[代码规范](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472)

[转至元数据结尾](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#page-metadata-end)

* 创建： [莫鹏程](http://10.11.112.49:8090/display/~MOPENGCHENG524)，最新修改于： [十一月 30, 2016](http://10.11.112.49:8090/pages/diffpagesbyversion.action?pageId=2579472&selectedPageVersions=9&selectedPageVersions=10)

[转至元数据起始](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#page-metadata-start)

目录

[**命名规则**](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#naming)

* [项目命名](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#project-naming)
* [目录命名](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#folder-naming)
* [EJS、JS文件命名原则](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#folder-naming)
* [JS文件命名](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-naming)
* [CSS文件命名](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#css-naming)
* [HTML、EJS文件命名](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#html-naming)

[**HTML**](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#html)

* [重要说明](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#html-warning)
* [语法](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#html-syntax)
* [HTML5 doctype、lang](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#html-doctype)
* [引入CSS, JS](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#html-style-script)
* [属性顺序](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#html-attribute-order)
* [JS生成标签](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#html-javascript)

[**JavaScript**](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#javascript)

* [缩进](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-indentation)
* [分号](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-semicolon)
* [空格](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-space)
* [空行](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-blank-line)
* [换行](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-new-line)
* [单行注释](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-comments-single-line)
* [多行注释](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-comments-multiline)
* [文档注释](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-comments-documentation)
* [引号](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-quote-marks)
* [变量命名](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-variable-naming)
* [变量声明](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-variable-declaration)
* [函数](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-function)
* [数组、对象](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-array-object)
* [括号](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-brace)
* [null](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-literal-value-null)
* [undefined](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-literal-value-undefined)
* [jshint](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-jshint)
* [杂项](http://10.11.112.49:8090/pages/viewpage.action?pageId=2579472#js-miscellaneous)

命名规则

**项目命名**

全部采用小写方式， 以下划线分隔。

例：my\_project\_name

**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

**EJS、JS文件命名原则**

实现同一页面的EJS、JS应当使用相同的命名，以建立明了的对应关系。

例：error\_report.ejs 对应 error\_report.js

**JS文件命名**

参照项目命名规则。

例：account\_model.js

**CSS文件命名**

参照项目命名规则。

例：retina\_sprites.css

**HTML、EJS文件命名**

参照项目命名规则。

例：error\_report.ejs

HTML(注意重要说明)

**重要说明**

H5开发人员请不要改变HTML、EJS文件，如需改变请联系重构进行修改。

**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，要在js文件中使用的属性名以“js-”开头；
* 特殊代码块要有特定标识，如模板id应包含“Tpl”字段
* 自定义属性要加data-，如"data-code"；
* 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>。

**HTML5 doctype、lang、字符编码（由header引入）**

在页面开头使用这个简单地doctype来启用标准模式，DOCTYPE 需要大写；

应在html标签上加上lang属性。

字符编码通常指定为'UTF-8'

**引入CSS, JS**

一般在HTML中不需要H5开发人员加入样式，样式由重构提供统一的css。如H5开发过程中需加入样式，请在头部通过 <style> ... </style>引入

通常在引入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。

**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。

**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。

JavaScript

**缩进**

使用soft tab（4个空格）。

**分号**

以下几种情况后需加分号：

* 变量声明
* 表达式
* return
* throw
* break
* continue
* do-while

**空格**

以下几种情况不需要空格：

* 对象的属性名后
* 前缀一元运算符后
* 后缀一元运算符前
* 函数调用括号前
* 无论是函数声明还是函数表达式，'('前不要空格
* 数组的'['后和']'前
* 对象的'{'后和'}'前
* 运算符'('后和')'前

以下几种情况需要空格：

* 二元运算符前后
* 三元运算符'?:'前后
* 代码块'{'前
* 下列关键字前：else, while, catch, finally
* 下列关键字后：if, else, for, while, do, switch, case, try, catch, finally, with, return, typeof
* 单行注释'//'后（若单行注释和代码同行，则'//'前也需要），多行注释'\*'后
* 对象的属性值前
* for循环，分号后留有一个空格，前置条件如果有多个，逗号后留一个空格
* 无论是函数声明还是函数表达式，'{'前一定要有空格
* 函数的参数之间

**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

**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

**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。

**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

**文档注释**

各类标签@param, @method等；

建议在以下情况下使用：

* 所有常量
* 所有函数
* 所有类

**引号**

最外层统一使用单引号。

**变量命名**

* 标准变量采用驼峰式命名（除了对象的属性外，主要是考虑到cgi返回的数据）
* '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

**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。

**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；

不要给inline function命名；

参数之间用', '分隔，注意逗号后有一个空格。

**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。

**括号**

下列关键字后必须有大括号（即使代码块的内容只有一行）：if, else, for, while, do, switch, try, catch, finally, with。

**null**

适用场景：

* 初始化一个将来可能被赋值为对象的变量
* 与已经初始化的变量做比较
* 作为一个参数为对象的函数的调用传参
* 作为一个返回对象的函数的返回值

不适用场景：

* 不要用null来判断函数调用时有无传参
* 不要与未初始化的变量做比较

**undefined**

永远不要直接使用undefined进行变量判断；

使用typeof和字符串'undefined'对变量进行判断。

**jshint**

* 用'===', '!=='代替'==', '!='；
* for-in里一定要有hasOwnProperty的判断；
* 不要在内置对象的原型上添加方法，如Array, Date；
* 不要在内层作用域的代码里声明了变量，之后却访问到了外层作用域的同名变量；
* 变量不要先使用后声明；
* 不要在一句代码中单单使用构造函数，记得将其赋值给某个变量；
* 不要在同个作用域下声明同名变量；
* 不要在一些不需要的地方加括号，例：delete(a.b)；
* 不要使用未声明的变量（全局变量需要加到.jshintrc文件的globals属性里面）；
* 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；
* 不要像这样使用构造函数，例：new function () { ... }, new Object；

**杂项**

* 不要混用tab和space；
* 不要在一处使用多个tab或space；
* 换行符统一用'LF'；
* 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；
* switch的falling through和no default的情况一定要有注释特别说明；
* 不允许有空的代码块。