前端编码规范

一 、文件管理

     1. 文件目录：

               ：root（根目录）

具体模块（ad-service）

 dep （引用的库、插件）

                               js

                               css

                          src （源文件）（注意：css和js文件用对应模块前缀区分开来）

                              js

common.js

…

                              css

reset.css

common.css

ad-common.css

media-common.css

…

                               img

        2. 文件命名：

        · 使用英文名，全部小写，以 “ - ” 来分隔 ( 不要用汉字和数字 )。

例：

wom-video-resource-list.html

wom-video-resource-list.css

wom-video-resource-list.js

二、html标准：

       1. 考虑浏览器兼容性，不建议使用H5新标签，如：<header></header>、<nav></nav>等。但是class类最好命名成语义型如：class=“header”   class=“nav”。

       2. 嵌套元素应当缩进一次，用四个空格。

       3. 对于属性的定义，确保全部使用双引号，绝不要使用单引号。

       4. 不要省略可选的结束标签。（例如，</li>或 </body>）。

       5. 任何时候都要尽量使用最少的标签并保持最小的复杂度，避免冗余的父元素。例如下面这种情况，span标签完全可以去掉：

  <div>

                  <span>

                                 <img src="" alt="">

                    </span>

           </div>

6. 属性顺序

class

           id, name

           data-\*

           src, for, type, href

           title, alt

7. 通过 JavaScript 生成的标签让内容变得不易查找、编辑，并且降低性能。能避免时尽量避免。

三、css标准：

     1. 编码统一为utf-8  css文件头部加    @charset “utf-8";

     2. 命名：

· css的命名一定要有意义，使用有组织的或目的明确的名称，不要使用表现形式的名称。要针对该元素在页面上结构功能命名。

          （如 #mainContainer，.sidebar 这样，而不是根绝它表现出来的样式来命名，如 .red, .blue）

        · class命名只能出现小写字母和破折号 “ - ” （不是下划线，也不是驼峰命名法）。

· id命名建议使用驼峰法，专门留给JS用，这样才符合表现与行为分离的原则。

        · 名称尽可能短（但不能过度简写），并且意义明确。(例：.btn 代表 button，但是 .s不能表达任何意思。)

· 加前缀（前缀类似于命名空间）,（例：.header .header-nav .header-nav-help ,   .btn 和.btn-danger  ）。

        · 命名优先级：

                           模块   （ .header   .content   .footer等）

                           功能   （ .header-nav-help ）

                           方位   （ .header-nav-help-left ）

     3. 选择器

          · 为选择器分组时，将单独的选择器单独放在一行。

         .nav,

           .content-media,

           .footer {

                     padding: 15px;

                     border: 1px solid #ccc;

           }

        · 选择器要尽可能短，并且尽量限制组成选择器的元素个数，建议不要超过 3 。

        · 只有在必要的时候才将 class 限制在最近的父元素内（也就是后代选择器）（例如，不使用带前缀的 class 时 -- 前缀类似于命名空间）。

          例如：

               html结构:

                    <div class="header">

                           <div class="header-nav"></div>

                           <div class="header-logo"></div>

                           <div class=“header-search">

                    <input class="text" type="text">

                    <input class="btn" type=“button">

</div>

</div>

               css样式:

               .header{}

                .header-nav{}

                .header-logo{}

                .header-search{ }

                .header-search .text{}

                .header-search .btn{}

        4. 重置样式表：reset.css（讨论一份）

           盒模型：border-box  还是 content-box

           字体大小：font-12, font-14, font-18, font-20

           字体颜色： color - 000  , color - fff

           背景颜色： bg - 000 , bg - fff

…

        5. 公用样式:

          开发前分析哪些部分样式是公用的, 写成公用 common.css，单独引用，放在文件上面（ js同样 ）。如有少许变动，可在之后的样式表中更改。

        6. 关于清除浮动：

          最好不要添加冗余的dom结构，只需給父级添加类名 class=“clearfix”。

          .clearfix{zoom:1}

          .clearfix:after {content:“”;clear:both;display:table;}

7. 不要将样式写成单行，降低可读性。 如：.hotel-content {margin: 10px; background-color: #efefef;}。

8. 最好不使用行内（inline）样式。 如：<div style=“width:100px;height:50px;”></div>

9. 响应式布局代码应该与其相应代码写在一块，不要写在一块放在尾部。

四、js标准 ：

       1. 脚本加载：放在尾部，考虑老旧浏览器，最好采用异步（async），防止因js的加载阻塞DOM结构。

        <html>

               <head>

                      <link rel="stylesheet"  href="main.css">

               </head>

               <body>

                       <script src="main.js"  async></script>

              </body>

         </html>

        2. 函数命名：驼峰命名法，最好用动词，以fn开头。(见附录二)

例如：function  fnGetName( ){

}

        3. 创建独立隔绝的定义域，防止全局命名空间被污染，还可确保你的代码不会轻易被其它全局命名空间里的代码所修改。最好用 var 声明变量

                 (function()

                         // Code goes here

                 }( ));

               $(function( ) {

               });

         4. 引入JS库文件, 文件名须包含库名称、版本号、是否为压缩版, 比如jquery-1.4.1.min.js;

5. jquery的事件格式，最好写成：$(“this”).on(“click",function(){ }),这是jquery的最终执行方式，执行速度最快。

6. 当从form表单读取input、textarea等输入时，需要 使用 $.trim 来去掉前后空格，这个细节很多人一般不大会注意。

五、注释的写法

· 注释起始和结束位置。

· 结构性注释尽可能简短（html），功能性注释要详细（js）。

/\*~~~~start~~~~\*/

     内容区

/\* ~~~~End ~~~ \*/

规范示例：

      <!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="UTF-8">

    <meta http-equiv="X-UA-Compatible" content="IE=Edge">

<meta name="keywords" content="谦玛网络，谦玛互动，沃米优选，上海谦玛，谦玛广告">

<meta name="description" content="微信公众号推广,微信朋友圈推广，微博推广,软文发稿,媒体推广,自媒体营销,广告投放,网络广告,APP广告">

<title>Document</title>

<link rel="shortcut icon" href="images/icon.png">

<link rel="stylesheet" href="../css/reset.css">

<link rel="stylesheet" href="../css/index.css">

</head>

<body>

    <script src="main.js"></script>

</body>

</html>

六、其他标准：

1. 为了更好兼容IE浏览器。头部这样写：<meta http-equiv="X-UA-Compatible" content="IE=Edge">

Edge 模式通知 IE浏览器以最高级别的可用模式显示内容。

2. 图片处理

· 图片压缩，提高网页加载速度。一般情况下，图片应该切成带透明背景（PNG8），并裁减成紧密贴合图片外边框。

· 合并 CSS图片，节省 HTTP 请求数（css sprites），但不能过度使用，会降低网站的可访问性，例如传达重要信息的图片。

· 对于搜索引擎来说，蜘蛛是不认识图片的，它们只能通过alt标签来判断图片内容。图片的”alt”属性具有注释作用，这样对于搜索引擎来说的话比较容易识别，有助于网站排名优化。（关键图片设置alt）

· 命名全部用小写英文字母 、数字 、 “ \_ ”的组合，其中不得包含汉字、空格、特殊字符；尽量用易懂的词汇, 便于团队其他成员理解。如ad\_left01.png 、btn\_submit.png。

七、关于响应式页面、UI设计

1. 沃米优选平台有自己的移动端页面设计，暂时不考虑做响应式。

2.UI视觉设计稿以后按照1600px 设计，内容部分以1200px为准。

附一：常用css命名

头：header

内容：content/container

尾：footer

导航：nav

侧栏：sidebar

栏目：column

页面外围控制整体布局宽度：wrapper

左右中：left right center

登录条：loginbar

标志：logo

广告：banner

页面主体：main

热点：hot

新闻：news

下载：download

子导航：subnav

菜单：menu

子菜单：submenu

搜索：search

友情链接：friendlink

页脚：footer

版权：copyright

滚动：scroll

内容：content

标签页：tab

文章列表：list

提示信息：msg

小技巧：tips

栏目标题：title

加入：joinus

指南：guide

服务：service

注册：register

状态：status

投票：vote

合作伙伴：partner

(1) 页面结构

容器: container

页头：header

内容：content/container

页面主体：main

页尾：footer

导航：nav

侧栏：sidebar

栏目：column

页面外围控制整体布局宽度：wrapper

左右中：left right center

(2) 导航

导航：nav

主导航：mainbav

子导航：subnav

顶导航：topnav

边导航：sidebar

左导航：leftsidebar

右导航：rightsidebar

菜单：menu

子菜单：submenu

标题: title

(3) 功能

标志：logo

广告：banner

登陆：login

登录条：loginbar

注册：register

搜索：search

功能区：shop（如购物车）

标题：title

加入：joinus

状态：status

按钮：btn

滚动：scroll

标签页：tab

文章列表：list

提示信息：msg

当前的: current

小技巧：tips

图标: icon

注释：note

指南：guide

服务：service

热点：hot

新闻：news

下载：download

投票：vote

合作伙伴：partner

友情链接：link

版权：copyright

附二：常用JS命名动词

get 获取/set 设置,

add 增加/remove 删除

create 创建/destory 移除

start 启动/stop 停止

open 打开/close 关闭

read 读取/write 写入

load 载入/save 保存

create 创建/destroy 销毁

begin 开始/end 结束

backup 备份/restore 恢复

import 导入/export 导出

split 分割/merge 合并

inject 注入/extract 提取

attach 附着/detach 脱离

bind 绑定/separate 分离

view 查看/browse 浏览

edit 编辑/modify 修改

select 选取/mark 标记

copy 复制/paste 粘贴

undo 撤销/redo 重做

insert 插入/delete 移除

add 加入/append 添加

clean 清理/clear 清除

index 索引/sort 排序

find 查找/search 搜索

increase 增加/decrease 减少

play 播放/pause 暂停

launch 启动/run 运行

compile 编译/execute 执行

debug 调试/trace 跟踪

observe 观察/listen 监听

build 构建/publish 发布

input 输入/output 输出

encode 编码/decode 解码

encrypt 加密/decrypt 解密

compress 压缩/decompress 解压缩

pack 打包/unpack 解包

parse 解析/emit 生成

connect 连接/disconnect 断开

send 发送/receive 接收

download 下载/upload 上传

refresh 刷新/synchronize 同步

update 更新/revert 复原

lock 锁定/unlock 解锁

check out 签出/check in 签入

submit 提交/commit 交付

push 推/pull 拉

expand 展开/collapse 折叠

begin 起始/end 结束

start 开始/finish 完成

enter 进入/exit 退出

abort 放弃/quit 离开

obsolete 废弃/depreciate 废旧

collect 收集/aggregate 聚集