**文件命名规则：**

文件名称统一使用小写英文字母、数字和下划线组合，不得包含汉字、空格和特殊字符；

**1. HTML命名规划**

文件统一使用.html，.html，.jsp，.aspx，.php文件名，且文件名小写；

**2. 图片命名规则**

图片的命名分为头尾二部分，用下划线隔开，头部表示此图片的大类性质，如广告（ad）、标志（banner）、菜单（menu）、按钮（btn）等；

**3. JavaScript文件命名**

根据类型、功能或模块进行命名，如广告：ad.js，菜单：menu.js；弹框：pop.js

**4. CSS文件命名**

根据文件类型及模块进行命名，如side.css，menu.css，style.css

**Html编码规范：**

**1. 标签**

自闭合（self-closing）标签，无需闭合( 例如：img,input,br,hr等 )；可选的闭合标签（closing tag），需闭合( 例如：</li>或</body>)；

标签中不允许出现样式，必须用class来声明样式，如：

<input type=”text” id=”idName” style=”text-align:right;”/> 错误

<input type=”text” id=”idName” class=”class-name”/> 正确

**语义化标签：**

p,h1,h2,h3,h4,h5,h6,ul,li,ol,strong,em,i,abbr,code,blcokquote

**Html5语义化标签：**

header, nav, section, article, aside, footer，figure

**已废除的标签：**

s，i，b，font

**2. 文档类型**

**使用html5文档类型**

<!DOCTYPE html>

**3. 语言属性**

<!-- 中文 -->

<html lang="zh-cn"> // zh\_cn已被淘汰

<html lang="zh-cmn-Hans"> <!-- 标准的 lang 属性写法 -->

**4. 字符编码**

<meta charset="utf-8">

**5. IE、Chome兼容模式**

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">

**6. SEO优化**

<title>标题</title>

<meta name="description" content="不超过150个字符" /> <!-- 页面描述 -->

<meta name="keywords" content=""/> <!-- 页面关键词 -->

<meta name="author" content="name, email@gmail.com" /> <!-- 网页作者 -->

<meta name="robots" content="index,follow" /> <!-- 搜索引擎抓取 -->

<link rel="alternate" type="application/rss+xml" title="RSS" href="/rss.xml" /> <!-- 添加 RSS 订阅 -->

<meta name="google" value="notranslate"> <!-- 禁止Chrome浏览器中自动提示翻译 -->

**7. Meta相关**

<!-- 关闭电话号码识别 -->

<meta name="format-detection" content="telephone=no" />

<!-- 关闭邮箱地址识别： -->

<meta name="format-detection" content="email=no" />

<!-- 指定 iOS 的 safari 顶端状态条的样式， 可选default、black、black-translucent -->

<meta name="apple-mobile-web-app-status-bar-style" content="black" />

**8. viewport标准定义**

1. viewport: 一般指的是浏览器窗口内容区的大小，不包含工具条、选项卡等内容；
2. width: 浏览器宽度，输出设备中的页面可见区域宽度；
3. device-width: 设备分辨率宽度，输出设备的屏幕可见宽度；
4. initial-scale: 初始缩放比例；
5. maximum-scale: 最大缩放比例；
6. minimum-scale: 最小缩放比例；
7. user-scalable: 是否允许用户进行缩放，值为"no"或"yes", no 代表不允许，yes代表允许；

例：页面窗口自动调整到设备宽度，并禁止用户缩放页面

<meta name="viewport" content="width=device-width, initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-scalable=no" />

**9. IOS图标**

<!-- iOS 设备 begin -->

<!-- 添加到主屏后的标题（iOS 6 新增） -->

<meta name="apple-mobile-web-app-title" content="标题">

<!-- 是否启用 WebApp 全屏模式 -->

<meta name="apple-mobile-web-app-capable" content="yes" />

<!-- 设置状态栏的背景颜色，只有在 `"apple-mobile-web-app-capable" content="yes"` 时生效 -->

<meta name="apple-mobile-web-app-status-bar-style" content="black-translucent" />

<!-- 禁止数字识自动别为电话号码 -->

<meta name="format-detection" content="telephone=no" />

<!-- 添加智能 App 广告条 Smart App Banner（iOS 6+ Safari） -->

<meta name="apple-itunes-app" content="app-id=myAppStoreID, affiliate-data=myAffiliateData, app-argument=myURL">

<!-- iOS 图标 begin -->

<!-- iPhone 和 iTouch，默认 57x57 像素，必须有 -->

<link rel="apple-touch-icon-precomposed" href="/apple-touch-icon-57x57-precomposed.png" />

<!-- Retina iPhone 和 Retina iTouch，114x114 像素，可以没有，但推荐有 -->

<link rel="apple-touch-icon-precomposed" sizes="114x114" href="/apple-touch-icon-114x114-precomposed.png" />

<!-- Retina iPad，144x144 像素，可以没有，但推荐有 -->

<link rel="apple-touch-icon-precomposed" sizes="144x144" href="/apple-touch-icon-144x144-precomposed.png" />

<!-- iOS 图标 end -->   
<!-- iOS 启动画面 begin -->

<!-- iPad 竖屏 768 x 1004（标准分辨率） -->

<link rel="apple-touch-startup-image" sizes="768x1004" href="/splash-screen-768x1004.png" />

<!-- iPad 竖屏 1536x2008（Retina） -->

<link rel="apple-touch-startup-image" sizes="1536x2008" href="/splash-screen-1536x2008.png" />

<!-- iPad 横屏 1024x748（标准分辨率） -->

<link rel="apple-touch-startup-image" sizes="1024x748" href="/Default-Portrait-1024x748.png" />

<!-- iPad 横屏 2048x1496（Retina） -->

<link rel="apple-touch-startup-image" sizes="2048x1496" href="/splash-screen-2048x1496.png" />

<!-- iPhone/iPod Touch 竖屏 320x480 (标准分辨率) -->

<link rel="apple-touch-startup-image" href="/splash-screen-320x480.png" />

<!-- iPhone/iPod Touch 竖屏 640x960 (Retina) -->

<link rel="apple-touch-startup-image" sizes="640x960" href="/splash-screen-640x960.png" />

<!-- iPhone 5/iPod Touch 5 竖屏 640x1136 (Retina) -->

<link rel="apple-touch-startup-image" sizes="640x1136" href="/splash-screen-640x1136.png" />

<!-- iOS 启动画面 end -->

<!-- iOS 设备 end -->

**10. favicon**

1. 在 Web Server 根目录放置 favicon.ico 文件；
2. 使用 link 指定 favicon；
3. favicon不要使用.png,jpg等图片格式，使用.ico图标格式，大小32\*32或64\*64；

<link rel="shortcut icon" href="path/to/favicon.ico">

**11. DNS预解析**

<meta http-equiv="x-dns-prefetch-control" content=“on">

  <link rel=“dns-prefetch” href=“”>

**12. ID和Class**

同一个ID在页面中只能有一个，可有多个class；

**13. 属性**

编写Html代码时，属性应该按照特定的顺序出现以保证易读性，从前往后：id-class-name-data-\*-src,for,type,href-title,alt-aria-xxx,role；

如：<a id="..." class="..." data-modal="toggle" href="###"></a>

属性引用时，使用双引号，不要使用单引号；

属性值必须有值，且采用驼峰命名；

自定义属性应遵循 " data-\* "规则；

**14. 元素**

所有元素必须小写；

元素需成对出现，特殊如<br /><hr /><img /><input />除外；

**严格嵌套约束在所有的浏览器下都不被允许；而语义嵌套约束，浏览器大多会容错处理，生成的文档树可能相互不太一样。**

**语义嵌套约束：**

<li> 用于 <ul> 或 <ol> 下；

<dd>, <dt> 用于 <dl> 下；

<thead>, <tbody>, <tfoot>, <tr>, <td> 用于 <table> 下；

**严格嵌套约束：**

<a>里不可以嵌套交互式元素<a>、<button>、<select>等;

<p>里不可以嵌套块级元素<div>、<h1>~<h6>、<p>、<ul>/<ol>/<li>、<dl>/<dt>/<dd>、<form>等。

**15. 布尔值属性**

在HTML5 规范中， disabled、checked、selected 等特殊属性默认不用设置值；

**16. 注释**

注释格式 ,<!--注释内容-->, '--'只能在注释的始末位置,不可置入注释文字区域；

**CSS编码规范：**

**1. 代码组织**

1. 以组件为单位组织代码段；
2. 制定一致的注释规范； /\* 注释内容 \*/
3. 组件块和子组件块以及声明块之间使用一空行分隔，子组件块之间三空行分隔；
4. 如果使用了多个 CSS 文件，将其按照组件而非页面的形式分拆，因为页面会被重组，而组件只会被移动；

**2. 声明块格式**

1. 选择器分组时，保持独立的选择器占用一行；
2. 声明块的左括号 { 前添加一个空格；
3. 声明块的右括号 } 应单独成行；
4. 声明语句中的 : 后应添加一个空格；
5. 声明语句应以分号 ; 结尾；
6. 一般以逗号分隔的属性值，每个逗号后应添加一个空格；
7. rgb()、rgba()、hsl()、hsla() 或 rect() 括号内的值，逗号分隔，但逗号后不添加一个空格；
8. 对于属性值或颜色参数，省略小于 1 的小数前面的 0 （例如，.5 代替 0.5；-.5px 代替 -0.5px）；
9. 十六进制值应该全部小写和尽量简写，例如，#fff 代替 #ffffff；
10. 避免为 0 值指定单位，例如，用 margin: 0; 代替 margin: 0px;；
11. **ID和Class**
12. 使用语义化、通用的命名方式；
13. 使用连字符 - 作为 ID、Class 名称界定符，不要驼峰命名法和下划线；
14. 避免选择器嵌套层级过多，尽量少于 3 级；
15. 避免选择器和 Class、ID 叠加使用；
16. 元素尽量使用class选择器匹配，对于特殊功能可以考虑使用 ID选择器；
17. 明确各选择器的优先级，作用范围；优先级是根据权重来进行定义的，标签的权重为1，class的权重为10，id的权重为100；

**4. 引号使用**

url() 、属性选择符、属性值使用双引号；

**4. 媒体查询**

将媒体查询放在尽可能相关规则的附近。不要将他们打包放在一个单一样式文件中或者放在文档底部。

如：@media (max-width: 768px) { }

@media screen and (min-width:960px) { }

@media screen and (min-width:960px) and (max-width:1200px){ }

**5. CSS引入方式**

1. 链接式： 在网页的<head></head>标签对中使用<link>标记来引入外部样式表文件，使用html规则引入外部css （推荐此方式引用）；
2. 导入式： 将一个独立的.css文件引入HTML文件中，导入式@import使用CSS规则引入外部CSS文件，<style>标记也是写在<head>标记中；
3. 内联css文件，直接在header 里面写css；
4. 直接在标签里面写样式；

link是html加载前就引用，而import是html加载后才引用，采用import引用，会先显示无样式的页面，然后再把样式放进去。如果用JavaScript动态引用CSS，得使用link引用方式才可实现

与 <link> 相比，@import 要慢很多，不光增加额外的请求数，还会导致不可预料的问题，禁止使用

内联和直接写在标签里样式，不建议使用

**6. CSS编写规范**

1. CSS写完之后必须使用“;”号结尾，虽然浏览器支持不用分号结尾，但是不建议这么做**；**
2. 除非编写HTML5页面，否则不允许使用CSS3伪类，如nth-child(2n)，:first-child等伪类**；**
3. 编写CSS样式时，不允许换行，样式必须一行写完，自动换行 **;**
4. 代码缩进与格式: 建议单行书写,
5. 对于子元素样式采用链选择器进行选择,如：btn-save div， btn-save >div；
6. css属性书写顺序, 建议遵循:  布局定位属性-->自身属性-->文本属性-->其他属性: 此条可根据自身习惯书写, 但尽量保证同类属 性写在一起；
7. 样式表中中文字体名, 请务必转码成unicode码, 以避免编码错误时乱码；
8. 减少使用影响性能的属性, 比如position:absolute || float，建议使用CSS3新特性，如flex，grid等布局方式；
9. 杜绝使用<meta http-equiv="X-UA-Compatible" content="IE=7" /> 兼容 ie8;
10. 必须为大区块样式添加注释, 小区块适量注释;
11. 背景图片请尽可能使用 sprite 技术, 减小 http 请求,
12. 使用 table 标签时(尽量避免使用 table 标签),
13. 请不要用 width/ height/cellspacing/cellpadding 等 table 属性直接定义表现 , 应尽可能的利用 table 自身私有属性分离结构与表现 , 如 thead,tr,th,td,tbody,tfoot,colgroup,scope;

**7. A链接顺序**

a:link -> a:visited -> a:hover -> a:active

**8. 优雅降级与线性增强**

1. **渐进增强（Progressive Enhancement）**：一开始就针对低版本浏览器进行构建页面，完成基本的功能，然后再针对高级浏览器进行效果、交互、追加功能达到更好的体验。
2. **优雅降级（Graceful Degradation）**：一开始就构建站点的完整功能，然后针对浏览器测试和修复。比如一开始使用 CSS3 的特性构建了一个应用，然后逐步针对各大浏览器进行 hack 使其可以在低版本浏览器上正常浏览。

/\*渐进增强写法\*/

.transition {

-webkit-transition: all .5s;

-moz-transition: all .5s;

-o-transition: all .5s;

transition: all .5s;

}

/\*优雅降级写法\*/

.transition {

transition: all .5s;

-o-transition: all .5s;

-moz-transition: all .5s;

-webkit-transition: all .5s;

}

**9. 网页布局的五种方式**

1. **浮动布局** (float)
2. **定位布局**（relative， absolute， fixed）
3. **弹性布局**（Flex）
4. **表格布局**（Table）
5. **网格布局**（Grid）

浮动布局和定位布局建议尽量少用，推荐使用弹性、表格（此表格非传统表格）、网络这三种CSS3布局方式；

**10. 注释**

单行注释：

内容区 /\* 这儿是单行注释 \*/

多行注释：

/\* Start 模块名及说明 \*/

内容区

/\* End 模块名及说明 \*/

**11. CSS命名规则**

|  |  |  |
| --- | --- | --- |
| 头：header | 内容content/container | 尾：footer |
| 页面主体：main | 下载：download | 子导航：subnav |
| 导航：nav | 侧栏：sidebar | 栏目：column |
| 外围布局宽度：wrapper | 左右中left right center | 登录条：loginbar |
| 广告：banner | 热点：hot | 新闻：news |
| 标志：logo | 搜索：search | 版权：copyright |
| 菜单：menu | 子菜单：submenu | 滚动：scroll |
| 页脚：footer | 友情链接：friendlink | 内容：content |
| 标签页：tab | 文章列表：list | 提示信息：msg |
| 小技巧：tips | 指南：guild | 状态：status |
| 标题：title | 加入：joinus | 服务：service |
| 注册：regsiter | 合作伙伴：partner | 投票：vote |
| 页面主体：main | 主导航：mainbav | 子导航：subnav |
| 顶导航：topnav | 边导航：sidebar | 摘要: summary |
| 左导航：leftsidebar | 右导航：rightsidebar | 登陆：login |

**附录：**

/\***解决li中的2px Bug**\*/

1. vertical-align:bottom;

2. float:left;

/\***解决中英文行高Bug**\*/

font-family:simsun;

/\***解决IE6设置最小/最大行高行高**\*/

height:auto !important;

height:640px;

min-height:640px; //max-height:640px;

/\***清除谷歌，Firefox输入框边线**\*/

outline: none;

/\***清除textarea自动伸缩**\*/

resize: none;

/\***设置字间距**\*/

letter-spacing: 10px;

/\***CSS设置网页图片透明度**\*/

filter:alpha(opacity=50);      /\* IE \*/

-moz-opacity:0.5;           /\* Moz + FF \*/

-khtml-opacity: 0.5;         /\* 兼容一些老版本的Safari浏览器\*/

opacity: 0.5;              /\* 支持CSS3的浏览器（FF 1.5也支持）\*/

/\***CSS设置边框（圆，椭圆效果）**\*/

-webkit-border-radius:24px;

-moz-border-radius:24px;

border-radius:24px;

/\***图层阴影效果**\*/

-webkit-box-shadow: 3px 3px 3px;

-moz-box-shadow: 3px 3px 3px;

box-shadow: 3px 3px 3px;

/\***背景颜色透明度**\*/

background:#000;

filter:alpha(opacity=50);

-moz-opacity:0.5;

-khtml-opacity:0.5;

opacity:0.5;

/\***a标签无效**\*/

javascript:void(0);

/\***文本设置阴影**\*/

text-shadow ： none | <length> none | [<shadow>, ] \* <shadow> 或none | <color> [, <color> ]\*

也就是：

text-shadow:[颜色(Color)  x轴(X Offset) y轴(Y Offset) 模糊半径(Blur)],[颜色(color) x轴(X Offset) y轴(Y Offset) 模糊半径(Blur)]...

或者

text-shadow:[x轴(X Offset) y轴(Y Offset)  模糊半径(Blur)  颜色(Color)],[x轴(X Offset) y轴(Y Offset)  模糊半径(Blur)  颜色(Color)]...

/\***文本超出部分显示**\*/

white-space: nowrap;

overflow: hidden;

text-overflow: ellipsis;

/\***设置Div自适应高度/宽度**\*/

.div { height: calc(100% - 98px); height: -webkit-calc(100%-98px); height: -moz-calc(100%-98px); }

.div { width: calc(100% - 98px); width: -webkit-calc(100%-98px); width: -moz-calc(100%-98px); }

/\***IE6不支持position:fixed的解决方法**\*/

/\*让position:fixed在IE6下可用! \*/

.fixed-top                      /\* 头部固定 \*/{position:fixed;bottom:auto;top:0px;}

.fixed-bottom                     /\* 底部固定 \*/{position:fixed;bottom:0px;top:auto;}

.fixed-left                      /\* 左侧固定 \*/{position:fixed;right:auto;left:0px;}

.fixed-right                     /\* 右侧固定 \*/{position:fixed;right:0px;left:auto;}

/\* 上面的是除了IE6的主流浏览器通用的方法 \*/

\* html,\* html body             /\* 修正IE6振动bug \*/{background-image:url(about:blank);background-attachment:fixed;}

\* html .fixed-top                /\* IE6 头部固定 \*/{position:absolute;bottom:auto;top:expression(eval(document.documentElement.scrollTop));}

\* html .fixed-right               /\* IE6 右侧固定 \*/ {position:absolute;right:auto;left:expression(eval(document.documentElement.scrollLeft+document.documentElement.clientWidth-this.offsetWidth)-(parseInt(this.currentStyle.marginLeft,10)||0)-(parseInt(this.currentStyle.marginRight,10)||0));}

\* html .fixed-bottom         /\* IE6 底部固定  \*/

{position:absolute;bottom:auto;top:expression(eval(document.documentElement.scrollTop+document.documentElement.clientHeight-this.offsetHeight-(parseInt(this.currentStyle.marginTop,10)||0)-(parseInt(this.currentStyle.marginBottom,10)||0)));}

\* html .fixed-left               /\* IE6 左侧固定 \*/{position:absolute;right:auto;left:expression(eval(document.documentElement.scrollLeft));}

**JavaScript编码规范：**

**1. 注释**

**原则**

1. As short as possible（如无必要，勿增注释）：尽量提高代码本身的清晰性、可读性。
2. As long as necessary（如有必要，尽量详尽）：合理的注释、空行排版等，可以让代码更易阅读、更具美感。

**单行注释**

必须独占一行。// 后跟一个空格，缩进与下一行被注释说明的代码一致。

**多行注释**

避免使用 /\*...\*/ 这样的多行注释。有多行注释内容时，使用多个单行注释。

**函数/方法注释**

1. 函数/方法注释必须包含函数说明，有参数和返回值时必须使用注释标识。；
2. 参数和返回值注释必须包含类型信息和说明；
3. 当函数是内部函数，外部不可访问时，可以使用 @inner 标识；

/\*\*

\* 函数描述 \*

\* @param {string} p1 参数1的说明

\* @param {string} p2 参数2的说明，比较长

\* @param {number=} p3 参数3的说明（可选）

\* @return {Object} 返回值描述

\*/

**文件注释**

文件注释用于告诉不熟悉这段代码的读者这个文件中包含哪些东西。 应该提供文件的大体内容, 它的作者, 依赖关系和兼容性信息。如下:

/\*\*

\* @fileoverview Description of file, its uses and information

\* about its dependencies.

\* @author user@meizu.com (Firstname Lastname)

\* Copyright 2009 Meizu Inc. All Rights Reserved.

\*/

**2. 命名**

**变量**, 使用 驼峰命名法；

var loadingModules = {};

**私有属性、变量和方法**以下划线 \_ 开头；

var \_privateMethod = {};

**常量**, 使用全部字母大写，单词间下划线分隔的命名方式；

var HTML\_ENTITY = {};

**函数,** 使用驼峰命名法；

**函数的参数**, 使用驼峰命名法；

function hear(theBells) {}

**类**, 使用驼峰命名法

**类的方法/属性,** 使用驼峰命名法；

枚举变量 使用帕斯卡命名法；

枚举的属性， 使用全部字母大写，单词间下划线分隔的命名方式； （帕斯卡命名法指当变量名和函式名称是由二个或二个以上单字连结在一起，而构成的唯一识别字）

var TargetState = {

READING: 1,

READED: 2,

APPLIED: 3,

READY: 4

};

由多个单词组成的 **缩写词**，在命名中，根据当前命名法和出现的位置，所有字母的大小写与首字母的大小写保持一致。

function XMLParser() { }

function insertHTML(element, html) { }

var httpRequest = new HTTPRequest();

**3. 命名语法**

**类名**，使用名词。

function Engine(options) {}

**函数名**，使用动宾短语。

function getStyle(element) {}

**boolean** 类型的变量使用 is 或 has 开头。

var isReady = false;var hasMoreCommands = false;

**Promise 对象**用动宾短语的进行时表达。

var loadingData = ajax.get('url'); loadingData.then(callback);

**4. True 和 False 布尔表达式**

类型检测优先使用 typeof。对象类型检测使用 instanceof。null 或 undefined 的检测使用 == null。

下面的布尔表达式都返回 false:

1. null
2. undefined
3. '' 空字符串
4. 0 数字0

但小心下面的, 可都返回 true:

1. '0' 字符串0
2. [] 空数组
3. {} 空对象

**5. Array循环**

Array的循环方式有五种：for循环，for-in 循环，forEach， map遍历，for-of遍历（ES6新增功能）

for-in 循环只用于 object/map/hash 的遍历, 对 Array 用 for-in 循环有时会出错. 因为它并不是从 0 到 length - 1 进行遍历, 而是所有出现在对象及其原型链的键值。

**7. 二元和三元操作符**

操作符始终写在前一行, 以免**分号的隐式插入**产生预想不到的问题。

var x = a ? b : c;

var y = a ? longButSimpleOperandB : longButSimpleOperandC;

.操作符也是如此：

var x = foo.bar().doSomething().doSomethingElse();

**8. 条件(三元)操作符 (?:)**

三元操作符用于替代 if 条件判断语句。

// Not recommended

if (val != 0) {

return foo();

} else {

return bar();

}

// Recommended

return val ? foo() : bar();

**9. && 和 ||**

二元布尔操作符是可短路的, 只有在必要时才会计算到最后一项；// Recommended

function foo(opt\_win) {

var win = opt\_win || window; // ...

}

var kid = node && node.kids && node.kids[index];

if (kid) {

foo(kid);

}

**10. JS编写规范**

1. JS须使用外部文件方式加载，建议放在页尾，Html页面中不建议嵌入JS代码；
2. 每一个函数与事件监听都应有注释，声明其作用，如果代码过长，那么请对一个功能模块进行注释；
3. JS代码换行时，必须使用缩进；
4. 不要污染全局变量，尽量采用局部变量；
5. 采用链式调用，可优化代码结构及提供性能；
6. 规范定义JSON对象，补全双引号；
7. 用“＝＝＝“取代“＝＝“；
8. 统一使用缩进大小，以tab为基准；
9. var 的时候若有常量，全用大写；
10. js中避免过多的DOM操作，需创建文档片段；
11. js代码要做到高内聚/低耦合；
12. 定义多个函数的时候，如果参数相同，可以用apply()和call()改变环境对象，得到引用；
13. 不要使用eval()函数；
14. 提倡对象字面量；
15. 建议多使用innerHtml；
16. 一次性修改样式属性；
17. 用数组方式来遍历Dom对象集合；
18. 不要在函数体内做过多的嵌套判断；
19. 适当用while代替for循环，可以提高性能；
20. 操作DOM前，先把DOM节点删除或隐藏，因为隐藏的节点不会触发重排；
21. 不要使用JS关键字和保留字定义变量；

**11. JQuery编写规范**

1. 使用jQuery选择器如果是唯一的，请使用ID选择器；
2. 用class选择器时，在class前加上标签名，如： $(“div:class”)正确；
3. 尽量使用ID选择器代替class选择器；
4. 使用驼峰命名法命名变量；
5. 如果一个变量存放的是jQuery对象的话，那么请用”$”符号 开头，声明这是一个jQuery对象；
6. 避免使用live()，bind()函数绑定事件，可以使用on()代替；
7. 当要操作 DOM 元素的时候，尽量将其分离节点，操作结束后，再插入节点；
8. 使用字符串连接或 array.join 要比 .append()性能更好；
9. 尽可能的使用 ID 选择器，因为它会调用浏览器原生方法 document.getElementById 查找元素。当然直接使用原生 document.getElementById 方法性能会更好；
10. 如果需要，对事件使用自定义的 namespace，这样容易解绑特定的事件，而不会影响到此 DOM 元素的其他事件监听；
11. 对 Ajax 加载的 DOM 元素绑定事件时尽量使用事件委托。事件委托允许在父元素绑定事件，子代元素可以响应事件，也包括 Ajax 加载后添加的子代元素；
12. 尽量使用链式写法而不是用变量缓存或者多次调用选择器方法；
13. 当链式写法超过三次或者因为事件绑定变得复杂后，使用换行和缩进保持代码可读性；
14. 多个参数使用对象字面量存储；
15. 不要将 CSS 写在 jQuery 里面；
16. 正则表达式仅准用 .test() 和 .exec() 。不准用 "string".match() ；

**12. JS性能优化**

1. 避免不必要的 DOM 操作，浏览器遍历 DOM 元素的代价是昂贵的。最简单优化 DOM 树查询的方案是，当一个元素出现多次时，将它保存在一个变量中，就避免多次查询 DOM 树了；
2. 循环无疑是和 JavaScript 性能非常相关的一部分，通过存储数组的长度，可以有效避免每次循环重新计算；
3. 异步加载第三方内容；
4. 避免使用 jQuery 实现动画，禁止使用 slideUp/Down() fadeIn/fadeOut() 等方法，尽量不使用 animate() 方法；

**13. 关键字**

|  |  |  |  |
| --- | --- | --- | --- |
| break | do | instanceof | typeof |
| case | else | new | var |
| catch | finally | return | void |
| continue | for | switch | while |
| debugger | function | this | with |
| default | if | throw | try |
| delete | in |  |  |

**14. 保留字**

|  |  |  |  |
| --- | --- | --- | --- |
| abstract | enum（枚举） | int | short |
| boolean | export | interface | static |
| byte | extends | long | super |
| char | final | native | synchronized |
| class | float | package | throws |
| const | goto | private | transient |
| double | implements | protected | volatile |
| yield | import | public | let |

**参考**

[编码规范 by @mdo](http://codeguide.bootcss.com/)

[Baidu 前端规范](https://github.com/ecomfe/spec)

[Qunar 前端规范](https://github.com/doyoe/html-css-guide)

[常用的 HTML 头部标签](https://github.com/yisibl/blog/issues/1)

[Google Style Guide](http://docs.kissyui.com/1.4/docs/html/tutorials/style-guide/google-js-style.html)

[Amazeui Style Guide](http://amazeui.org/getting-started/)

[CSS Guidelines](http://cssguidelin.es/)

[20 Snippets You should be using from Html5 Boilerplate](http://www.1stwebdesigner.com/snippets-html5-boilerplate/)

[谈谈CSS性能](http://www.w3.org/2015/Talks/0109-CSSConf-xq/)

[Web动画性能指南](http://alexorz.github.io/animation-performance-guide/)

[Why Moving Elements With Translate() Is Better Than Pos:abs Top/left](http://www.paulirish.com/2012/why-moving-elements-with-translate-is-better-than-posabs-topleft/)

[High Performance Animations](http://www.html5rocks.com/zh/tutorials/speed/high-performance-animations/)

[jQuery Standards](http://lab.abhinayrathore.com/jquery-standards/)

[Learn jQuery](http://learn.jquery.com/about-jquery/)

[无线Web开发经验谈](http://am-team.github.io/amg/dev-exp-doc.html)

[移动端前端笔记大全](http://segmentfault.com/a/1190000002581619)