| 元素 | {element} | e | eSample |
| 函数 | {function} | f | fSample |

### 注释规范

#### 1.2.1 文件声明：

包括类说明、作者、类名

类说明  
@author 作者  
@class Example

如：

1. /\*\*
2. \* 示例类
3. \* @author 郑银辉
4. \* @class Example
5. \*/

#### 1.2.2 方法注释：

包括方法描述、方法名称、参数说明、返回值说明  
无参数或返回值填写@param {void}、@return {void}，  
方法名称中[]里表示可选参数，  
参数是object类型时，后面加{具体说明},类型是array时，后面加[具体说明]  
参数可以多种类型的情况用|分开，如:{string|boolean}

方法描述  
@method example(param)  
@param param{type} 说明  
@return {type} 说明

如:

1. /\*\*
2. \* 示例方法
3. \* @method fFunc(sName,[oData,]aArray)
4. \* @param sName{string} 名称
5. \* @param oData{object} 绑定的数据(可选){
6. \* nId{number}:id号,
7. \* sContent{string}:内容
8. \* }
9. \* @param aArray{array} 元素数组[
10. \* {
11. \* element{element|string}:要删除的元素或元素id,
12. \* sTag{string}:要删除的元素的图片标签名,
13. \* }
14. \* ]
15. \* @return {void}
16. \*/

## 2 css

### 2.1 提高页面渲染效率

#### 2.1.1 html代码中尽量不用样式相关的代码

#### 2.1.2 css类名的层级尽量浅 最多3级

#### 2.1.3 class和id前面不要添加标签名

#### 2.1.4 尽量不在css中直接使用html标签匹配

#### 2.1.5 颜色值使用大写字母

### 2.2 其它

#### 2.2.1 class命名时：g-表示html结构，m-/w-表示内容，f-表示通用功能

#### 2.2.2 class命名同时体现父子关系

如g-sd表示左栏结构，g-sd-bd表示左栏body结构，w-sd-bd-menu表示左栏body体内的菜单组件

#### 2.2.3 关于背景图片和颜色的css样式只能在皮肤的css中

## 3 后端接口数据

统一ajax返回数据

code:{String}操作返回的状态,成功为S\_OK, 其它都为异常

data:{Array} 操作返回的数据,单个数据也包装在array里面,统一格式.

(其中更新操作,必须包含主键值和有更新的属性. 删除操作,必须包含主键值)

其它数据(如分页相关属性)

例如:

{

code : "S\_OK",

data : [

{

type: "",

val : "",

mid: "mid1"

},

{

type: "",

val : "",

mid: "mid2"

}

],

其它数据(如分页相关属性)

}