|  |  |  |  |
| --- | --- | --- | --- |
| **深圳国泰安教育技术股份有限公司** | **版本** | **密级** | **页数** |
| **V1.0** | **机密** | **共23页** |
| **文档编号:** **GTA\_CMMI\_CodeStandard\_Web** | | |

**国泰安软件编码规范**

**HTML、CSS、Javascript**

|  |  |  |  |
| --- | --- | --- | --- |
| **作者** | 刘兵 | **编制日期** | 2014-04-10 |
| **审核** |  | **审核日期** |  |
| **批准** |  | **批准日期** |  |



深圳国泰安教育技术股份有限公司

版权所有 侵权必究

**文档修改记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **版号** | **修改日期** | **修改内容** | **修改人** |
| V1.0 | 2014-04-10 | 根据CMMI过程改进要求编制该文档 | 刘兵 |
|  |  |  |  |
|  |  |  |  |

**目录**

[1 简介 5](#_Toc384969411)

[2 适用范围 5](#_Toc384969412)

[3 文体 5](#_Toc384969413)

[4 HTML 编码规范 5](#_Toc384969414)

[4.1 基本规范 5](#_Toc384969415)

[4.1.1 语义 5](#_Toc384969416)

[4.1.2 大小写 6](#_Toc384969417)

[4.1.3 缩进 6](#_Toc384969418)

[4.1.4 空格 6](#_Toc384969419)

[4.1.5 嵌套 7](#_Toc384969420)

[4.1.6 引号 7](#_Toc384969421)

[4.1.7 自定义属性 7](#_Toc384969422)

[4.1.8 DOCTYPE 8](#_Toc384969423)

[4.1.9 编码 8](#_Toc384969424)

[4.1.10 注释 8](#_Toc384969425)

[4.1.11 协议 8](#_Toc384969426)

[4.1.12 TODO 9](#_Toc384969427)

[4.1.13 焦点分离 9](#_Toc384969428)

[4.2 元素 10](#_Toc384969429)

[4.3 结构性元素 10](#_Toc384969430)

[4.4 头部元素 10](#_Toc384969431)

[4.5 文本元素 11](#_Toc384969432)

[4.5.1 媒体元素 11](#_Toc384969433)

[4.5.2 列表元素 12](#_Toc384969434)

[4.5.3 表单元素 12](#_Toc384969435)

[4.6 文档模板 12](#_Toc384969436)

[5 CSS 编码规范 13](#_Toc384969437)

[5.1 css 校验 13](#_Toc384969438)

[5.2 代码规范 13](#_Toc384969439)

[5.2.1 一般性命名 13](#_Toc384969440)

[5.2.2 id 和类的命名 14](#_Toc384969441)

[5.2.3 元素选择器 15](#_Toc384969442)

[5.2.4 简写属性名字 15](#_Toc384969443)

[5.2.5 0 和单位 16](#_Toc384969444)

[5.2.6 0 前缀情况 16](#_Toc384969445)

[5.2.7 16 进制的颜色值表示 16](#_Toc384969446)

[5.2.8 前缀 16](#_Toc384969447)

[5.2.9 hacks 16](#_Toc384969448)

[5.3 格式规范 17](#_Toc384969449)

[5.3.1 属性声明顺序 17](#_Toc384969450)

[5.3.2 块缩进 17](#_Toc384969451)

[5.3.3 分号 18](#_Toc384969452)

[5.3.4 空格 18](#_Toc384969453)

[5.3.5 空行 19](#_Toc384969454)

[5.3.6 引号 20](#_Toc384969455)

[5.3.7 注释 20](#_Toc384969456)

[6 JavaScript 编码风格 21](#_Toc384969457)

[6.1 行与缩进 21](#_Toc384969458)

[6.1.1 语句行 21](#_Toc384969459)

[6.1.2 空格 21](#_Toc384969460)

[6.1.3 空行 22](#_Toc384969461)

[6.1.4 缩进 22](#_Toc384969462)

[6.2 括号 22](#_Toc384969463)

[6.3 变量 23](#_Toc384969464)

[6.4 字符串 23](#_Toc384969465)

[6.5 命名规范 23](#_Toc384969466)

# 简介

本规范为一套编写高效可靠的 html、css、javascript代码的标准、约定和指南。它以安全可靠的软件工程原则为基础，使代码易于理解、维护和增强，提高生产效率。同时，将带来更大的一致性，使软件开发团队的效率明显提高。

# 适用范围

本规范适用于公司所有的web网页代码，为详细设计，代码编写和代码审核提供参考和依据。

# 文体

本规范中的建议分为四种：**要，建议，避免，不要**，表示需要遵循的级别。文档中会以粗体表示。对于应遵循的规范，前面会以“Ö”来表示，对不好的做法前面会以“´”来表示：

**要**：描述必须遵循的规范。例如：

Ö 异常类**要**以“Exception”做为后缀；

**建议**：描述在一般情况下应该遵循的规范，但如果完全理解规范背后的道理，并有很好的理由不遵循它时，也不畏惧打破常规。例如：

Ö 强制类型转换时，在类型和变量之间**建议**加一空格。

**不要**：描述一些几乎绝对绝不应该违反的规范。例如：

´ 每个函数有效代码（不包括注释和空行）长度**不要**超过50行。

**避免**：与**建议**相对，一般情况下应该遵循，但有很好的理由时也可以打破。例如：

´ **避免**块内部的变量与它外部的变量名相同。

对一些规范内容一并提供了示例代码。

# HTML 编码规范

## 基本规范

### 语义

**要**使用符合语义的标签书写 HTML 文档, 选择恰当的元素表达所需的含义;

*<!-- 不推荐 -->*

**<div** onclick="goToRecommendations();"**>**All recommendations**</div>**

*<!-- 推荐 -->*

**<a** href="recommendations/"**>**All recommendations**</a>**

### 大小写

元素的标签和属性名必须**要**小写, 属性值必须**要**加双引号; 例如

*<!-- 不推荐 -->*

**<A** HREF='/'**>**Home**</A>**

*<!-- 推荐 -->*

**<a** href="/"**>**Home**</a>**

### 缩进

* 使用四个空格来表示缩进，**不要**使用 tab 键;
* **要**在块状元素，列表，表格元素后面使用新行，并且对它的子元素进行缩进.

例如

**<ul>**

**<li>**

1

**</li>**

**</ul>**

### 空格

**建议**去除比不必要的空格; 例如

*<!-- 不推荐 -->*

**<p>**test **</p>**

*<!-- 推荐 -->*

**<p>**test**</p>**

### 嵌套

* 元素嵌套**要**遵循 (X)HTML Strict 嵌套规则, **建议**使用Firefox插件 [HTML Validator](http://www.w3.org/TR/html4/) 进行检查;
* 正确区分自闭合元素和非自闭合元素. 非法闭合包括：<br>..</br>、<script />、<iframe />, 非法闭合会导致页面嵌套错误问题;

*<!-- 不推荐 -->*

**<title>**Test**</title>**

**<article>**This is only a test.

*<!-- 推荐 -->*

<!DOCTYPE html>

**<meta** charset="utf-8"**>**

**<title>**Test**</title>**

**<article>**This is only a test.**</article>**

### 引号

**建议**使用双引号来标识 html 的属性; 例如

*<!-- 不推荐 -->*

**<a** class='maia-button maia-button-secondary'**>**Sign in**</a>**

*<!-- 推荐 -->*

**<a** class="maia-button maia-button-secondary"**>**Sign in**</a>**

### 自定义属性

**建议**通过给元素设置自定义属性来存放与 JavaScript 交互的数据, 属性名格式为 data-xx (例如：data-lazyload-url)

### DOCTYPE

页面文档类型统一使用HTML5 DOCTYPE. 代码如下：

<!doctype html>

### 编码

声明方法遵循HTML5的规范.**推荐**使用 utf-8 编码.

**<meta** charset="utf-8" **/>**

### 注释

**建议**对超过10行的页面模块进行注释, 以降低开发人员的嵌套成本和后期的维护成本. 例如：

**<div** id="sample"**>**

...

**</div>** *<!-- #sample END -->*

**<div** class="sample"**>**

...

**</div>** *<!-- .sample END -->*

### 协议

如果链接和当前页面一致则忽略链接的协议部分，例如

*<!-- 不推荐 -->*

**<script** src="http://www.taobao.com/fp.js"**></script>**

*<!-- 推荐 -->*

**<script** src="//www.taobao.com/fp.js"**></script>**

*/\* 不推荐 \*/*

**.example** {

**background**: url(http://www.taobao.com/fp.css);

}

*/\* 推荐 \*/*

**.example** {

**background**: url(//www.taobao.com/fp.css);

}

### TODO

* **建议**使用 TODO 来标记待做事情，便于后期搜索.
* 在 TODO 后添加 (姓名或邮件) 来表示分类.

例如

*<!-- TODO(yiminghe): remove duplicate tag -->*

**<p><span>**2**</span></p>**

### 焦点分离

* 将表现，行为和结构分离：**不要**在 html 和模板中加入除了结构以外的东西.
* 在文档中引入尽可能少的样式和脚本

*<!-- 不推荐 -->*

<!DOCTYPE html>

**<title>**HTML sucks**</title>**

**<link** rel="stylesheet" href="base.css" media="screen"**>**

**<link** rel="stylesheet" href="grid.css" media="screen"**>**

**<link** rel="stylesheet" href="print.css" media="print"**>**

**<h1** style="font-size: 1em;"**>**HTML sucks**</h1>**

**<p>**I’ve read about this on a few sites but now I’m sure:

**<u>**HTML is stupid!!1**</u>**

**<center>**I can’t believe there’s no way to control the styling of

my website without doing everything all over again!**</center>**

*<!-- 推荐 -->*

<!DOCTYPE html>

**<title>**My first CSS-only redesign**</title>**

**<link** rel="stylesheet" href="default.css"**>**

**<h1>**My first CSS-only redesign**</h1>**

**<p>**I’ve read about this on a few sites but today I’m actually

doing it: separating concerns and avoiding anything in the HTML of

my website that is presentational.

**<p>**It’s awesome!

## 元素

## 结构性元素

* p 表示段落. 只能包含内联元素, 不能包含块级元素;
* div 本身无特殊含义, 可用于布局. 几乎可以包含任何元素;
* br 表示换行符;
* hr 表示水平分割线;
* h1-h6 表示标题. 其中 h1 用于表示当前页面最重要的内容的标题;
* blockquote 表示引用, 可以包含多个段落. 请勿纯粹为了缩进而使用 blockquote, 大部分浏览器默认将 blockquote 渲染为带有左右缩进;
* pre 表示一段格式化好的文本;

## 头部元素

* title 每个页面**必须有且仅有一个** title 元素;
* base 可用场景：首页、频道等大部分链接都为新窗口打开的页面;
* link link 用于引入 css 资源时, 可省去 media(默认为all) 和 type(默认为text/css) 属性;
* style type 默认为 text/css, 可以省去;
* script type 属性可以省去; 不赞成使用lang属性; 不要使用古老的<!– //–>这种hack脚本, 它用于阻止第一代浏览器(Netscape 1和Mosaic)将脚本显示成文字;

*<!-- 不推荐 -->*

**<link** rel="stylesheet" href="//www.google.com/css/maia.css"

type="text/css"**>**

*<!-- 不推荐 -->*

**<script** src="//www.google.com/js/gweb/analytics/autotrack.js"

type="text/javascript"**></script>**

*<!-- 推荐 -->*

**<link** rel="stylesheet" href="//www.google.com/css/maia.css"**>**

*<!-- 推荐 -->*

**<script** src="//www.google.com/js/gweb/analytics/autotrack.js"**></script>**

* noscript 在用户代理不支持 JavaScript 的情况下提供说明;

## 文本元素

* a a 存在 href 属性时表示链接, 无 href 属性但有 name 属性表示锚点;
* em,strong em 表示句意强调, 加与不加会引起语义变化, 可用于表示不同的心情或语调; strong 表示重要性强调, 可用于局部或全局, strong强调的是重要性, 不会改变句意;
* abbr 表示缩写;
* sub,sup 主要用于数学和化学公式, sup还可用于脚注;
* span 本身无特殊含义;
* ins,del 分别表示从文档中增加(插入)和删除

### 媒体元素

* img 请**不要**将img元素作为定位布局的工具, **不要**用他显示空白图片;**要** 给img元素增加alt属性;例如

*<!-- 不推荐 -->*

**<img** src="spreadsheet.png"**>**

*<!-- 推荐 -->*

**<img** src="spreadsheet.png" alt="Spreadsheet screenshot."**>**

* object 可以用来插入Flash;

### 列表元素

* dl 表示关联列表, dd是对dt的解释; dt和dd的对应关系比较随意： 一个dt对应多个dd、多个dt对应一个dd、多个dt对应多个dd, 都合法; 可用于名词/单词解释、日程列表、站点目录;
* ul 表示无序列表;
* ol 表示有序列表, 可用于排行榜等;
* li 表示列表项, 必须是ul/ol的子元素;

### 表单元素

* **推荐**使用 button 代替 input, 但必须声明 type;
* **推荐**使用 fieldset, legend 组织表单
* 表单元素的 name **不要**设定为 action, enctype, method, novalidate, target, submit 会导致表单提交混乱

## 文档模板

<!doctype html>

**<html>**

**<head>**

**<meta** charset="utf-8" **/>**

**<title>**Sample page**</title>**

**<link** rel="stylesheet" href="css\_example\_url" **/>**

**</head>**

**<body>**

**<div** id="page"**>**

**<div** id="header"**>**

页头

**</div>**

**<div** id="content"**>**

主体

**</div>**

**<div** id="footer"**>**

页尾

**</div>**

**</div>**

**<script** src="js\_example\_url"**></script>**

**<script>**

// 你的代码

**</script>**

**</body>**

**</html>**

# CSS 编码规范[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "css" \o "Permalink to this headline)

## css 校验[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id2" \o "Permalink to this headline)

除了 css hack 和浏览器私有属性， **推荐**使用 [w3c css validator](http://jigsaw.w3.org/css-validator/) 校验其余的代码.

## 代码规范[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id3" \o "Permalink to this headline)

### 一般性命名[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id4" \o "Permalink to this headline)

使用小写字母，复合词**建议**以 - 分隔; 例如 nav.css , login-nav.css , login-page

### id 和类的命名[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id" \o "Permalink to this headline)

为 id 和样式类使用有意义或通用的名字，**避免**由于 css 命名更改引起的不必要的文档或模板改变；例如

/\* 不推荐： 无意义 \*/

#yee-1901 {}

/\* 不推荐： 表现层的命名 \*/

.button-green {}

/\* 推荐: 具体 \*/

#gallery {}

#login {}

.video {}

/\* 推荐: 通用 \*/

.effect {}

.alt {}

id 和 class 的命名长度**建议**适中，不要太简略也不要太详细；例如

/\* 不推荐 \*/

#navigation {}

.atr {}

/\* 推荐 \*/

#nav {}

.author {}

### 元素选择器[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id5" \o "Permalink to this headline)

为了 [性能原因](http://www.stevesouders.com/blog/2009/06/18/simplifying-css-selectors/) ， 请**避免**元素选择器和类选择器以及 id 选择器混用;例如

/\* 不推荐 \*/

ul#example {}

div.error {}

/\* 推荐 \*/

#example {}

.error {}

### 简写属性名字[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id7" \o "Permalink to this headline)

为了提高可读性，**建议**使用简写属性。例如

/\* 不推荐 \*/

border-top-style: none;

font-family: palatino, georgia, serif;

font-size: 100%;

line-height: 1.6;

padding-bottom: 2em;

padding-left: 1em;

padding-right: 1em;

padding-top: 0;

/\* 推荐 \*/

border-top: 0;

font: 100%/1.6 palatino, georgia, serif;

padding: 0 1em 2em;

### 0 和单位[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id8" \o "Permalink to this headline)

**建议**对属性值为 0 的情况省略单位；例如

margin: 0;

padding: 0;

### 0 前缀情况[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id9" \o "Permalink to this headline)

**建议**省略属性值中的 0 前缀;例如

font-size: .8em;

### 16 进制的颜色值表示[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id10" \o "Permalink to this headline)

**建议**尽可能使用 3 个字符的 16 进制颜色值；例如

/\* 不推荐 \*/

color: #eebbcc;

/\* 推荐 \*/

color: #ebc;

### 前缀[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id11" \o "Permalink to this headline)

为了防止冲突，对于应用特定的样式附加应用前缀；例如

.login-help {} /\* login page \*/

#detail-note {} /\* detail page \*/

### hacks[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "hacks" \o "Permalink to this headline)

**避免** css hack ， 考虑使用特定浏览器前缀表示；例如

.ks-ie6 p {

margin: 1em 0;

}

## 格式规范[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id12" \o "Permalink to this headline)

### 属性声明顺序[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id13" \o "Permalink to this headline)

按照字母顺序声明属性，排序时忽略私有的浏览器前缀，对于特定的浏览器，私有的浏览器前缀应该参与排序;例如

background: fuchsia;

border: 1px solid;

-moz-border-radius: 4px;

-webkit-border-radius: 4px;

border-radius: 4px;

color: black;

text-align: center;

text-indent: 2em;

### 块缩进[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id14" \o "Permalink to this headline)

块的内容**要**进行缩进；例如

@media screen, projection {

html {

background: #fff;

color: #444;

}

}

### 分号[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id15" \o "Permalink to this headline)

**推荐**使用分号结束单个属性的定义；例如

/\* 不推荐 \*/

.test {

display: block;

height: 100px

}

/\* 推荐 \*/

.test {

display: block;

height: 100px;

}

### 空格[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id16" \o "Permalink to this headline)

**推荐**在属性名冒号后加一个空格，例如

/\* 不推荐 \*/

.test {

display:block;

}

/\* 推荐 \*/

.test {

display: block;

}

### 空行[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id17" \o "Permalink to this headline)

**推荐**多个选择以及声明间以行分隔；例如

/\* 不推荐 \*/

a:focus, a:active {

position: relative; top: 1px;

}

/\* 推荐 \*/

h1,

h2,

h3 {

font-weight: normal;

line-height: 1.2;

}

多个 css 规则间以空行分隔；例如

html {

background: #fff;

}

body {

margin: auto;

width: 50%;

}

### 引号[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id18" \o "Permalink to this headline)

尽可能的**避免**使用引号，迫不得已时使用单引号.

/\* 不推荐 \*/

@import url("//www.google.com/css/maia.css");

html {

font-family: "open sans", arial, sans-serif;

}

/\* 推荐 \*/

@import url(//www.google.com/css/maia.css);

html {

font-family: 'open sans', arial, sans-serif;

}

### 注释[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/css-coding-style.html" \l "id19" \o "Permalink to this headline)

成组的 css 规则间**建议**使用块状注释和空行分离;例如

/\* Header \*/

#login-header {}

#login-header-below {}

/\* Footer \*/

#login-footer {}

#login-footer-below {}

/\* Gallery \*/

.login-gallery {}

.login-gallery-other {}

# JavaScript 编码风格[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/js-style-rules.html" \l "javascript" \o "Permalink to this headline)

## 行与缩进[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/js-style-rules.html" \l "id1" \o "Permalink to this headline)

### 语句行[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/js-style-rules.html" \l "id2" \o "Permalink to this headline)

1. 尽可能**避免**每行超过 120 个字符;
2. 语句必须以分号作为结束符, **不要**忽略分号;

### 空格[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/js-style-rules.html" \l "id3" \o "Permalink to this headline)

* 数值操作符(如, +/-/\*/% 等)**建议**两边留空;
* 赋值操作符/等价判断符两边**建议**留一空格;
* for 循环条件中, 分号后**建议**留一空格;
* 变量声明语句, 数组值, 对象值及函数参数值中的逗号后**建议**留一空格;
* 空行**不要**有空格;
* 行尾**不要**有空格;
* 逗号和冒号后一定**要**跟空格;
* 点号前后**不要**出现空格;
* 空对象和数组**避免**填入空格;
* 函数名末尾和左括号之间**不要**出现空格;

### 空行[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/js-style-rules.html" \l "id4" \o "Permalink to this headline)

1. **建议**逻辑上独立的代码块使用空行分隔;
2. **建议**文件末尾留 1~2 个空行;
3. 不要吝啬空行. **建议**使用空行将逻辑相关的代码块分割开, 以提高程序的可读性.

### 缩进[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/js-style-rules.html" \l "id5" \o "Permalink to this headline)

* **要**以 4 个空格为一缩进层次;
* 变量声明:
* 多个变量声明时, **建议**换行表示;
* 参照 var 关键字, 缩进一层次;
* 函数参数:
* 函数参数**建议**写在同一行上;
* 传递匿名函数时, 函数体**要**从调用该函数的左边开始缩进;
* 数组和对象初始化时:
* 如果初始值不是很长, **建议**保持写在单行上;
* 初始值占用多行时, **建议**缩进一层次;
* 对象中, 比较长的变量/数值, **不要**以冒号对齐;
* 二元/三元操作符:
* **建议**操作符始终跟随前行;
* 实在需要缩进时, 按照上述缩进风格;
* 表达式中的缩进同变量声明时;

## 括号[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/js-style-rules.html" \l "id6" \o "Permalink to this headline)

原则: **不要**滥用括号, 必要时一定要使用.

* if/else/while/for 条件表达式必须**要**有小括号;
* 语句块必须**要**有大括号;
* 一元操作符(如 delete, typeof, void)或在某些关键词(如 return, throw, case, new) 之后, **不要**使用括号;

## 变量[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/js-style-rules.html" \l "id7" \o "Permalink to this headline)

* 变量如有较广的作用域, 使用全局变量; 如果是在类中, 可以设计成为一个类的成员;
* 函数体中, 多个局部变量集中在一起声明, 避免分散;
* 适当延迟变量的初始化;

## 字符串[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/js-style-rules.html" \l "id8" \o "Permalink to this headline)

* JS 代码中, 单行字符串使用单引号;
* JS 代码中, 多行字符串使用 + 拼接形式, 不要使用 \ 拼接;
* HTML 中 Element 属性, 使用双引号;

## 命名规范[¶](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/js-style-rules.html" \l "id9" \o "Permalink to this headline)

原则： \* 尽量避免潜在冲突; \* 精简短小, 见名知意;

* 普通变量统一使用驼峰形式;
* 常量使用全部大写, 多个单词以下划线分隔;
* 枚举量, 同常量;
* 私有变量, 属性和方法, 名字以下划线开头;
* 保护变量, 属性和方法, 名字同普通变量名;
* 方法和函数的可选参数, 名字以 opt\_ 开头, 使用 @param 标记说明;
* **方法和函数的参数个数不固定时:**
  + 可添加参数 var\_args 为参数个数;
  + 取代使用 arguments;
  + 使用 @param 标记说明;
* **Getter/Setter 命名:**
  + 以 getFoo/setFoo(value) 形式;
  + 布尔类型使用 isFoo()/hasFoo()/canDo()/shouldDO() 也可;
* **命名空间:**
  + 为全局代码使用命名空间, 如 sloth.\*;
  + 外部代码和内部代码使用不同的命名空间;
* 重命名那些名字很长的变量, 不要在全局范围内创建别名, 而仅在函数块作用域中使用;
* 文件名应全部使用小写字符, 且不包含除 - 和 \_ 外的标点符号;
* 临时的重复变量建议以 i, j, k, ..., 命名;