# web前端开发分享

## 1. web前端开发分享-css,js入门篇

关注前端这么多年，没有大的成就，就入门期间积累了不少技巧与心得，跟大家分享一下，不一定都适合每个人，毕竟人与人的教育背景与成长环境心理活动都有差别，但就别人的心得再结合自己的特点，然后探索适合自己的学习之路是比较好的。学习没有捷径，但学习是有技巧与方法。

一，css入门篇：

推荐书籍：css哪些事儿，精通css。

理由：css那些事儿，他是一本介绍css基础类的书，是入门的经典读物。

系统的介绍了css的选择符，伪类，工作环境，盒模型，两列，三列自适应布局。文字样式，图片处理，列表，表单，还有选项卡，相册，导航，新闻列表等其它大部分的实践。

精通css，这书已经出到第二版了，相比第一版，第二版的大部分目录结构保持不变，内容变得更丰满了。此书从一另一个视角介绍了css的博大精深。

二，js入门篇：

1, 建议阅读“javascript高级程序设计”第三版电子版。

经过前辈们的总结或本人的实践经历，自学的同学建议是这样，找本js入门的书和妙味视频结合看。一个主攻面试，一个主攻实践。

看书的目的主要是掌握理论知识，梳理js的知识结构，最主要的一点是为了应付面试。面试的主要环节是为了表达自己的想法，只凭简单的理解，是不能完全的表达出一个问题的广度或深度，只有结合了行业中的术语约定或一些习惯叫法，表达出来的意思才更专业。如果不是这样，表达出来的意思别说面试官听不懂，连自己都感觉要么表达不出来，要么表达的不是想要的。但书着重于对理论知识的阐述，缺少实际例子的支持，很难快速的消化。这时候视频可以弥补这个缺陷，而且妙味视频用简单的代码实现一些常见的效果，学习兴趣很快就能被吸引。但是对js的高级特性，个人感觉视频没有达到那个深度。而且js的一些基础的问题，或一个知识的扩充，限于各种条件的限制，可能无法完全表达出来。所以要结合书来补充。

一般学习的心态是：不急不躁，不快不慢。持之以恒，相信自己。

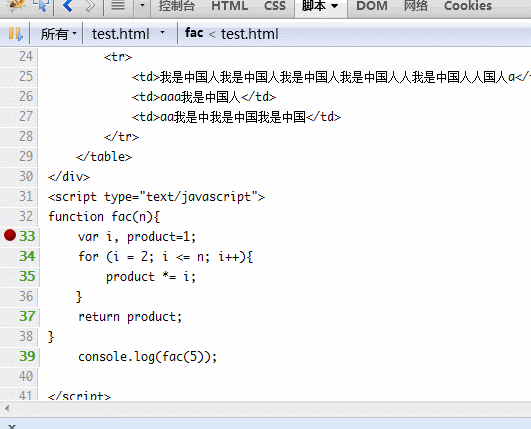
2, 建议欣赏，妙味及智能社视频资料。

学习主要技巧是动手及主动思考，视频在欣赏的同时记得跟敲代码不低于三遍，直到隔一天隔一周能重复敲出代码或回忆起思路为止。而且边敲边做笔记，以做复习时快速的查阅之用。学习的技巧是动手，学习的灵魂是坚持，学习的快乐就是分享。只要坚持，其实每天都在成长，只是感觉不明显而已。但这个月与前几月相比，已经有量的变化，量变引起质变。

另外一个建议：QQ群不是学习的主战场，群只是学习的辅助手段，要是想安心学习就是关QQ，断网，靠自己。

学习的态度需要务实，生活是残酷的，战场是血腥的，一份耕耘，一份收获，务实的学习好基础才能走的更远。

     敲出来的代码，思路不清晰时，可以在firebug,chrome里边打个断点跟一下，理理思路，理解代码的逻辑，这样影响才会深刻。初学者没办法都是这样，也只能这样。但是初期可能稍为慢一点，但一直不会这么慢，也不会想象的那么长，因为随着基本概念的掌握，编码技巧的熟悉，开发工具的熟练，消化视频的速度会越来越快。



妙味智能社视频：

<http://zhinengshe.com/video.html>

js视频

<http://bbs.miaov.com/forum.php?mod=viewthread&tid=7385>

<http://www.miaov.com/2013/miaovideo/miaovideo.html>

[知识点类] 妙味课堂

其它书籍总结分享：

<http://www.cnblogs.com/Henrya2/archive/2009/01/22/1380096.html>

js 书籍：

<http://www.cnblogs.com/Darren_code/archive/2011/06/29/javascript-learn.html>

js学习流程＋书籍

<http://limu.iteye.com/blog/1267475>

javascript那些书

<http://www.javascript100.com/?p=394>

如果学习js

<http://limu.iteye.com/blog/1267475>

javascript那些书

<http://www.yaohaixiao.com/?p=8>

写给想学JavaScript朋友的一点经验之谈

<http://www.zhihu.com/question/19713563>

如何循序渐进有效学习 JavaScript？

<http://www.zhihu.com/question/20246142>

前端其它资源

<http://www.w3cplus.com/page/books.html>

前端书籍分享

3. 等这些消化之后，阅读其它js书籍建议如下：

js dom编程艺术(2 version)->js高级程序设计 (3 version)-> js dom高级程序设计 -> js 语言精粹 -> 精通js -> 编写可维护的js -> 高性能js -> js面向对象编程指南  -> js模式 -> js权威指南(6 version) -> js语言精髓与编程实践

js dom编程艺术 : 他只是截取了js中的部分概念进行了介绍，并不完整或深入。

js高级程序设计 : 经典的js基础书，之所为经典是因为系统完善的介绍了js的历史，基础及预测了当下的流行趋势。与犀牛比更测重于实践结合

 js dom高级程序设计 : 主要讲如何去写一个框架或库。

js 语言精粹：json作者，道格拉斯写的，主要解析js语言本质或内涵。

精通js：jquery作者写的，浅浅的解析了jquery一些方法的实现，注意是很浅。

编写可维护的js，高性能js：这两本是高程作者尼古拉斯的，出版日期在高程后，深度可想而知。

js面向对象编程指南：主要讲面向对象的一些东西。

js模式： -> js权威指南(6 version)

js模式，这两本是淘宝团队翻译的，模式主要一半讲模式一半讲基础。  
js权威指南(6 version)，经典犀牛，一半讲基础，一讲是参考。

4. 编辑器推荐：

a.新手： sublime-text, <http://github.com/jikeytang/sublime-text>

b.熟手： webstorm: <http://note.youdao.com/share/?id=2d7ff03aeca64808fdc0f394ffb520bf&type=notebook>

推荐这两个的原因在于，有个技术叫emmet, http://docs.emmet.io，而这两个对他的支持是：

sublime text 是所有编辑器里边支持emmet比较好的的唯一一款。

webstorm(jetbrains系列产品，intellij idea, phpstorm, pycharm)是所有编辑器里边唯一内置emmet的一款，并且把emmet精神在往前推了一步的一款。

c. 其它：editplus, notepad++,vim.

准备多款的原因在于，没有最好的编辑器，只有最合适的，每个下面都有一款特色功能终究会吸引你。

## 2. web前端开发分享-css,js进阶篇

**一，css进阶篇：**

等css哪些事儿看了两三遍之后，需要对看过的知识综合应用，这时候需要大量的实践经验，

简单的想法：把qq首页全屏另存为jpg然后通过ps工具切图结合css转换成html，有无从下手的地方可以用firebug, chrome调试工具分析网站结构样式。如果技术熟练自信可以自己先写，写完之后在对比，以其来找寻自己的差距。结构是网站的骨架，如果写的不合理，将是bug产生的重要根源，所以学习分析大型网站的布局样式对初学者来说帮助是很大的。大多数初学者由于对技术的不自信，html结构往往写的很少，而且期望在少的结构上尽可能多的实现页面效果，这样无非加大了html结构所承载的样式属性，众多属性交织在一起就会容易产生bug。qq,sina,163三大官网撸一遍，基本是熟练了。

这里分享一个css哪些事儿的作者林小志运营的一个小站：[http://www.tianyizone.com](http://www.tianyizone.com/)　里边有一些新的特性的演示。

**二，js进阶篇：**

js进阶，等妙味看的差不多，心里难免痒痒，看老师用尽可能少的代码来实现了众多以前没法实现的动态效果。然后自己想写，但不知道写啥东西，下面就这个问题做以建议，写这些东西的目的有这么几个，一个是更加巩固自己的知识，另外一个是积累面试的作品以及以后工作中的可复用的代码片段：

1. 弹出层，

参照：<http://aui.github.com/artDialog/>

2. 日历

参照：<http://www.my97.net/dp/demo/index.htm>　mydate97

<http://bbs.blueidea.com/thread-3043966-1-1.html> 淘宝双日历

<http://bbs.blueidea.com/thread-3101895-1-1.html> 原生js日历

3. 图片轮换

SuperSlide2.1所有案例+TouchSlide1.0触屏滑动特效插件

<http://bbs.blueidea.com/thread-3049362-1-1.html>　轮换

<http://bbs.blueidea.com/thread-3032305-1-1.html> fgmcc经典作品

4. 模拟滚动条

参照：<http://www.oschina.net/question/89964_62203>

5. 模拟select，联动

参照：

<http://bbs.blueidea.com/thread-3009070-1-1.html>

<https://github.com/yessky/selector>

强大的selector

<http://aui.github.io/popupjs/doc/selectbox.html>

糖饼 select

<https://github.com/Johnqing/selectbox>

模拟下拉框

6. js分页

<http://bbs.blueidea.com/thread-3105789-1-1.html>

另外建议可以适当的看一些源码，看的目的无非是揭开神秘框架背后的代码实现，另外一个是开阔自己的视野，看了源码之后，不仅感叹到：原来js还可以这么玩，而且还玩的这么炫。当然有些源码并不时依目前的积累能看懂的，比如jQuery，是非常晦涩难懂，但为了方便分享，就全部罗列出来：

1. ext

<http://www.cnblogs.com/mogen_yin/archive/2009/04/30/1447236.html>

ext 2.2 经典50篇 -- 阅读ext源码

<http://www.cnblogs.com/gotolnc/archive/2009/03/10/1408063.html>

blog　资源汇总

<http://yahaitt.iteye.com/blog/193271>

透析Extjs的Ext.js源码（五）分析继承的实现

<http://www.cnblogs.com/peter-pc/articles/1612591.html>

Ext源代码阅读-001

<http://linder0209.iteye.com/category/58900>

源码解读系列文章(一直在看)

<http://snandy.iteye.com/category/124734>

snandy ext系列文件(准备要看的)

<http://liust1987.iteye.com/?page=2>

ext源码系列(准备要看的)

<http://linder0209.iteye.com/blog/1071818>

ext源码分析(准备要看的)

<http://blog.sina.com.cn/s/articlelist_1975505103_2_1.html>

ext源码分析( 比较零散，必看部分 )

[http://extasp.net](http://extasp.net/)

extasp.net框架结构

<http://www.iteye.com/topic/195409> 1111

ext继承 iteye

<http://yiminghe.iteye.com/category/64527>

何玉的iteyey

<http://lionheart.iteye.com/blog/232776>

js 继承

<http://blog.csdn.net/zhangxin09>

ext高手blog

<http://www.cnblogs.com/sanshi/archive/2009/07/08/1519036.html>

js继承

<http://www.cnblogs.com/minakata/archive/2011/10/07/2199872.html>

js高程继承

<http://www.iteye.com/topic/1047795>

ext其它扩展

<http://www.iteye.com/topic/1120551?page=2>

ext 2.3源码分析，bbs格式（其中有event对象的整理）

<http://www.cnblogs.com/iamlilinfeng/category/385121.html>

ext源码分析

<http://setting.iteye.com/category/53053>

ext基础教程

<http://andy-ghg.iteye.com/blog/1133456>

ext4的改变

<http://www.cnblogs.com/fangsui/archive/2012/04/18/2455537.html>

ext4开始

<http://docs.sencha.com/ext-js/4-0/#!/api>

ext在线api

<http://www.aspxcs.net/HTML/094538320.html>

ext体系结构

<http://linder0209.iteye.com/blog/865331>

ext文件结构

<http://hi.baidu.com/study_discuss/blog/item/d3c28913f7cb64936438dbf8.html>

extjs apply,applyif,override,extend 的区别

<http://www.oecp.cn/hi/xuan/blog/page/2>

ext源码

<http://www.open-open.com/home/space.php?uid=135360&do=blog&id=11658>

ext4.0基础概念与总结

 2. prototype

<http://blog.csdn.net/qgy18/article/details/673117>

prototype源码阅读

<http://www.cnblogs.com/NNUF/archive/2012/05/10/2494579.html>

prototype 源码解读

<http://www.cnblogs.com/cangqiong/archive/2007/12/21/1008801.html>

prototype-1.4.0(转载)

3. qwarp

<http://dev.qwrap.com/resource/js/_docs/_youa/#/content.htm>

api

<http://www.cnblogs.com/jkisjk/tag/QWrap/>

4. mootools

<http://www.cnblogs.com/oneroundseven/archive/2011/05/10/2041996.html>

mootools 源码分析之 Core.js

<http://suyang119.iteye.com/blog/1671065>

Mootools源码分析

<http://hmking.blog.51cto.com/3135992/675856>

MooTools 1.4 源码分析 - Class 修正版

5. jquery

<http://james.padolsey.com/jquery/>

jquery源码查询

<http://www.cnblogs.com/edison1105/archive/2012/07/31/2617518.html>

这些年、我收集的JQuery代码

<http://hxq0506.iteye.com/blog/1046334>

jquery1.6中的.prop()和.attr()异同

<http://www.cnblogs.com/zyj469470971/archive/2012/02/17/2355890.html>

jquery选择器

<http://www.cnblogs.com/sharpxiajun/archive/2011/09/22/2185566.html>

临摹jquery

<http://www.cnblogs.com/haogj/archive/2010/04/19/1715762.html>

jquery原型关系图

<http://www.ihiro.org/blog/wp-content/uploads/2010/08/sizzle_flow1.jpg>

jquery选择器

<http://www.cnblogs.com/lovesueee/archive/2012/10/18/2729829.html>

jquery 1.8.2 源码分析

<http://www.cnblogs.com/nuysoft/archive/2011/11/14/2248023.html>

jquery 1.6源码分析

<http://www.cnblogs.com/justany/archive/2013/01/22/2868501.html>

jquery事件分析

<http://www.cnblogs.com/hxling/>

jQuery工作原理解析以及源代码示例

jquery easyui 作者

<http://rapheal.sinaapp.com/>

jquery源码阅读

<http://www.ccvita.com/121.html>

<http://www.blog.highub.com/decoding-jquery/>

jquery源码阅读英文资料

<http://www.cnblogs.com/nuysoft/>

jquery源码阅读，这位作者出了一本书叫jquery技术内幕

<http://item.jd.com/1080321026.html>

<http://adamed.iteye.com/category/207898>

jq源代码阅读

<http://adamed.iteye.com/>

jquery源码分析

<http://www.cnblogs.com/krouky/archive/2011/12/27/Krouky.html>

jquery框架分析

<http://www.cnblogs.com/yaozhiyi/archive/2013/01/12/2855583.html>

jquery源码　元素位置

<http://www.oschina.net/news/46103/10-fresh-and-fantastic-jquery-plugins>

10个jquery插件，里边有个jquery滚动条写的不错。

<http://bbs.miaov.com/forum.php?mod=viewthread&tid=7385>

<http://www.miaov.com/2013/miaovideo/miaovideo.html>

[知识点类] 妙味课堂：一起学习jQuery源码【逐行分析jQuery源码的奥秘】

<http://www.cnblogs.com/aaronjs/p/3310937.html>

jquery 2.0.3源码分析

<http://www.cnblogs.com/aaronjs/p/3322466.html>

jquery 2.0.3源码分析 分析Sizzle引擎 - 编译函数（大篇幅）

<https://github.com/mythz/jquip>

jquery 压缩版

<http://www.cnblogs.com/aaronjs/p/3342344.html>

jQuery 2.0.3 源码分析 回调对象 - Callbacks

<http://bbs.miaov.com/forum.php?mod=viewthread&tid=7385>

jquery源码解读-妙味

<http://sunnylost.com/article/jquery.core1.html>

jQuery 源码阅读

<http://www.cnblogs.com/yangjunhua/>

浅析jQuery基础框架

<http://www.ihiro.org/sizzle-custom-selector>

sizzle源码

<http://www.cnblogs.com/52cik/p/jquery-source-position.html>

浅谈jQuery事件源码定位

<http://www.cnblogs.com/suprise/p/3622760.html>

jquery 1.11源码分析

<http://blog.csdn.net/baozhifei/article/details/6959392>

jQuery1.0源代码分析之domManip方法（七）

<http://blog.csdn.net/lihongxun945/article/category/1588879>

jquery2.0.4源码分析

<http://www.cnblogs.com/chyingp/category/546594.html>

jQuery源码分析系列

<http://www.cnblogs.com/baochuan/archive/2012/11/21/2781118.html>

通过jQuery源码学习javascript（一）

<http://pashanhu.github.io/show_by_tag.html?tag=jQuery>

学习jQuery源码

<http://www.zhihu.com/question/19671522?rf=20617754>

jQuery资料知乎

<http://www.cn-cuckoo.com/deconstructed/jquery.html>

李松峰的解构

<http://www.iteye.com/topic/786984>

jquery1.43源码分析之动画部分

<http://www.iteye.com/topic/783260>

jquery 队列函数

<http://www.cnblogs.com/guoyansi19900907/>

jquery 源码分析-思思博士

6. Backbone

<http://www.cnblogs.com/hao-dotnet/p/3444575.html>

Backbone源码阅读手记

7. seajs

<http://seajs.org/docs/>

seajs

<http://www.cnblogs.com/yinshen/p/3190429.html>

seajs源码分析

<http://www.cnblogs.com/webstone/articles/3043119.html>

seajs研究一二三

<http://www.heiniuhaha.com/seajs/2012/08/14/seajs-combat/>

seajs实战参考 Supporting tagline

<http://panxuepeng.github.io/seajslib>

SeaJS 组件库

<https://github.com/seajs/seajs/issues>

seajs issues

## 3. web前端开发分享-css,js提高篇

一. css基础知识掌握之后(个人的标准是：弄清块元素与内联元素的区别，弄清float的应用场景，弄清position[pə'zɪʃən] 下五个属性static['stætɪk],relative['rɛlətɪvli] ,absolute['æbsəlut] ,fixed[fɪkst],inherit[ɪn'hɛrɪt] 的区别与联系以及他们的应用场景，英文单词确定要能准确发音，方便与面试官探讨时准备的传达你要表达的意思)，剩下需要更多的练习以应对不同的设计的布局要求，然后遇到问题解决问题，成长就是需要一个这样跌宕起伏的过程。由于互联网资源很丰富，反而让初学者迷失了方向，下面推荐几款比较经典并且有特点的专题页面，可以进行分析，学习，模仿：

1. 新闻类型页面：

a. <http://news.qq.com/>  新闻综合页面

b. <http://news.qq.com/society_index.shtml>  新闻列表页面

c. <http://news.qq.com/a/20131210/000409.htm>  新闻详细页面

d. <http://coral.qq.com/1004281511>  评论页面

2. 图片类型页面：

a. <http://news.qq.com/photo.shtml>  图片综合页面

b. <http://news.qq.com/l/photon/ztlf/list2013089143749.htm>  图片列表

c. <http://news.qq.com/a/20131211/003572.htm> 图片详细

这个插件是用jquery写的，有兴趣的同学学习一下其源码

3. 股票类型页面

a. <http://finance.qq.com/hk/>  股票综合页面

b. <http://finance.qq.com/hk/mjls/news.htm>  股票中专家列表

4. 商城类：

a. [http://www.tmall.com](http://www.tmall.com/) 天猫首页



<http://brand.tmall.com/?spm=3.7095809.20000004.1.uVcdqb> 品牌街

<http://miao.tmall.com/?spm=3.7095809.20000004.2.uVcdqb> 喵鲜生

<http://vip.tmall.com/vip/index.htm?spm=3.7095809.20000004.3.uVcdqb> 积分聚乐部

<http://3c.tmall.com/?spm=3.7095809.20000004.4.uVcdqb> 电器城

<http://shoufa.tmall.com/?spm=3.7095809.20000004.5.uVcdqb> 新首发

<http://chaoshi.tmall.com/?spm=3.7095809.20000004.6.uVcdqb> 天猫超市

<http://www.tmall.com/yao?spm=3.7095809.20000004.7.uVcdqb> 医药馆

b. [http://www.jd.com](http://www.jd.com/) 京东



<http://channel.jd.com/fashion.html> 品牌街

<http://channel.jd.com/newarrivals.html> 新品

<http://channel.jd.com/topshops.html> 热销榜

<http://sale.jd.com/act/VgmuCbZvEzp.html> 潮流资讯

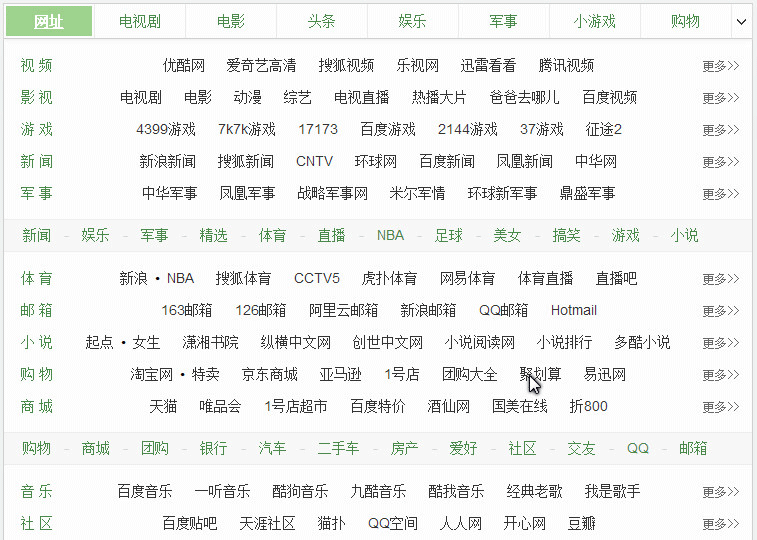
c.  [http://www.51buy.com](http://www.51buy.com/) 易迅

d. [http://www.dangdang.com](http://www.dangdang.com/) 当当

f.  [http://www.z.cn](http://www.z.cn/) 亚马逊

类似的页面还有很多就不一一列举：

[http://www.hao123.com](http://www.hao123.com/)



<http://gouwu.hao123.com/sc/>



二. js还有很多很多东西要学习了解。js是基于原型的OOP语言，想进一步了解纯的OOP思想，推荐了解一下JAVA。JAVA他本身除了是一个纯OOP语言之外，还有很多后台开发的东西可以一拼了解。然后关于构造函数，原型，实例等等之间的联系需要搞的明白一点。

      一、构造器(constructor)

    1、constructor始终指向创建当前对象的构造（初始化）函数。

<http://www.cnblogs.com/jikey/archive/2011/05/13/2045005.html>

    2、每个函数都有一个默认的属性prototype，而这个prototype的constructor默认指向这个函数

    From: <http://blog.csdn.net/kittyjie/article/details/4380918>

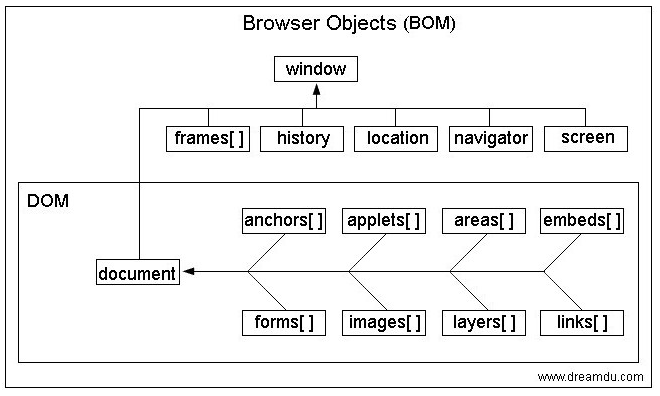
    其实在JS中构造器就是函数，函数就是构造器，对象实例就是通过var obj=new 函数();这种形式新建出来的实例。区别这些，在说prototype和constructor。从上面的英文中可以看出，prototype是个对象，里面定义了一个constructor，那么我们可以推论出，constructor是对象实例的属性！而不是函数（构造器）的属性。反过来，prototype是函数（构造器）的属性，而不是实例的属性！

    我要强调一点就是\_\_proto\_\_是实例和Person.prototype之间的关系，而constructor是实例和Person之间的关系

<http://blog.csdn.net/a569171010/article/details/7467514>

    他们的关系：

    instance -> \_\_proto\_\_ -> person.prototype -> constructor <- person.prototype -> function Person(){} -> Object() -> window



<http://www.cnblogs.com/o-andy-o/archive/2012/11/20/2778260.html>

<http://ejohn.org/apps/learn/#1>

    2. prototype

    一、什么是原型(prototype)

    1、prototype本质上还是一个JavaScript对象；

    2、每个函数都有一个默认的prototype属性；

    3、通过prototype我们可以扩展Javascript的内建对象

<http://blog.csdn.net/a569171010/article/details/7467514>

<http://www.iteye.com/topic/195409>

<http://stauren.net/log/eggxfygvg.html>

    二. 原型链： 常用原型来实现继承，其基本思想是利用原型让一个引用类型继承另一个引用类型的属性和方法。简单说，构造函数、原型和实例的关系：每个构造函数都有一个原型对象，原型对象都包含一个指向构造函数的指针，而实例都包含一个指向原型对象的内部指针。假如我们让原型对象等于另一个类型的实例。那结果将是：此时的原型对象将包含一个指向另一个原型的指针，相应地，另一个原型中也包含着一个指向另一个构造函数的指针。假如另一个原型又是另一个类型的实例，那么上述关系成立，如此层层递进，就构成了实例与原型的链接。这就是原型链的基本概念。

<http://blog.endlesscode.com/2010/01/23/javascript-prototype-chain/>

<http://www.cnblogs.com/ziyunfei/archive/2012/10/15/2723963.html>

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/JavaScript_Overview?redirectlocale=en-US&redirectslug=JavaScript%2FGuide%2FJavaScript_Overview>

<https://developer.mozilla.org/zh-CN/docs/JavaScript/Guide>

<http://blog.csdn.net/siczxw555/article/details/6138428>

<http://www.cn-cuckoo.com/2007/08/01/understand-javascript-closures-72.html>

<http://www.zhuoda.org/lunzi/101543.html>

3. 闭包

<http://www.cnblogs.com/luluping/archive/2009/05/27/1490551.html>

<http://www.cnblogs.com/xiaoruoen/archive/2011/09/23/2185832.html>

<http://ghsky.com/2010/11/deep-into-javascript-closure-part-two.html>

<http://www.felixwoo.com/archives/247>

<http://www.cnblogs.com/hh54188/archive/2012/03/01/2374921.html>

<http://blog.csdn.net/a569171010/article/details/7484125>

<http://bbs.51js.com/viewthread.php?tid=79018&extra=&highlight=%2Bwinter&page=1>

<https://developer.mozilla.org/zh_tw/Core_JavaScript_1.5_%E6%95%99%E5%AD%B8/%E9%96%89%E5%8C%85%E7%9A%84%E9%81%8B%E7%94%A8>

4. call,apply

<http://topic.csdn.net/u/20081209/23/0823c850-5a56-421e-8922-fa35b26a517a.html>

<http://www.cnblogs.com/treasurelife/archive/2008/03/05/1092251.html>

5. function

<http://desert3.iteye.com/blog/1198400>

<http://chillwarmoon.iteye.com/blog/77123>

<http://software.intel.com/zh-cn/articles/javascript-first-class-citizen-function/?cid=sw:prccsdn229032>

<http://www.cn-cuckoo.com/main/wp-content/uploads/2009/12/named-function-expressions-demystified.html#introduction>

6. js扩展的一些知识点：

<http://www.cn-cuckoo.com/main/wp-content/uploads/2009/12/named-function-expressions-demystified.html#introduction>

    命名函数表达式探秘

<http://bbs.51js.com/viewthread.php?tid=85245&highlight=scope>

    Scope Chain的问题

<http://bbs.51js.com/viewthread.php?tid=83841&highlight=%D7%F7%D3%C3%D3%F2>

    try catch作用域的疑问

<http://bbs.51js.com/viewthread.php?tid=85310&highlight=%D7%F7%D3%C3%D3%F2>

    变量作用域和词法作用域的区别

<http://hax.iteye.com/blog/104393>

    关于JScript 5.6及其之前（IE6）引擎的创建对象的性能问题

<http://bbs.51js.com/viewthread.php?tid=83190&extra=page%3D2%26amp;filter%3Ddigest>

    JScript下Array对象的性能问题

<http://bbs.51js.com/viewthread.php?tid=83586&extra=page%3D1&page=1>

    为为什么不用with

<http://bbs.51js.com/viewthread.php?tid=79018&extra=&highlight=%2Bwinter&page=1>

    JS引擎的Closure测试

<http://bbs.51js.com/viewthread.php?tid=74299&highlight>=

    转贴Aimingoo:JavaScript面向对象的支持

<http://bbs.51js.com/viewthread.php?tid=77398&highlight>=

    用functional思想编写js一例——缓动效果

<https://developer.mozilla.org/zh-CN/docs/JavaScript/Guide/Details_of_the_Object_Model>

    对象模型的细节

## 4. web前端开发分享-css,js工具篇

web前端开发乃及其它的相关开发，推荐[sublime text](https://github.com/jikeytang/sublime-text), webstorm(jetbrains公司系列产品)这两个的原因在于，有个技术叫emmet, http://docs.emmet.io，而这两个对他的支持是：

sublime text 是所有编辑器里边支持emmet比较好的的唯一一款。

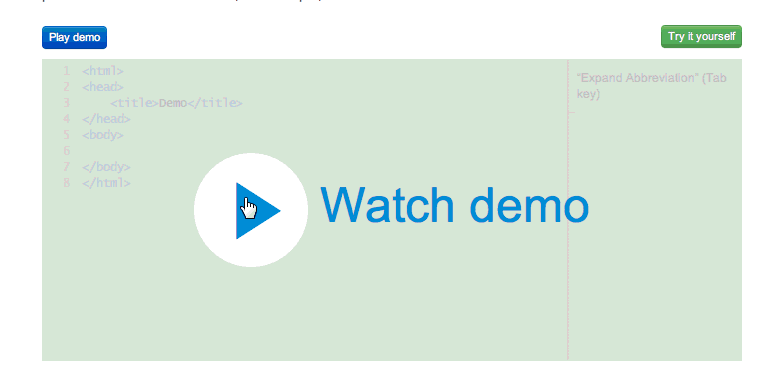
webstorm是所有编辑器里边唯一内置emmet的一款，并且将emmet的精神往前推动的一款。

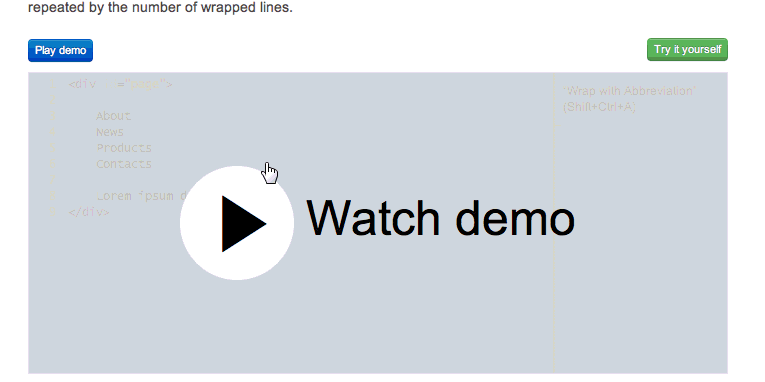
 phpstorm，idea包括所有webstorm的功能。只是phpstorm是针对phper的，idea是针对javaer，webstorm针对前端er。

他们的详细对比： <http://www.36ria.com/5884>

一. css工具:

1. emmet <http://docs.emmet.io/>





a. 可以以插件的形式安装在dreamweaver,notepad++,editplus,[sublime](http://www.sublimetext.com/)各个编辑器上。

<http://www.qianduan.net/zen-coding-a-new-way-to-write-html-code.html>

<http://www.qianduan.net/the-zen-coding-renamed-as-emmet.html>

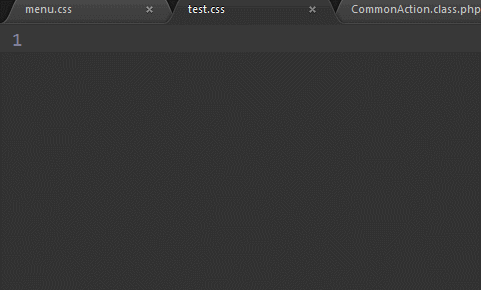
<http://www.qianduan.net/zen-coding-update-to-0-7.html>

<http://www.qianduan.net/zen-coding-resources-to-update.html>

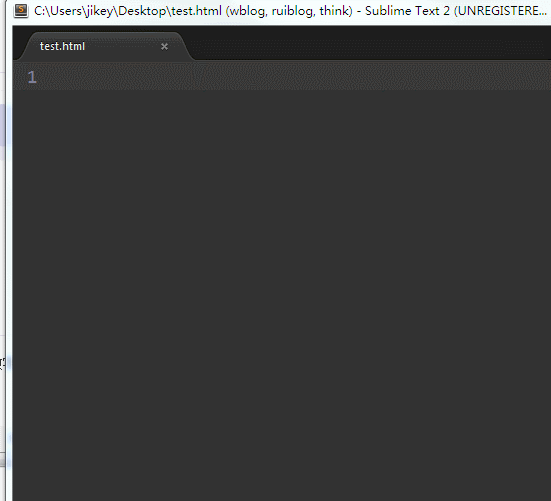
<http://www.qianduan.net/zen-coding-for-dreamweaver-plug-in-tutorial.html>

emmet在sublime上的效果：

1). css ：他与其它编辑器的区别在于细节上，比如ctrl,alt+方向键可以修和里边的数值

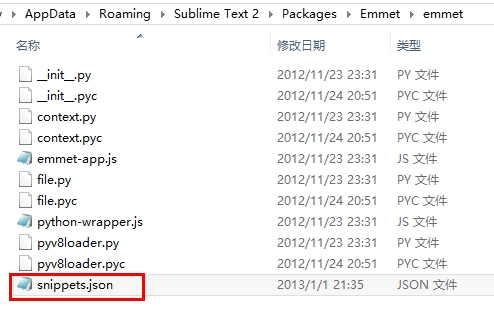


2). html 效果：



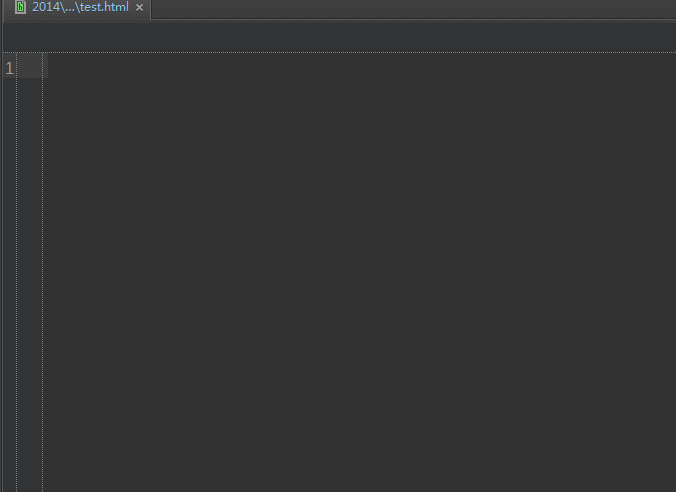
3). sublime的emmet规则修改：

packages->Emmet->emmet->snippets.json

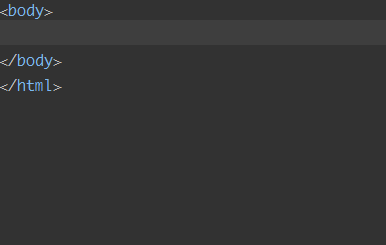


b. 只有一个例外jetbrain公司的产品线：webstorm,phpstorm,intellij idea都是内置的，不需要安装。

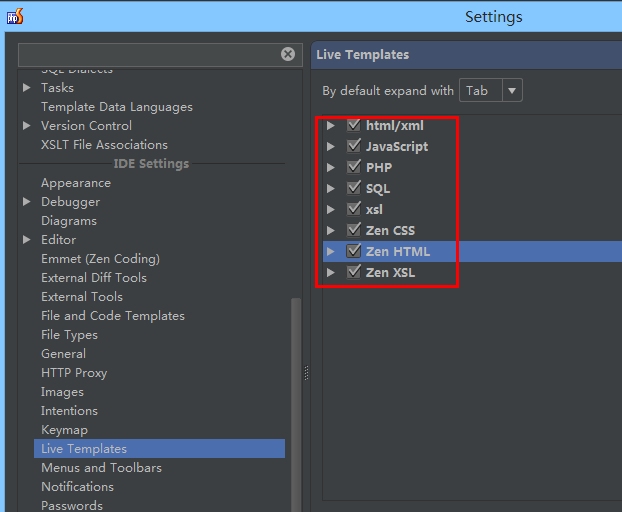
1).



写js的效果：



2). webstorm的emmet规则修改：Files -> settings -> Live Templates



2. css压缩办法：

a. 通过正则五步替换法，适用于任何编辑器：

<http://www.cnblogs.com/sky000/archive/2010/12/22/1913719.html>

b. 模块化工具

二. js工具：

1. 压缩等：

<http://www.yulonghu.com/?post=5>

在线压缩

<http://tools.css-js.com/>

在线压缩

<http://tool.oschina.net/>

工具集

2. 以emmet的方式开发js,php,jsp,java and so on。

<http://www.cnblogs.com/jikey/archive/2012/03/28/2420704.html>

<http://www.cnblogs.com/jikey/archive/2012/06/14/2549347.html>

3. firebug,chrome调试工具的使用：

<http://ued.taobao.org/blog/?p=5534>

chrome调试

<http://www.cnblogs.com/QLeelulu/archive/2011/08/28/2156402.html>

chrome的调试

<https://developers.google.com/chrome-developer-tools/docs/commandline-api>

chrome console 命令详解

<http://www.ruanyifeng.com/blog/2008/06/firebug_tutorial.html>

Firebug入门指南

<http://www.ruanyifeng.com/blog/2011/03/firebug_console_tutorial.html>

Firebug控制台详解

<https://developers.google.com/chrome-developer-tools/docs/mobile-emulation>

chrome移动版调试

<http://www.seejs.com/archives/296>

【译】Android上的远程调试

<http://blog.jobbole.com/31178/>

用Chrome开发者工具做JavaScript性能分析

<http://blog.moocss.com/code-snippets/mobile-web-code-snippets/1384.html>

用谷歌(Chrome)浏览器来当手机模拟器

<http://www.cnblogs.com/xiaoyao2011/p/3447421.html>

查看事件绑定

chrome.exe --user-agent="Mozilla/5.0 (Linux; U; Android 2.2; en-us; Nexus One Build/FRF91) AppleWebKit/533.1 (KHTML, like Gecko) Version/4.0 Mobile Safari/533.1"

<http://jinlong.github.io/blog/2013/08/29/devtoolsecrets/>

综合浏览器开发工具

<http://ccforward.net/2013/02/firebug_console.html>

关于FireBug和Chrome的控制台

<http://www.oschina.net/translate/performance-optimisation-with-timeline-profiles>

<http://h5dev.uc.cn/article-25-1.html>

chrome profiles

<http://www.cppblog.com/deercoder/archive/2011/10/22/158886.html>

Chrome神器Vimium快捷键学习记录

<http://www.iinterest.net/2014/05/09/chrome-dev-tool-workspace/>

Chrome 开发工具 Workspace 使用

4. js参考手册：

[w3c手册](http://download.csdn.net/detail/jikeytang/3249206)

5. 其它开发工具：

新手推荐：editplus，vim

团队开发：

webstorm, phpstorm(phper), pycharm(pythoner), intellij idea(javaer).

<http://note.youdao.com/share/?id=2d7ff03aeca64808fdc0f394ffb520bf&type=notebook>

6. 模块化开发grunt

<http://www.gruntjs.org/article/getting_started.html>

入门

<http://stackoverflow.com/questions/17059487/how-to-setup-grunt-in-webstorm-on-windows>

扩展工具grunt

<http://www.gruntjs.org/article/home.html>

<http://www.cnblogs.com/tugenhua0707/p/3497488.html>

前端项目构建工具---Grunt

<http://javascript.ruanyifeng.com/tool/grunt.html>

<http://www.infoq.com/cn/articles/GruntJs>

<http://blog.segmentfault.com/frankfang/1190000000372612>

享受 Grunt

<http://www.cnblogs.com/yexiaochai/p/3603389.html>

<http://www.cnblogs.com/yexiaochai/p/3594561.html>

叶小钗 第一弹

7. gulpjs

[http://gulpjs.com](http://gulpjs.com/)

官网

<http://www.tuicool.com/articles/n2Uj2y>

教程

<http://gratimax.github.io/search-gulp-plugins>

gulp plugins

https://github.com/gulpjs/gulp

<http://blog.segmentfault.com/laopopo/1190000000372547?page=1>

前端构建工具gulp入门教程

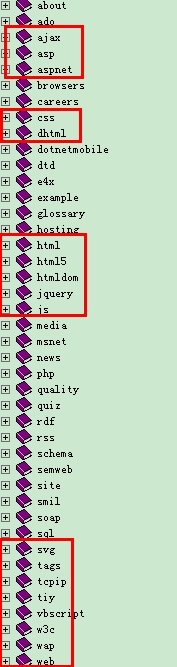
<http://zhuanlan.zhihu.com/TLA42/19691575>

Gulp 另一种自动化流水线

三：jq工具

1. oschina[参考手册](http://tool.oschina.net/apidocs)大全。

2. w3c\_all.chm，w3cschool离线版，包含了javascript,jquery,php,sql,html,css2.x,3.x的手册。



## 5. web前端开发分享-css,js深化篇

一. css练习网易专题：

1. <http://news.163.com/> 新闻

    跟腾讯的新闻版式大体没有大的变化，只是细节。

2. <http://news.163.com/photo/#Current>

    版式很独具一格。

3. <http://news.163.com/photoview/00AO0001/40918.html?from=ph_ss#p=9FSPHU1600AO0001>

    图片的详细页。

4. <http://sports.163.com/> 体育频道

5. <http://sports.163.com/gjb/> 体育列表

6. <http://ent.163.com/> 娱乐频道

二. js

1. js - oop

<http://msdn.microsoft.com/zh-cn/magazine/cc163419.aspx>

使用面向对象的技术创建高级 Web 应用程序

<http://desert3.iteye.com/blog/1198400>

JavaScript语言精粹读书笔记- JavaScript函数

<http://zzy603.iteye.com/blog/973649>

JS的Object漫想：从现象到“本质”

<http://chillwarmoon.iteye.com/blog/77123>

对于javascript的function的总结

<http://www.cnblogs.com/mrsunny/archive/2011/05/09/2041185.html>

javascript 类属性、类方法、类实例、实例属性、实例方法

<http://xiebiji.com/2010/01/static_js/?wptheme=Plainscape&ie=1>

Js中方法类的拟静态属性、公有属性、私有属性

<http://www.cnblogs.com/jeffwongishandsome/archive/2009/05/17/1457674.html>

javascript作用域链(Scope Chain)初探

<http://www.iteye.com/topic/288808>

javascript面向对象技术基础(四)

<http://www.cnblogs.com/RicCC/archive/2008/02/15/JavaScript-Object-Model-Execution-Model.html>

JavaScript对象模型-执行模型

<http://www.cnblogs.com/fool/tag/%E7%90%86%E8%A7%A3Javascript/>

理解Javascript

<http://www.cnblogs.com/qiantuwuliang/archive/2009/10/17/1584998.html>

JavaScript面向对象-静态方法-私有方法-公有方法-特权方法

<http://topic.csdn.net/u/20110123/15/f71c5ebb-7704-480b-9379-17a96e920fee.html>

JavaScript对象与继承教程之内置对象（上）

<http://topic.csdn.net/u/20110124/21/e6a88be3-d784-4973-b212-35789b05881f.html>

JavaScript对象与继承教程之内置对象（下）

<http://blog.csdn.net/aimingoo/article/details/6634977>

再谈JavaScript的数据类型问题

<http://www.linuxfly.org/post/524/>

in操作符

2. js 各开发中心

<http://www.ibm.com/developerworks/cn/web/>

ibm开发社区

<http://www.ibm.com/developerworks/cn/web/wa-ecma262/index.html>

ECMA-262，第 5 版

<https://developer.mozilla.org/zh-TW/>

firefox开发者专栏

<http://bonsaiden.github.com/JavaScript-Garden/zh/>

js秘密花园

<https://developer.apple.com/library/safari/navigation/>

safari开发中心

<http://developer.chrome.com/extensions/api_index.html>

chrome开发中心（chrome的内核已转向blink）

## 6. web前端开发分享-css,js移动篇

随着移动市场的逐步扩大及相关技术的日趋完善，对前端开发提出了新的岗位要求，在继承前人成果的基础上需要在新的历史条件下有新的创新。移动端的开发，虽然没有IE6众多问题的折磨，但是多平台，多设备的兼容，也是如恶梦般存在。不过话说回来，着重还是考验基础的css 2.x的功底，只要基础扎实了，应付移动端的兼容也是手到擒来。当然css3.x, html5新增加的标签还是需要学习，具体的推荐：

一. css

#### 1). 响应式web设计



<http://book.douban.com/subject/20390374/>

#### 2). HTML5 与 CSS3 权威指南

<http://book.douban.com/subject/6025285/>

包括html5,css3完整的基础入门书籍

二.js

### 1. 知识集合

<https://github.com/jtyjty99999/mobileTech>

99移动端知识集合

<https://github.com/AlloyTeam/Mars>

AlloyTeam-Mars

<http://www.oschina.net/translate/mobile-app-optimization-and-performance>

优化移动体验的HTML5技巧

<https://github.com/hoosin/mobile-web-favorites>

移动前端开发大杂烩

<http://html5boilerplate.com/>

<http://www.zhihu.com/question/19551815>

<http://www.ruanyifeng.com/blog/2012/05/responsive_web_design.html>

自适应网页设计

HTML5 Boilerplate

<http://eightmedia.github.io/hammer.js/>

移动端事件库

[http://webix.com](http://webix.com/)

支持移动端的框架

<http://www.xueui.cn/ui-tutorial-collection/>

移动端UI入门知识大集合

<http://www.cnblogs.com/iamzhanglei/>

当耐特

他出了一本书叫：[html5实验室-Canvas世界](http://book.douban.com/subject/10826093/)

### 2. 书籍分享：

#### 1). HTML5 Canvas游戏开发实战

<http://book.douban.com/subject/23820912/>

这书前半部分是canvas基础，后半部分是游戏的示例。

#### 2). HTML5高级程序设计



<http://book.douban.com/subject/5402708/>

根据目录大家可能也看出来了，这是一本基础入门的书籍。

**3).**[**移动web手册 移动Web第一书 前端国际大牛PPK最新力作**](http://book.douban.com/subject/26369130/)



本书彻底厘清移动Web开发与传统PC网站开发的本质区别

互联网产品移动端活跃用户数超越PC端已大势所趋

### 3. 工具分享：

#### 1). 测试工具

[多界面设备测试](http://www.responsinator.com/)

[Opera多终端界面模拟](http://www.opera.com/zh-cn/developer/mobile-emulator)

[其它辅助工具](http://www.zcool.com.cn/article/ZNjMzMjQ=.html)

#### 2). 开发工具

[webstorm](http://www.jetbrains.com/webstorm/)

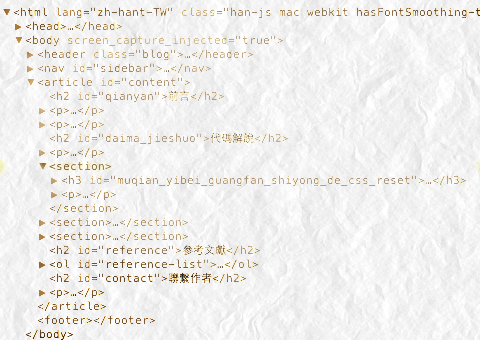
# 元素

## 慎用section标签

### 泛滥的<section>

在xhtml构建网页阶段，有一个名词叫“Divtis”（“Div症”），意为重构人员过多地使用了<div>包裹元素：嵌套繁琐或是错误使用（如该用<span>时也用<div>）。随着HTML5的流行，大量“语义化”标签的出现似乎拯救了他们，因为页面总算看起来没有那么多<div>了。  
  
然而实际情况却不是这样，除去一些伪html5页面[[1]](http://lightcss.com/careful-with-section/#target-fakehtml5)，我看到大部分页面里<div>仿佛在一夜之间就蒸发掉了。他们中的一部分变成了<header>、<footer>、<article>、<nav>、<aside>，剩下的却一股脑全变成了<section>，恐怖！

使用语义化的标签不是更好吗？为什么是恐怖？其实这并不好用一句话表达出来：**HTML5的新标签只是帮助页面里的不同板块角色更明确，使重要内容更容易被机器准确地识别出来。**在这方面或许跟标签的“role”属性比较像，我认为HTML5标签如果使用得当，“role”才是应该消失的东西。如果页面布局从div+css变成section+css，满屏都是同样的“语义化”标签就等于满屏都是没有语义的标签，因为机器看不出哪些<section>是容器哪些不是了，所谓的语义化也就大打折扣。所以，我起了一个比较吓人的标题：慎用<section>。



### <section>是什么？

其实在[W3C官方文档](http://www.w3.org/TR/html5/sections.html#the-section-element)中不难找到答案：

*The section element represents a generic section of a document or application. A section, in this context, is a thematic grouping of content, typically with a heading.*

<section>看起来像是有语义版的<div>，但实际的使用是用在一个专题性的版块，且通常带有一个标题。适合用于章节、标签切换效果的每一个tab容器或论文中有编号的地方，也可以用于网站主页中划分简介、新闻、联系信息等板块[[2]](http://lightcss.com/careful-with-section/#target-homepage_section)。

官方还给了一个提示：

*The section element is not a generic container element. When an element is needed for styling purposes or as a convenience for scripting, authors are encouraged to use the div element instead. A general rule is that the section element is appropriate only if the element’s contents would be listed explicitly in the document’s outline.*

说的十分明确，<section>不是一个普通的容器元素，当为了样式布局时，建议使用<div>。当元素内容明确的列举在大纲时建议使用<section>。

我想W3C的介绍说的已经足够明确，只要谨记一点就不容易用错：<section>不是一个通用的纯容器标签。对这个问题我困惑了很久，因为有些HTML5教程让我迷惑，而很多大网站也是在疯狂的使用<section>作为容器标签。

### 使用<section>

<section>可以相互嵌套，在页面中定义了一个特殊的顶级区块，于是可以从<h1>开始列提纲而不用担心会破坏网页的纲要。尽管没找到明确说明，但我个人认为在每一层<section>里还是仅使用一个<h1>为佳。

以Wordpress主题为例，我认为与#sidebar相对的#content部分，使用<section>包裹一堆<article>作为页面的主要内容并不太合适。而使用<section>用在整体布局是明显错误的了。

而适合使用<section>标签的地方有：

* 文章的评论列表，有着整齐的结构；
* 文章内容的目录，有着文章内部结构纲要；
* 侧栏widget，在[WebDesignWall](http://webdesignerwall.com/)中我看到了这样的设计，因为widget内容大都是评论列表、文章列表，有着清晰的结构且是独立完整的一部分；
* 包裹文章中各个章节的段落，但要在编辑器完成，目前来看并不易用。

随着标签越来越多样化，把标签用对本来就不是一件容易事，用得完全合理就更难了。好在目前这些东西用户是看不到的，产品经理也不会留意。如果没有强迫症，大可不必纠结与此。

最后我再啰嗦一次：  
**1、不要把<section>当<div>那样用；**  
**2、没有section+css这个东西，严格的说div+css都是错误的说法。**

好了，如果你有什么见解，欢迎发表评论与我交流。

伪html5页面[1]：即仅使用HTML5的<!DOCTYPE html>声明而再无HTML5特性、完全采用<div>标签布局的页面。

[2]：多为一些公司主页的版块化区域，“新闻”版块可能只有简单文章列表而没有正文，此类版块多在主要区域展示。

# 布局

## HTML网页布局：静态、自适应、流式、响应式

##### 静态布局（Static Layout）

即传统Web设计，对于PC设计一个Layout，在屏幕宽高有调整时，使用横向和竖向的滚动条来查阅被遮掩部分；

对于移动设备，单独设计一个布局，使用不同的域名如wap.或m.。

##### 自适应布局（Adaptive Layout）

自适应布局（Adaptive）的特点是分别为不同的屏幕分辨率定义布局。布局切换时页面元素发生改变，但在每个布局中，页面元素不随窗口大小的调整发生变化。  
你可以把自适应布局看作是静态布局的一个系列。

##### 流式布局（Liquid Layout）

流式布局（Liquid）的特点（也叫"Fluid") 是页面元素的宽度按照屏幕进行适配调整，主要的问题是如果屏幕尺度跨度太大，

那么在相对其原始设计而言过小或过大的屏幕上不能正常显示。

##### 响应式布局（Responsive Layout）

分别为不同的屏幕分辨率定义布局，同时，在每个布局中，应用流式布局的理念，即页面元素宽度随着窗口调整而自动适配。  
可以把响应式布局看作是流式布局和自适应布局设计理念的融合。

## 清理浮动的那些事儿（1）——6种清理浮动的方法

**文章目录**[ – ]

1. [前言](http://lightcss.com/all-about-clear-float/#toc-25)
2. [清理浮动的方法](http://lightcss.com/all-about-clear-float/#toc-26)
   1. [使用带有clear属性的空元素](http://lightcss.com/all-about-clear-float/#toc-27)
   2. [使用CSS的:after伪元素](http://lightcss.com/all-about-clear-float/#toc-28)
   3. [使用CSS的overflow属性进行怪异处理](http://lightcss.com/all-about-clear-float/#toc-29)
   4. [给浮动元素的容器添加浮动](http://lightcss.com/all-about-clear-float/#toc-30)
   5. [使用邻接元素清理](http://lightcss.com/all-about-clear-float/#toc-31)
   6. [其他方法](http://lightcss.com/all-about-clear-float/#toc-32)
3. [我的解决方案](http://lightcss.com/all-about-clear-float/#toc-33)
   1. [综合运用方案](http://lightcss.com/all-about-clear-float/#toc-34)
   2. [一劳永逸方案](http://lightcss.com/all-about-clear-float/#toc-35)
4. [小结](http://lightcss.com/all-about-clear-float/#toc-36)

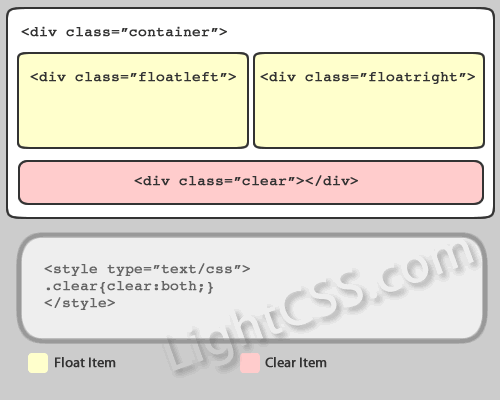
### 前言

(update 2011.02.01)配合图片重写了一下本文。近期有一项目涉及修改别人所写的主题，在清理浮动的环节上无比痛苦。在此整理一下关于清理浮动的方法和我的看法，希望能提供一些参考作用。

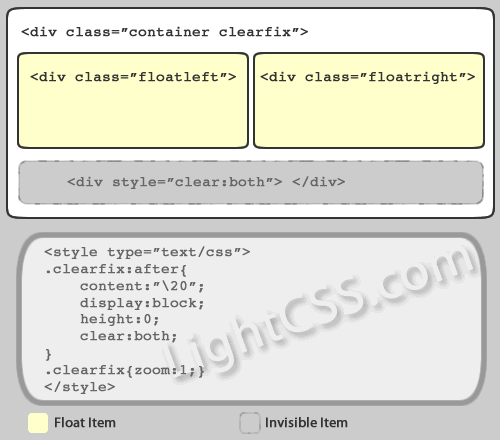
浮动（Float）最初的定义我想只是为了实现图文环绕效果，而如今通常在多栏布局时使用，是CSS里最最常用的一种布局手段。比起圣杯布局等，它更加容易上手并容易处理布局问题。由于浮动的特性，在布局中使用就产生了两个问题——会对后面的元素造成影响而且造成浮动元素的容器高度失效。因此我们需要随时清理浮动，方法有很多但各有弊端。选择合理的清理方式除了让代码看起来更优雅，还会对日后维护提供很大的方便。

### 清理浮动的方法

#### 使用带有clear属性的空元素

在浮动元素后使用一个空元素如<div class=”clear”></div>，并在CSS中赋予.clear{clear:both;}属性即可清理浮动。亦可使用<br class=”clear” />或<hr class=”clear” />来进行清理。  
  
我想说这并不是一个好方法，尽管它兼容所有浏览器并且随用随清。这个方法需要添加大量无语义的html元素，你能想象一个并不算复杂的footer里就使用4次div.clear吗？天哪！

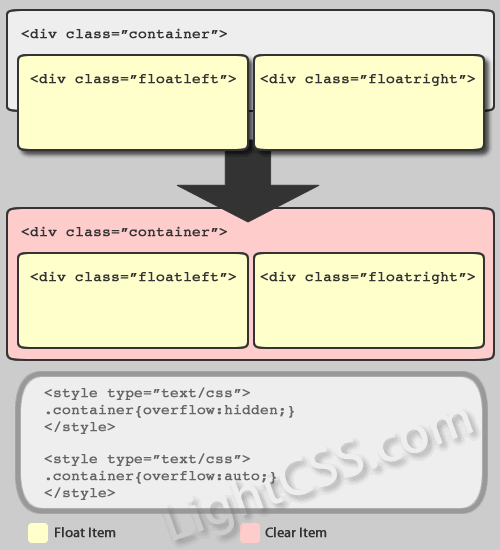
#### 使用CSS的:after伪元素

给浮动元素的容器添加一个clearfix的class，然后给这个class添加一个:after伪元素实现元素末尾添加一个看不见的块元素（Block element）清理浮动。这是我认为目前比较完美的解决方式。  
  
如图所见，通过CSS伪元素在容器的内部元素最后添加了一个看不见的空格“/20”或点“.” （后者可理解为一些国外书籍过时的介绍:0）并且赋予clear属性来清除浮动。需要注意的是为了IE6和IE7浏览器，要给clearfix这个class添加一条zoom:1;触发haslayout（你可以把它写到IE6、7的CSS hacker文件里，这样不会影响W3C标准验证）。

该方法需要给每组浮动元素都添加一个容器，推荐在页面布局时使用。大量使用依旧会对代码量造成一些影响。

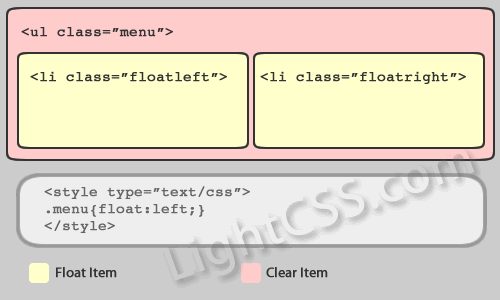
另外我想说的一点：我倾向于[《无懈可击的Web设计II》](http://book.douban.com/subject/4935289/)一书中提到的把.clearfix改为.group，让代码的可读性更好。group说明内部元素为一组，使用clearfix客户会误认为“你写了些神马？竟然用了这么多‘修正’！”。

#### 使用CSS的overflow属性进行怪异处理

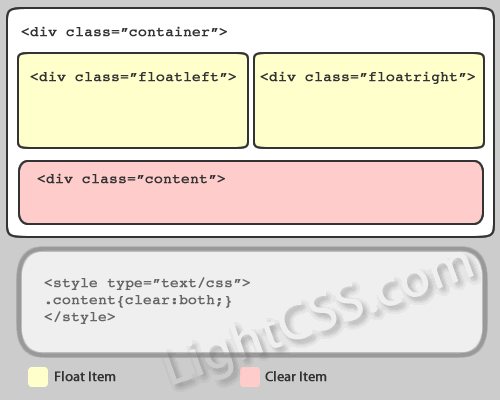
给浮动元素的容器添加overflow:hidden;或overflow:auto;可以怪异清理浮动。  
  
在清理浮动前（如箭头上图），浮动的元素跳离了container层，自己跑到上层去了（用阴影表示）。在添加overflow属性后（如箭头下图），浮动元素又回到了容器层，把容器高度撑起，达到了清理浮动的效果。

虽然很多地方以“将来浏览器可能不兼容”为由，不建议使用此方法。我倒是觉得目前使用此法的国内外网站众多，浏览器短时间内并不会在此问题上后退处理。此外这个方法无需添加额外的class，在做主题时比较实用。我建议在局部和无法添加class的地方使用该方法清理浮动，但是不要作为主要清理浮动方式。overflow:auto;还是不要用了。

#### 给浮动元素的容器添加浮动

给浮动元素的容器也添加上浮动属性即可清理内部浮动。同overflow法一样无须添加额外的class，使用方便但是会对下面的文档造成影响。  
  
建议在容器原本就浮动或者容器使用了绝对定位时使用该方法。不要在主要布局中使用。

#### 使用邻接元素清理

什么都不做，给浮动元素后面的元素添加clear属性。  
  
要确保的是content元素跟两个浮动元素都在同一层container容器内，如果content在container元素后面，建议使用:after伪元素清理。

#### 其他方法

使用display:table模拟表格布局也可以解决浮动的问题，并可自动对齐高度。但是由于table的一系列问题，还是不要使用了。

### 我的解决方案

#### 综合运用方案

在网页主要布局时使用:after伪元素方法并作为主要清理浮动方式；在小模块如ul或textwidget里使用overflow:hidden;（留意可能产生的隐藏溢出元素问题）；如果本身就是浮动元素则可自动清除内部浮动，无需格外处理；正文中使用邻接元素清理之前的浮动。

优点：可以有效降低代码量，使用得当可以完善解决浮动清理问题。  
缺点：需要熟悉各种清理方式的原理和利弊，需做好注释，否则对日后维护造成不必要的麻烦。

#### 一劳永逸方案

整站使用相对完美的:after伪元素法清理浮动。本方案适合文档结构清晰、模块化的页面。

优点：使用方便，维护方便，很少出现错误。  
缺点：需要清晰的文档结构，否则会造成代码量剧。

### 小结

通过对各种方法的分析，体现出网页重构时除了要对css应用得当并考虑良好的兼容性，还要培养写良好html文档结构的能力。只有html的结构整洁，模块清晰，才能更好的使用css处理布局中遇到的各种问题。

下一篇中我打算结合简单的案例综合运用各方法解决清理浮动的各种问题。

# 浏览器兼容

# 零碎知识点

## CSS !important

这是因为IE对盒之间距离的解释的bug造成的.

!important是CSS1就定义的语法，作用是提高指定样式规则的应用优先权(参见：W3.org的解释)。

语法格式{ sRule!important },直接写在定义的最后面，如：

p{color:green !important;}

注意：IE一直都不支持这个语法，而其他的浏览器都支持。因此我们就可以利用这一点来分别给FF和IE浏览器样式定义。

<html>

<head>

<style type="text/css">

#box {

height:30px!important;

height:18px;

border:1px solid #000

}

</style>

</head>

<body>

<span id="box">!important</span>

</body>

</html>

在Mozilla中浏览时候，能够理解!important的优先级，因此显示height:30px;的高度。

在IE中浏览时候，不能够理解!important的优先级，因此显示height:18px;的高度。

利用!important，我们可以针对IE和IE以外浏览器设置不同的样式，只要在IE以外浏览器样式的后面加上!important。

请记住!important、掌握!important、并合理的加以利用!important,它将会为你的CSS布局起到意想不到的效果！

# 资源

## Hack

<http://blog.csdn.net/freshlover/article/details/12132801>

## CSS 常用代码

#### 1. css 2.x

* 文字换行

/\*强制不换行\*/

white-space:nowrap;

/\*自动换行\*/

word-wrap: break-word;

word-break: normal;

/\*强制英文单词断行\*/

word-break:break-all;

* 两端对齐

text-align:justify;text-justify:inter-ideogra

* [去掉Webkit(chrome)浏览器中input(文本框)或textarea的黄色焦点框](http://www.cnblogs.com/niao/archive/2012/09/07/2674511.html)

input,button,select,textarea{ outline:none;}

textarea{ font-size:13px; resize:none;}

* [去掉chrome记住密码后自动填充表单的黄色背景](http://www.tuicool.com/articles/EZ777n)
* ie6: position:fixed

.fixed-top /\* position fixed Top \*/{position:fixed;bottom:auto;top:0; }

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

\*html{background-image:url(about:blank);background-attachment:fixed;}

* clearfix

.clearfix:after{visibility:hidden;display:block;font-size:0;content:" ";clear:both;height:0;}

.clearfix{display:inline-block;}

html[xmlns] .clearfix{display:block;}

\* html .clearfix{height:1%;}

.clearfix{\*zoom: 1;}

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

.clearfix{overflow:hidden;\_zoom:1;}

<http://www.daqianduan.com/3606.html>

* seperate-table

.tab{border-collapse:separate;border:1px solid #e0e0e0;}

.tab th,.tab td{padding:3px;font-size:12px;background:#f5f9fb;border:1px solid;border-color:#fff #deedf6 #deedf6 #fff;}

.tab th{background:#edf4f0;}

.tab tr.even td{background:#fff;}

<table class="tab" width="100%" cellpadding="0" cellspacing="0" border="0">

<tr>

<th>111</th>

<td>222</td>

</tr>

<tr>

<th>111</th>

<td>222</td>

</tr>

</table>

* min-height: 最小高度兼容代码

.minheight500{min-height:500px;height:auto !important;height:500px;overflow:visible;}

* 鼠标不允许点击

cursor:not-allowed;

* mac font: osx平台字体优化

font-family:"Hiragino Sans GB","Hiragino Sans GB W3",'微软雅黑';

* 文字过多后显示省略号

.ellipsis,.ell{white-space:nowrap;overflow:hidden;text-overflow:ellipsis}

#### 2. css 3

* title 换行

&#13;

* 关闭 x 符号

&#215;

* 投影

.b{box-shadow:inset 1px -1px 0 #f1f1f1;text-shadow:1px 1px 0px #630;}

filter:progid:DXImageTransform.Microsoft.gradient(enabled='true',startColorstr='#99000000',endColorstr='#99000000');background:rgba(0,0,0,.6);

background:rgba(0,0,0,0.5);filter:progid:DXImageTransform.Microsoft.gradient(GradientType=0,startColorstr='#50000000',endColorstr='#50000000')\9;

* [search占位](http://www.qianduan.net/search-box-style-custom-webkit.html)

::-webkit-input-placeholder {}

::-moz-input-placeholder {}

input:focus::-webkit-input-placeholder { color: transparent; }

-webkit-appearance:none; google边框去除

input[type="search"]{-webkit-appearance:textfield;} // 去除chrome默认样式

http://i.wanz.im/2011/02/04/remove\_border\_from\_input\_type\_search/

http://blog.csdn.net/do\_it\_\_/article/details/6789699

line-height: normal; /\* for non-ie \*/

line-height: 22px\9; /\* for ie \*/

* [全部浏览器的兼容代码生成](http://www.colorzilla.com/gradient-editor/) [CSS 实现 textArea 的 placeholder 换行](http://segmentfault.com/a/1190000000362621)
* 阻止默认事件

pointer-events:none;

* [去掉输入框聚焦时候的白色背景](http://ntesmailfetc.blog.163.com/blog/static/20628706120139184457401/)

-webkit-user-modify: read-write-plaintext-only;

* [input:focus时input不随软键盘升起而抬高的情况](http://www.cnblogs.com/hongru/archive/2013/02/06/2902938.html)

:focus{-webkit-tap-highlight-color:rgba(255, 255, 255, 0);

-webkit-user-modify:read-write-plaintext-only;}

* 变灰 gray

html{

filter: grayscale(100%);

-webkit-filter: grayscale(100%);

-moz-filter: grayscale(100%);

-ms-filter: grayscale(100%);

-o-filter: grayscale(100%);

filter: url("data:image/svg+xml;utf8,<svg xmlns=\'http://www.w3.org/2000/svg\'><filter id=\'grayscale\'><feColorMatrix type=\'matrix\' values=\'0.3333 0.3333 0.3333 0 0 0.3333 0.3333 0.3333 0 0 0.3333 0.3333 0.3333 0 0 0 0 0 1 0\'/></filter></svg>#grayscale");

filter:progid:DXImageTransform.Microsoft.BasicImage(grayscale=1);

-webkit-filter: grayscale(1);

}

* firefox 阻止选中

-moz-user-focus:ignore;-moz-user-input:disabled;-moz-user-select:none;

* 箭头

display:block;border:solid transparent;line-height: 0;width:0; height:0;border-top:solid #0288ce;border-width:8px 6px 0 6px;

border-style:solid; border-width:7px; border-color:transparent transparent transparent #ff7020;

position:absolute;top: 0;left: 0;border-width:20px;border-style:solid;border-color:#d1ddde transparent transparent #d1ddde;

ie6 bug测试，把border-style设为dashed.

* 取消textarea右下角可拖动手柄

resize:none

* 取消chrome form表单的聚焦边框

input,button,select,textarea{outline:none}

textarea{resize:none}

* 取消a链接的黄色边框

a{-webkit-tap-highlight-color:rgba(0,0,0,0);}

* 取消input,button焦点或点击时蓝色边框

input{outline:none;}

* webkit 水平居中

display:-webkit-box;-webkit-box-pack:center; -webkit-box-align: center;

position:absolute; top:50%;left:50%;transform:translate(-50%,-50%);

* 取消chrome 搜索x提示

input[type=search]::-webkit-search-decoration,

input[type=search]::-webkit-search-cancel-button,

input[type=search]::-webkit-search-results-button,

input[type=search]::-webkit-search-results-decoration {

display: none;

}

* [chrome取消默认黄色背景](http://stackoverflow.com/questions/2338102/override-browser-form-filling-and-input-highlighting-with-html-css)

input:-webkit-autofill {-webkit-box-shadow: 0 0 0px 1000px white inset;}

input:-webkit-autofill,

textarea:-webkit-autofill,

select:-webkit-autofill {

-webkit-box-shadow: 0 0 0 1000px white inset;

}

autocomplete="off"

* 手机版本网页a标记虚线框问题

a:focus {outline:none;-moz-outline:none;}

* 焦点去除背景

-webkit-tap-highlight-color:rgba(255, 255, 255, 0);

-webkit-tap-highlight-color:transparent; // i.e. Nexus5/Chrome and Kindle Fire HD 7''

* placeholder占位符颜色自定义

input:-moz-placeholder {color: #369;}

::-webkit-input-placeholder {color:#369;}

* [IOS 禁用高亮](http://hi.barretlee.com/2014/03/31/tap-highlight-in-webview/)

-webkit-tap-highlight-color:rgba(255,0,0,0.5);-webkit-tap-highlight-color:transparent; /\* For some Androids \*/

* IOS iframe 滚动 [滚动回弹特效](http://www.cnblogs.com/flash3d/archive/2013/09/28/3343877.html)

-webkit-overflow-scrolling:touch;overflow-y:scroll;

* [禁止选中文本](http://www.qianduan.net/introduce-user-select/)

-moz-user-select:none;

-webkit-user-select:none;

-ms-user-select:none;

user-select:none;

* [模糊(毛玻璃)效果1](http://www.zhangxinxu.com/wordpress/2013/11/%E5%B0%8Ftip-%E4%BD%BF%E7%94%A8css%E5%B0%86%E5%9B%BE%E7%89%87%E8%BD%AC%E6%8D%A2%E6%88%90%E6%A8%A1%E7%B3%8A%E6%AF%9B%E7%8E%BB%E7%92%83%E6%95%88%E6%9E%9C/)
* [模糊(毛玻璃)效果2](http://mao.li/css3-blur-filter-pratice/)
* [模糊(毛玻璃)逼真效果](http://codepen.io/ariona/pen/geFIK)

.blur {

-webkit-filter: blur(10px); /\* Chrome, Opera \*/

-moz-filter: blur(10px);

-ms-filter: blur(10px);

filter: blur(10px);

}

<img src="mm1.jpg" />

<img src="mm1.jpg" class="blur" />

* 显示旋转加载图片，[下拉加载数据](https://github.com/chalecao/chale/blob/master/iscroll.js)

#pullDown .pullDownIcon{display:inline-block;vertical-align:middle;width:40px;height:40px;background:url(https://github.com/chalecao/chale/blob/master/pull-icon%402x.png) 0 0 no-repeat;-webkit-background-size:40px 80px;background-size:40px 80px;-webkit-transition-property:-webkit-transform;-webkit-transition-duration:250ms}

#pullDown .pullDownIcon{-webkit-transform:rotate(0deg) translateZ(0)}

#pullDown .pullDownLabel{display:inline-block;vertical-align:middle;margin-left:5px;}

#pullDown.flip .pullDownIcon{-webkit-transform:rotate(-180deg) translateZ(0)}

#pullDown.loading .pullDownIcon{background-position:0 100%;-webkit-transform:rotate(0deg) translateZ(0);-webkit-transition-duration:0ms;-webkit-animation-name:loading;-webkit-animation-duration:2s;-webkit-animation-iteration-count:infinite;-webkit-animation-timing-function:linear}

@-webkit-keyframes loading{

from{-webkit-transform:rotate(0deg) translateZ(0)}

to{-webkit-transform:rotate(360deg) translateZ(0)}

}

<div id="pullDown" class="none loading">

<span class="pullDownIcon"></span><span class="pullDownLabel">正在载入中...</span>

</div>

* 手机多终端适配 media query[web app iphone4 iphone5 iphone6 响应式布局 适配代码](http://club.zoomla.cn/PItem?id=12594)

@media (device-height:480px) and (-webkit-min-device-pixel-ratio:2){/\* 兼容iphone4/4s \*/

.class{}

}

@media (device-height:568px) and (-webkit-min-device-pixel-ratio:2){/\* 兼容iphone5 \*/

.class{}

}

@media (device-height:667px) and (-webkit-min-device-pixel-ratio:2){/\* 兼容iphone6 \*/

.class{}

}

@media (device-height:736px) and (-webkit-min-device-pixel-ratio:2){/\* 兼容iphone6 Plus \*/

.class{}

}

* 屏蔽苹果浏览器对数字的识别[Meta标签中的format-detection属性及含义](http://blog.sina.com.cn/s/blog_51048da70101cgea.html)

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

* 移除HTML5 input在type="number"时的上下小箭头
  + 在chrome下：
* input::-webkit-outer-spin-button,input::-webkit-inner-spin-button{
* -webkit-appearance: none !important;
* margin: 0;

}

* + Firefox下：

input[type="number"]{-moz-appearance:textfield;}

* + 第二种方案：
    - 将type="number"改为type="tel"，同样是数字键盘，但是没有箭头。
* [HTML5手机浏览直接给一个号码打电话，发短信](http://java-er.com/blog/html5-mobile-call-sms/)

<a href="tel:15222222222">移动WEB页面JS一键拨打号码咨询功能</a>

<a href="sms:15222222222">移动WEB页面JS一键发送短信咨询功能</a>

<!--移动web页面自动探测电话号码-->

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

<meta http-equiv="x-rim-auto-match" content="none">

* [CSS判断横屏竖屏](http://www.w3cways.com/1772.html)

@media screen and (orientation: portrait) {

/\*竖屏 css\*/

}

@media screen and (orientation: landscape) {

/\*横屏 css\*/

}

//判断手机横竖屏状态：

window.addEventListener("onorientationchange" in window ? "orientationchange" : "resize", function() {

if (window.orientation === 180 || window.orientation === 0) {

alert('竖屏状态！');

}

if (window.orientation === 90 || window.orientation === -90 ){

alert('横屏状态！');

}

}, false);

//移动端的浏览器一般都支持window.orientation这个参数，通过这个参数可以判断出手机是处在横屏还是竖屏状态。

* rem 适配，内容太多，只贴网址
  + [rem自适应方案](https://github.com/imweb/mobile/issues/3)
  + [html5移动端页面分辨率设置及相应字体大小设置的靠谱使用方式](http://www.cnblogs.com/willian/p/3573353.html)
  + [移动端高清、多屏适配方案](http://www.html-js.com/article/Mobile-terminal-H5-mobile-terminal-HD-multi-screen-adaptation-scheme%203041)
  + [通过rem布局+media-query:aspect-ratio实现移动端全机型适配覆盖](http://xiaoyuze88.github.io/blog/2015/05/12/%E9%80%9A%E8%BF%87rem%E5%B8%83%E5%B1%80+media-query%E7%9A%84aspect-ratio%E5%AE%9E%E7%8E%B0%E7%A7%BB%E5%8A%A8%E7%AB%AF%E5%85%A8%E6%9C%BA%E5%9E%8B%E9%80%82%E9%85%8D%E8%A6%86%E7%9B%96/)
  + [web app变革之rem](http://isux.tencent.com/web-app-rem.html)
  + [手机淘宝的flexible设计与实现](http://www.html-js.com/article/2402)
  + [移动端自适应方案](https://github.com/amfe/lib-flexible)
  + [【原创】移动端高清、多屏适配方案](http://www.html-js.com/article/3041)
  + [6个html5页面适配iphone6的技巧](http://qietuwang.baijia.baidu.com/article/73861)
  + [关于移动端 rem 布局的一些总结](http://segmentfault.com/a/1190000003690140)
  + [从网易与淘宝的font-size思考前端设计稿与工作流](http://www.cnblogs.com/lyzg/p/4877277.html)
  + [移动端自适应方案](http://f2e.souche.com/blog/yi-dong-duan-zi-gua-ying-fang-an/)
  + [MobileWeb 适配总结](http://www.w3ctech.com/topic/979)
  + [移动端web app自适应布局探索与总结](http://www.html-js.com/article/JavaScript-learning-notes%203234)
  + 公式
  + var PAGE\_MAX\_WIDTH = 1280,
  + BASE\_FONT\_SIZE = 50;
  + (function() {
  + function n() {
  + e.fontSize = Math.min(window.innerWidth / PAGE\_MAX\_WIDTH \* BASE\_FONT\_SIZE, BASE\_FONT\_SIZE) + "px"
  + }
  + var e = document.documentElement.style;
  + window.addEventListener("load", n),
  + window.addEventListener("resize", n),
  + n();

}());

## JS 常用代码

#### 1. PC - js

* 返回指定范围的随机数(m-n之间)的公式

Math.random()\*(n-m)+m

* [return false](http://stackoverflow.com/questions/1357118/event-preventdefault-vs-return-false)
* [return false](http://www.75team.com/archives/201)

// event.preventDefault()会阻挡预设要发生的事件.

// event.stopPropagation()会阻挡发生冒泡事件.

// 而return false则是前面两者的事情他都会做：

// 他会做event.preventDefault();

// 他会做event.stopPropagation();

// 停止callback function的执行并且立即return回来

* 防止被Iframe嵌套

if(top != self){

location.href = ”about:blank”;

}

* 两种图片lazy加载的方式 第一个By JS中级交流群 成都-猎巫 第二个By 上海-zenki

// @description 准备为图片预加载使用的插件

// 使用的图片容器css类名为lazy-load-wrap

// 图片真实地址为data-lazy-src

// 当lazy-load-wrap容器进入视口，则开始替换容器内所有需要延迟加载的图片路径，并更改容器的加载状态

//第一种方法

$.fn.compassLazyLoad=function(){

var \_HEIGHT=window.innerHeight,

\_lazyLoadWrap=$('.lazy-load-wrap');

var methods={

setOffsetTop:function(){

$.each(\_lazyLoadWrap,function(i,n){

$(n).attr({

'top':n.offsetTop-\_HEIGHT,

'status':'wait'

});

})

},

isShow:function(){

var \_scrollTop=$(window).scrollTop;

//利用image容器判断是否进入视口，而非image本身

$.each(\_lazyLoadWrap,function(){

var \_that=$(this);

if (\_that.attr('status')==='done') {

return;

};

if (\_that.attr('top')<=\_scrollTop) {

\_that.find('img[data-lazy-src]').each(function(i,n){

n.src=$(n).data('lazy-src');

});

\_that.attr('status','done');

};

})

},

scroll:function(){

$(window).on('scroll',function(){

methods.isShow();

});

},

init:function(){

methods.setOffsetTop();

methods.isShow();

methods.scroll();

}

};

methods.init();

}

//第二种方法

var exist=(function($){

var timer=null,

temp=[].slice.call($('.container'));

ret={};

for(var i=0,len=temp.length-1;i<=len;i++){

ret[i]=temp[i];

}

var isExist=function(winTop,winEnd){

for(var i in ret){

console.log(ret);

var item=ret[i],

eleTop=item.offsetTop,

eleEnd=eleTop+item.offsetHeight;

if((eleTop>winTop&&eleTop<=winEnd)||(eleEnd>winTop&&eleEnd<=winEnd)){

$(item).css('background','none');

new Tab($(item).attr('id'),data).init;

delete ret[i];

}

}

}

return {

timer:timer;

isExist:isExist;

};

})($);

//第三种方法

Zepto(function ($) {

var swiper = new Swiper('.swiper-container', {

pagination: '.swiper-pagination',

paginationClickable: true,

autoplay: 3000,

loop: true,

autoplayDisableOnInteraction: false

});

(function lazyLoad() {

var imgs = $(".lazyLoad");

var src = '';

$.each(imgs, function (index, item) {

src = $(item).attr('data-src');

$(item).attr('src', src);

});

})();

});

$(function () {

var lazyLoadTimerId = null;

/// 智能加载事件

$(window).bind("scroll", function () {

clearTimeout(lazyLoadTimerId);

lazyLoadTimerId = setTimeout(function () {

// 延迟加载所有图片

var isHttp = (location.protocol === "http:");

$("#ym\_images img").each(function () {

var self = $(this);

if (self.filter(":above-the-fold").length > 0) {

var originUrl = self.attr("data-original");

self.attr("src", originUrl);

}

});

}, 500);

});

});

* 某年某月的1号为星期几

var weekday = ["星期日", "星期一", "星期二", "星期三", "星期四", "星期五", "星期六"];

weekday[new Date(2015, 9, 1).getDay()]; //2015年10月1号

#### 2. Mobile - js

* [js 判断IOS, 安卓](http://caibaojian.com/browser-ios-or-android.html)

var u = navigator.userAgent, app = navigator.appVersion;

var isAndroid = u.indexOf('Android') > -1 || u.indexOf('Linux') > -1; //android终端或者uc浏览器

var isiOS = !!u.match(/**\(**i[^;]+;( U;)? CPU.+Mac OS X/); //ios终端

alert('是否是Android：'+isAndroid);

alert('是否是iOS：'+isiOS);

#### 3. [微信 weixin](http://loo2k.com/blog/detecting-wechat-client/)

* UserAgent 判断微信客户端

// Mozilla/5.0 (iPhone; CPU iPhone OS 8\_3 like Mac OS X) AppleWebKit/600.1.4 (KHTML, like Gecko) Mobile/12F70 MicroMessenger/6.1.5 NetType/WIFI

function isWechat() {

var ua = navigator.userAgent.toLowerCase();

return /micromessenger/i.test(ua) || /windows phone/i.test(ua);

}

* JS接口安全域名不填写，分享onMenuShareAppMessage直接会取默认值。

// 分享onMenuShareAppMessage直接会取默认值

* 关闭当前页面

WeixinJSBridge.call('closeWindow');

* [支付接口方法调用必须在addevent里边调用](http://www.cnblogs.com/true_to_me/p/3565039.html)

document.addEventListener('WeixinJSBridgeReady', function onBridgeReady(){

that.initOrder();

}, false);

* 支付接口方法调用必须在

WeixinJSBridge.invoke('getBrandWCPayRequest', d, function(res){

if(res.err\_msg == "get\_brand\_wcpay\_request:ok"){

// alert("支付成功");

// union.release(d.orderId);

resetUrl();

paySuccess('home', d.orderId);

} else {

cancelOrder(d.orderId);

// alert(res.err\_msg);

}

loading.hide();

});

* 瀑布流无限加载实例

// be dependent on jquery & jquery.infinitescroll.min.js

// insert this '<div id="more"><a href="api?page="></a></div>' to your page.html

(function($){

$(function(){

var $container = $('.list-wrap-gd');

function layOutCallBack() {

$container.imagesLoaded(function(){

$container.masonry({

itemSelector: '.item-bar',

gutter: 10

});

});

$container.imagesLoaded().progress( function() {

$container.masonry('layout');

});

}

layOutCallBack();

$container.infinitescroll({

navSelector : "#more",

nextSelector : "#more a",

itemSelector : ".item-bar",

pixelsFromNavToBottom: 300,

loading:{

img: "/images/masonry\_loading.gif",

msgText: ' ',

finishedMsg: "<em>已经到最后一页</em>",

finished: function(){

$("#more").remove();

$("#infscr-loading").hide();

}

},

errorCallback:function(){

$(window).unbind('.infscr');

},

pathParse: function (path, nextPage) {

var query = "";

var keyword=$("#search\_keyword").val();

var cat\_id=$("#cat\_id").val();

var brand\_id=$("#brand\_id").val();

var country\_id = $("#country\_id").val();

query = query + "&namekeyword="+keyword;

query = query +"&cat\_id="+cat\_id

query = query + "&brand\_id=" + brand\_id;

query = query + "&country\_id=" + country\_id;

path = [path,query];

return path;

}

},

function(newElements) {

var $newElems = $( newElements ).css({ opacity: 0 });

$newElems.imagesLoaded(function(){

$newElems.animate({ opacity: 1 });

$container.masonry( 'appended', $newElems, true );

layOutCallBack();

});

});

});

})(jQuery);

* [iOS，Safari浏览器，input等表单focus后fixed元素错位问题](https://www.snip2code.com/Snippet/176582/--iOS-Safari----input---focus-fixed-----)

if( /iPhone|iPod|iPad/i.test(navigator.userAgent) ) {

$(document).on('focus', 'input, textarea', function()

{

$('header').css("position", 'absolute');

$('footer').css("position", 'absolute');

});

$(document).on('blur', 'input, textarea', function()

{

$('header').css("position", 'fixed');

$('footer').css("position", 'fixed');

});

}

* 得到地理位置

function getLocation(callback){

if(navigator.geolocation){

navigator.geolocation.getCurrentPosition(

function(p){

callback(p.coords.latitude, p.coords.longitude);

},

function(e){

var msg = e.code + "\n" + e.message;

}

);

}

}

* [rem计算适配](http://isux.tencent.com/web-app-rem.html)

(function(doc, win){

var docEl = doc.documentElement,

resizeEvt = 'orientationchange' in window ? 'orientationchange' : 'resize',

recalc = function(){

var clientWidth = docEl.clientWidth;

if(!clientWidth) return;

docEl.style.fontSize = 20 \* (clientWidth / 320) + 'px';

};

if(!doc.addEventListener) return;

win.addEventListener(resizeEvt, recalc, false);

doc.addEventListener('DOMContentLoaded', recalc, false);

})(document, window);

* [另外一种rem方案](http://www.html-js.com/article/3041)

var dpr, rem, scale;

var docEl = document.documentElement;

var fontEl = document.createElement('style');

var metaEl = document.querySelector('meta[name="viewport"]');

dpr = window.devicePixelRatio || 1;

rem = docEl.clientWidth \* 2 / 10;

scale = 1 / dpr;

// 设置viewport，进行缩放，达到高清效果

metaEl.setAttribute('content', 'width=' + dpr \* docEl.clientWidth + ',initial-scale=' + scale + ',maximum-scale=' + scale + ', minimum-scale=' + scale + ',user-scalable=no');

// 设置data-dpr属性，留作的css hack之用

docEl.setAttribute('data-dpr', dpr);

// 动态写入样式

docEl.firstElementChild.appendChild(fontEl);

fontEl.innerHTML = 'html{font-size:' + rem + 'px!important;}';

// 给js调用的，某一dpr下rem和px之间的转换函数

window.rem2px = function(v) {

v = parseFloat(v);

return v \* rem;

};

window.px2rem = function(v) {

v = parseFloat(v);

return v / rem;

};

window.dpr = dpr;

window.rem = rem;

* 获取js所在路径

function getJsDir (src) {

var script = null;

if (src) {

script = [].filter.call(document.scripts, function (v) {

return v.src.indexOf(src) !== -1;

})[0];

} else {

script = document.scripts[document.scripts.length - 1];

}

return script ? script.src.substr(0, script.src.lastIndexOf('/')) : script;

}

* 从全局捕获错误

window.onerror = function (errMsg, scriptURI, lineNumber, columnNumber, errorObj) {

setTimeout(function () {

var rst = {

"错误信息：": errMsg,

"出错文件：": scriptURI,

"出错行号：": lineNumber,

"出错列号：": columnNumber,

"错误详情：": errorObj

};

alert(JSON.stringify(rst, null, 10));

});

};

* [如何通过 js 修改微信浏览器的title?](https://www.zhihu.com/question/26228251/answer/32405529)

var $body = $('body');

document.title = 'title'; // hack在微信等webview中无法修改document.title的情况

var $iframe = $('<iframe src="/favicon.ico"></iframe>').on('load', function(){

setTimeout(function(){

$iframe.off('load').remove()

}, 0)

}).appendTo($body)

#### 1. 常用方法 - js

* 字符串长度截取

function cutstr(str, len) {

var temp,

icount = 0,

patrn = /[^**\x**00-**\x**ff]/，

strre = "";

for (var i = 0; i < str.length; i++) {

if (icount < len - 1) {

temp = str.substr(i, 1);

if (patrn.exec(temp) == null) {

icount = icount + 1

} else {

icount = icount + 2

}

strre += temp

} else {

break;

}

}

return strre + "..."

}

* 替换全部

String.prototype.replaceAll = function(s1, s2) {

return this.replace(new RegExp(s1, "gm"), s2)

}

* 清除空格

String.prototype.trim = function() {

var reExtraSpace = /^\s\*(.\*?)\s+$/;

return this.replace(reExtraSpace, "$1")

}

* 清除左空格/右空格

function ltrim(s){ return s.replace( /^(\s\*|　\*)/, ""); }

function rtrim(s){ return s.replace( /(\s\*|　\*)$/, ""); }

* 判断是否以某个字符串开头

String.prototype.startWith = function (s) {

return this.indexOf(s) == 0

}

* 判断是否以某个字符串结束

String.prototype.endWith = function (s) {

var d = this.length - s.length;

return (d >= 0 && this.lastIndexOf(s) == d)

}

* 转义html标签

function HtmlEncode(text) {

return text.replace(/&/g, '&').replace(/**\"**/g, '"').replace(/</g, '<').replace(/>/g, '>')

}

* 时间日期格式转换

Date.prototype.Format = function(formatStr) {

var str = formatStr;

var Week = ['日', '一', '二', '三', '四', '五', '六'];

str = str.replace(/yyyy|YYYY/, this.getFullYear());

str = str.replace(/yy|YY/, (this.getYear() % 100) > 9 ? (this.getYear() % 100).toString() : '0' + (this.getYear() % 100));

str = str.replace(/MM/, (this.getMonth() + 1) > 9 ? (this.getMonth() + 1).toString() : '0' + (this.getMonth() + 1));

str = str.replace(/M/g, (this.getMonth() + 1));

str = str.replace(/w|W/g, Week[this.getDay()]);

str = str.replace(/dd|DD/, this.getDate() > 9 ? this.getDate().toString() : '0' + this.getDate());

str = str.replace(/d|D/g, this.getDate());

str = str.replace(/hh|HH/, this.getHours() > 9 ? this.getHours().toString() : '0' + this.getHours());

str = str.replace(/h|H/g, this.getHours());

str = str.replace(/mm/, this.getMinutes() > 9 ? this.getMinutes().toString() : '0' + this.getMinutes());

str = str.replace(/m/g, this.getMinutes());

str = str.replace(/ss|SS/, this.getSeconds() > 9 ? this.getSeconds().toString() : '0' + this.getSeconds());

str = str.replace(/s|S/g, this.getSeconds());

return str

}

* 判断是否为数字类型

function isDigit(value) {

var patrn = /^[0-9]\*$/;

if (patrn.exec(value) == null || value == "") {

return false

} else {

return true

}

}

* 设置cookie值

function setCookie(name, value, Hours) {

var d = new Date();

var offset = 8;

var utc = d.getTime() + (d.getTimezoneOffset() \* 60000);

var nd = utc + (3600000 \* offset);

var exp = new Date(nd);

exp.setTime(exp.getTime() + Hours \* 60 \* 60 \* 1000);

document.cookie = name + "=" + escape(value) + ";path=/;expires=" + exp.toGMTString() + ";domain=360doc.com;"

}

* 获取cookie值

function getCookie(name) {

var arr = document.cookie.match(new RegExp("(^| )" + name + "=([^;]\*)(;|$)"));

if (arr != null) return unescape(arr[2]);

return null

}

* 加载样式文件表

function LoadStyle(url) {

try {

document.createStyleSheet(url)

} catch(e) {

var cssLink = document.createElement('link');

cssLink.rel = 'stylesheet';

cssLink.type = 'text/css';

cssLink.href = url;

var head = document.getElementsByTagName('head')[0];

head.appendChild(cssLink)

}

}

* 返回脚本内容

function evalscript(s) {

if(s.indexOf('<script') == -1) return s;

var p = /<script[^**\>**]\*?>([^**\x**00]\*?)<**\/**script>/ig;

var arr = [];

while(arr = p.exec(s)) {

var p1 = /<script[^**\>**]\*?src=**\"**([^**\>**]\*?)**\"**[^**\>**]\*?(reload=**\"**1**\"**)?(?:charset=**\"**([\w**\-**]+?)**\"**)?><**\/**script>/i;

var arr1 = [];

arr1 = p1.exec(arr[0]);

if(arr1) {

appendscript(arr1[1], '', arr1[2], arr1[3]);

} else {

p1 = /<script(.\*?)>([^**\x**00]+?)<**\/**script>/i;

arr1 = p1.exec(arr[0]);

appendscript('', arr1[2], arr1[1].indexOf('reload=') != -1);

}

}

return s;

}

* 清除脚本内容

function stripscript(s) {

return s.replace(/<script.\*?>.\*?<**\/**script>/ig, '');

}

* 动态加载脚本文件

function appendscript(src, text, reload, charset) {

var id = hash(src + text);

if(!reload && in\_array(id, evalscripts)) return;

if(reload && $(id)) {

$(id).parentNode.removeChild($(id));

}

evalscripts.push(id);

var scriptNode = document.createElement("script");

scriptNode.type = "text/javascript";

scriptNode.id = id;

scriptNode.charset = charset ? charset : (BROWSER.firefox ? document.characterSet : document.charset);

try {

if(src) {

scriptNode.src = src;

scriptNode.onloadDone = false;

scriptNode.onload = function () {

scriptNode.onloadDone = true;

JSLOADED[src] = 1;

};

scriptNode.onreadystatechange = function () {

if((scriptNode.readyState == 'loaded' || scriptNode.readyState == 'complete') && !scriptNode.onloadDone) {

scriptNode.onloadDone = true;

JSLOADED[src] = 1;

}

};

} else if(text){

scriptNode.text = text;

}

document.getElementsByTagName('head')[0].appendChild(scriptNode);

} catch(e) {}

}

* 返回按ID检索的元素对象

function $(id) {

return !id ? null : document.getElementById(id);

}

* 检验URL链接是否有效

function getUrlState(URL){

var xmlhttp = new ActiveXObject("microsoft.xmlhttp");

xmlhttp.Open("GET",URL, false);

try{

xmlhttp.Send();

}catch(e){

}finally{

var result = xmlhttp.responseText;

if(result){

if(xmlhttp.Status==200){

return(true);

}else{

return(false);

}

}else{

return(false);

}

}

}

* 获取当前路径

var currentPageUrl = "";

if (typeof this.href === "undefined") {

currentPageUrl = document.location.toString().toLowerCase();

}else {

currentPageUrl = this.href.toString().toLowerCase();

}

* 获取页面高度

function getPageHeight(){

var g = document, a = g.body, f = g.documentElement, d = g.compatMode == "BackCompat"

? a

: g.documentElement;

return Math.max(f.scrollHeight, a.scrollHeight, d.clientHeight);

}

* 获取页面可视宽度

function getPageViewWidth(){

var d = document, a = d.compatMode == "BackCompat" ?

d.body: d.documentElement;

return a.clientWidth;

}

* 获取页面宽度

function getPageWidth(){

var g = document, a = g.body, f = g.documentElement, d = g.compatMode == "BackCompat"?

a: g.documentElement;

return Math.max(f.scrollWidth, a.scrollWidth, d.clientWidth);

}

* 随机数时间戳

function uniqueId(){

var a=Math.random,b=parseInt;

return Number(new Date()).toString()+b(10\*a())+b(10\*a())+b(10\*a());

}

* 日期格式化函数

Date.prototype.format = function(format){

var o = {

"M+" : this.getMonth()+1, //month

"d+" : this.getDate(), //day

"h+" : this.getHours(), //hour

"m+" : this.getMinutes(), //minute

"s+" : this.getSeconds(), //second

"q+" : Math.floor((this.getMonth()+3)/3), //quarter

"S" : this.getMilliseconds() //millisecond

};

if(/(y+)/.test(format)) format=format.replace(RegExp.$1,

(this.getFullYear()+"").substr(4 - RegExp.$1.length));

for(var k in o){

if(new RegExp("("+ k +")").test(format))

format = format.replace(RegExp.$1,RegExp.$1.length==1 ? o[k] :("00"+ o[k]).substr((""+ o[k]).length));

}

return format;

}

//调用

//new Date().format("yyyy-MM-dd hh:mm:ss");

* 返回顶部的通用方法

function backTop(btnId) {

var btn = document.getElementById(btnId);

var d = document.documentElement;

var b = document.body;

window.onscroll = set;

btn.style.display = "none";

btn.onclick = function() {

btn.style.display = "none";

window.onscroll = null;

this.timer = setInterval(function() {

d.scrollTop -= Math.ceil((d.scrollTop + b.scrollTop) \* 0.1);

b.scrollTop -= Math.ceil((d.scrollTop + b.scrollTop) \* 0.1);

if ((d.scrollTop + b.scrollTop) == 0) clearInterval(btn.timer, window.onscroll = set);

}, 10);

};

function set() {

btn.style.display = (d.scrollTop + b.scrollTop > 100) ? 'block': "none"

}

};

backTop('goTop');

* 获得URL中GET参数值

// 用法：如果地址是 test.htm?t1=1&t2=2&t3=3, 那么能取得：GET["t1"], GET["t2"], GET["t3"]

function get\_get(){

querystr = window.location.href.split("?")

if(querystr[1]){

GETs = querystr[1].split("&");

GET = [];

for(i=0;i<GETs.length;i++){

tmp\_arr = GETs.split("=")

key=tmp\_arr[0]

GET[key] = tmp\_arr[1]

}

}

return querystr[1];

}

* 数组去重

String.prototype.unique=function(){

var x=this.split(/[**\r\n**]+/);

var y='';

for(var i=0;i<x.length;i++){

if(!new RegExp("^"+x.replace(/([^\w])/ig,"\\$1")+"$","igm").test(y)){

y+=x+"\r\n"

}

}

return y

};

* 按字典顺序，对每行进行数组排序

function SetSort(){

var text=K1.value.split(/[**\r\n**]/).sort().join("\r\n");//顺序

var test=K1.value.split(/[**\r\n**]/).sort().reverse().join("\r\n");//反序

K1.value=K1.value!=text?text:test;

}

* 字符串反序输出

function IsReverse(text){

return text.split('').reverse().join('');

}

* 金额大写转换函数

//格式转换

function transform(tranvalue) {

try {

var i = 1;

var dw2 = new Array("", "万", "亿"); //大单位

var dw1 = new Array("拾", "佰", "仟"); //小单位

var dw = new Array("零", "壹", "贰", "叁", "肆", "伍", "陆", "柒", "捌", "玖"); //整数部分用

//以下是小写转换成大写显示在合计大写的文本框中

//分离整数与小数

var source = splits(tranvalue);

var num = source[0];

var dig = source[1];

//转换整数部分

var k1 = 0; //计小单位

var k2 = 0; //计大单位

var sum = 0;

var str = "";

var len = source[0].length; //整数的长度

for (i = 1; i <= len; i++) {

var n = source[0].charAt(len - i); //取得某个位数上的数字

var bn = 0;

if (len - i - 1 >= 0) {

bn = source[0].charAt(len - i - 1); //取得某个位数前一位上的数字

}

sum = sum + Number(n);

if (sum != 0) {

str = dw[Number(n)].concat(str); //取得该数字对应的大写数字，并插入到str字符串的前面

if (n == '0') sum = 0;

}

if (len - i - 1 >= 0) { //在数字范围内

if (k1 != 3) { //加小单位

if (bn != 0) {

str = dw1[k1].concat(str);

}

k1++;

} else { //不加小单位，加大单位

k1 = 0;

var temp = str.charAt(0);

if (temp == "万" || temp == "亿") //若大单位前没有数字则舍去大单位

str = str.substr(1, str.length - 1);

str = dw2[k2].concat(str);

sum = 0;

}

}

if (k1 == 3){ //小单位到千则大单位进一

k2++;

}

}

//转换小数部分

var strdig = "";

if (dig != "") {

var n = dig.charAt(0);

if (n != 0) {

strdig += dw[Number(n)] + "角"; //加数字

}

var n = dig.charAt(1);

if (n != 0) {

strdig += dw[Number(n)] + "分"; //加数字

}

}

str += "元" + strdig;

} catch(e) {

return "0元";

}

return str;

}

//拆分整数与小数

function splits(tranvalue) {

var value = new Array('', '');

temp = tranvalue.split(".");

for (var i = 0; i < temp.length; i++) {

value = temp;

}

return value;

}

# 前端之路

## web前端研发工程师编程能力成长之路

### ****【背景】****

如果你是刚进入WEB前端研发领域，想试试这潭水有多深，看这篇文章吧；  
如果你是做了两三年WEB产品前端研发，迷茫找不着提高之路，看这篇文章吧；  
如果你是四五年的前端开发高手，没有难题能难得住你的寂寞高手，来看这篇文章吧；

WEB前端研发工程师，在国内是一个朝阳职业，自07-08年正式有这个职业以来，也不过三四年的时间。这个领域没有学校的正规教育，没有行内成体 系的理论指引，几乎所有从事这个职业的人都是靠自己自学成才。自学成才，一条艰辛的坎坷路，我也是这样一路走来。从2002年开始接触WEB前端研发至今 已然有了9个年头，如今再回首，期间的走了很多弯路。推已及人，如果能让那些后来者少走些弯路，辛甚辛甚！

### ****【前言】****

所谓的天才，只不过是比平常人更快的掌握技能、完成工作罢了；只要你找到了正确的方向，并辅以足够的时间，你一样能够踏上成功彼岸。

本文将WEB前端研发编程能力划分了八个等级，每个等级都列举出了对应的特征及破级提升之方法，希望每位在看本文的同学先准确定位自己的等级（不要以你目前能力的最高点，而是以你当前能力的中档与之等级作对比，以免多走弯路），参考突破之法破之。

所谓的级别，只是你面对需求时的一种态度：能够完成、能够完美地完成、能够超出预期地完成。以追求完美的态度加以扎实的编程功力，那就是你的编程水平。

切记心浮气燥，级别够了，那级别里的东西自然就懂了。悟了就是悟了，没悟也没关系，静下心来，投入时间而已。

### ****一.【入门】****

**能够解决一些问题的水平。**有一定的基础（比如最常见的HTML标签及其属性、事件、方法；最常见的CSS属性；基础的JavaScript编程能力），能够完成一些简单的WEB前端研发需求。

举个例子：删除一字符串中指定的字符。

var str="www.baidu.com/?page";  
str=str.replace('?page',"");  
alert(str);  
str=str.substring(0,str.indexOf("/"));  
alert(str);

首先不要苛责代码的对错严谨，毕竟每个程序员都有这样的一个过程；其次，这两段代码在这个实例里没有什么大过错，可能会有瑕疵，但能够解决问题（删除指定的字符），这就是这个级别的特征。

再举个例子：

// 计算系统当前是星期几  
var str = "";  
var week = new Date().getDay();  
if (week == 0) {  
 str = "今天是星期日";  
} else if (week == 1) {  
 str = "今天是星期一";  
} else if (week == 2) {  
 str = "今天是星期二";  
 } else if (week == 3) {  
 str = "今天是星期三";  
 } else if (week == 4) {  
 str = "今天是星期四";  
 } else if (week == 5) {  
 str = "今天是星期五";  
 } else if (week == 6) {  
 str = "今天是星期六";  
 }  
// 或者更好一些  
var str1 = "今天是星期";  
var week = new Date().getDay();  
switch (week) {  
 case 0 :  
 str1 += "日";  
 break;  
 case 1 :  
 str1 += "一";  
 break;  
 case 2 :  
 str1 += "二";  
 break;  
 case 3 :  
 str1 += "三";  
 break;  
 case 4 :  
 str1 += "四";  
 break;  
 case 5 :  
 str1 += "五";  
 break;  
 case 6 :  
 str1 += "六";  
 break;  
}  
alert(str);  
alert(str1);

入门”阶段是每个程序员的必经之路，只要“入门”，你就上路了。所谓“师傅领进门，修行靠个人”，有了这个“入门”的基础，自己就可以摸索着前进了。

【进阶之路】

将JavaScript、HTML、CSS之类的编码帮助手册里的每个方法/属性都通读几遍！只有将基础打好，以后的路才能走的顺畅。参考这些帮助文档，力争写出无瑕疵的代码。

这些编码文档建议不仅是在入门提高期看，在你以后每个阶段破阶的时候都应该看看，最基础的东西往往也是最给力的东西，有时能够给你带来意想不到的收获。

### ****二.【登堂】****

**能够正确地解决问题。**不管你是通过搜索网络，或者通过改造某些成品代码（jQuery/Dojo/Ext/YUI）案例，只要能够无错地完成需求。

同样以上面的那段“字符串剪裁”代码为例：

var str="www.baidu.com/?page";  
str=str.replace(/?page/,"");  
alert(str);

仅仅解决问题对于“登堂”阶段来说已经不是问题，这个级别所给出方案不能是漏洞百出。以上面这段代码为例：replace方法的第一个参数虽然可以支持字符串，但最佳的类型是正则表达式；

var a = new Array("日", "一", "二", "三", "四", "五", "六");  
var week = new Date().getDay();  
var str = "今天是星期"+ a[week];  
alert(str);

对比“入门级”的代码，不管是从代码量、代码效率、代码优美性、代码思路来说，“登堂”级的这个日期处理代码都要优秀很多。

【进阶之路】

这个阶段虽然能够给出正确的解题方案，但是不一定是最优秀的方案。如何才能得到最优秀的方案呢？首先就是积累各种能够解决需求的方案，然后再验证每个方案，在这些方案中选择最好的一种。因此该阶段的进阶之路就是“行万里路，看万卷书”，积累各个需求的各个解决方案。

你可以扎身在专业论坛（蓝色理想、无忧、CSDN）里，通读所有的FAQ及帖子；你可以打开搜索引擎，穷举所有的搜索结果。自己建立测试环境一一验 证这些代码：去揣摩每段代码的意图，去比较每段代码之间的差异。这两条路可以让你快速完成原始积累，当你再面对大多数需求时能够说这些问题我以前做过，那 你就水到渠成地晋阶了。

### ****三.【入室】****

**最强代码，知道所有能够解决需求的各种方案，能够选择使用最优秀的方案满足需求。**这个级别基本上产品开发编程中的代码主力。给出的一招一式，招招都是绝招。

还以上面的那个例子为例，你能说出1、2、3之间的差别，以及适用于那种环境吗？

var str="www.baidu.com/?page";  
// 1、字符串剪裁  
str.substring(0, str.indexOf("?page"));  
// 2、正则表达式  
str.replace(/?page/, "");  
// 3、字符串分拆、合并  
str.split("?page").join("");

能够解决问题的方法会有很多，但是对于程序员来说应该选择最优秀的。上面这段代码从代码量来说“正则表达式”最优秀；从代码执行效率来说： “字符串剪裁”法最高（Chrome中“正则表达式”效率最高），split法最次；从可扩展性上来说，“正则表达式”法最优。具体使用那种方案视具体的 需求环境而定。

“入室”阶段，程序员应该能够肯定的回答：**对于这个需求而言，我的代码就是最优秀的代码。**

再以“今天是星期几”为例，“登堂”级的代码你敢说是最优秀的代码了吗？

// 计算系统当前是星期几  
var str = "今天是星期" + "日一二三四五六".charAt(new Date().getDay());

对比“登堂”级的示例代码，上面这段代码给你什么感受？程序员追求的就是完美。“入室”级别追求的就是每一招每一式的完美无缺。

从WEB前端编程来说，通过2年左右的努力，很多人能够达到这个水平，但是，很大一部分人的编程能力也就止步于此。或限于产品的需求单一性，或限于需求开发的时间紧迫性，或限于人的惰性，能够完美地解决当前的需求就够了。

由于长期处于技术平台期，技术上得不到提高，通常这个级别的工程师会比较燥。技术上小有所成；或追求个人的突破；或追求产品差异性带来的新鲜感；或者只是想换个心情；因此很多此级别的工程师会经常换公司。

**戒骄戒躁：**

切勿以为自己能写一手漂亮的代码而自满；  
切莫以为别人“尊称”你一声“大侠”你就以 “大侠”自居；  
切莫以为自己积累了一些得意的代码就成了框架式开发。

细节决定成败，优秀的方案并不能保证最终的成功。还以“删除指定字符串”为例，原始字符串从格式上来看应该是了个URL链接，在去除“pn=0”之 后，最末尾处留了一个尾巴“?”；如果原始字符串是“http://www.xxx.com/?pn=0&a=1”，去除“pn=0”之后 ? 和 & 两个符号紧贴一起，这更是明显的bug。

【进阶之路】

此阶段进阶之路就是：**切勿心浮气躁**；你不再被需求牵着走，而是你牵着需求走。注重细节，注意那些当前需求里没有明文给出的细节：代码性能的差异、运行平台（浏览器）的差异、需求的隐性扩展、代码的向后兼容等等。

再通读几遍HTML/CSS/JavaScript帮助文档。

我建议这个级别的工程师做一做WebTreeView控件，要求总节点量一万左右操作流畅，你的晋升之路就在这个控件的编码过程中。

### ****四.【入微】****

**最强解决方案。**你能够走在需求的前面，将当前需求里有的、没有直接提出来的、现在暂时没有但将来可能有的等等，及前端编程潜规则等各个方方面面都综合考虑，给出最优方案。以一招胜万招。

var str = "http://www.xxx.com/?pn=0"; // 删除指定字符 pn=0  
// 我将这个字符串里所可能想到的各种情况都列举出来  
var a = [  
 "http://www.xxx.com/VMpn=?pn=0"// pn= 可能出现在 ? 前  
 , "http://www.xxx.com/VMpn=?pn="// URL里允许pn 值为空  
 , "http://www.xxx.com/VMpn=?pn=0&a=1"// URL 里可有多个字段  
 , "http://www.xxx.com/VMpn=?a=1&pn=0"// 可能排在最后  
 , "http://www.xxx.com/VMpn=?a=1&pn=0&pn=1"// 可能有多个 pn 字段  
 , "http://www.xxx.com/VMpn=?a=1&pn=0&b=2"// 可能在中间  
 , "http://www.xxx.com/VMpn=?a=1&pn=0&pn=1&b=1" // 可能在中间成组  
 , "http://www.xxx.com/VMpn=?a=1&pn=0&b=1&pn=1" // 可能零星分布  
];  
/\* 需求的不言之秘：  
? 若出现在字符串最尾则要去之  
? & 两个符号不可重叠  
\*/  
var reg = /((\?)(pn=[^&]\*&)+(?!pn=))|(((\?|&)pn=[^&]\*)+$)|(&pn=[^&]\*)/g;  
  
for (var i = 0; i < a.length; i++) {  
 alert(a[i] + "\n" + a[i].replace(reg, "$2"));  
}

这个阶段已经不再追求一招一式，对你来说不是使用什么创新绝招解决需求，而是给出成熟稳重的方案，从根上解决问题。针对某个当前需求你的代码可能不是最优，但是针对此类的需求你的代码却是最优秀的代码。

【进阶之路】

很多WEB前端研发工程师在做了3-4年之后就会进入一个瓶颈期：产品开发需求是小菜一碟，没有新鲜的可以挑战的东西；代码开发中的稀奇的解题方法 都已经尝试过。没有了可挑战的难题，失去了探索的激情，也就没有了再上升的动力，好不容易走过“入室”级别的人又会有八九成止步于此。或转做技术领导人， 或转到其它的领域，或换公司。

这些人的上升之路在哪里呢？

这个阶段单单依靠技巧和数量的累积已经没有什么效果了，突破之路在第5层《化蝶》里会详细说明，我建议你在这个阶段末尾着重关注编程理论：面向对象/过程、代码组织形式、编译、代码规范、其它的框架设计等等。

我建议这个级别的工程师做一做WebEditor控件，不要求完整功能，但是该控件里的模块划分、代码组织、编程思想做到位，给出一个系统的解决方案。

### ****五.【化蝶】****

**破茧重生，这个层次关注的是编程语言本身，而不再关心产品需求。**什么是茧？产品需求就是茧。当你一招胜万招，打遍天下需求之时，你如果还拘泥于需求开发，那就是你限于茧中而不自知。要么就在这个茧里默默地老去，要么就破开茧获得新生。

还是以那个“字符串剪裁”的老例子：

/\*\*  
 \* 在拼接正则表达式字符串时，消除原字符串中特殊字符对正则表达式的干扰  
 \* @author:meizz  
 \* @version: 2010/12/16  
 \* @param {String} str 被正则表达式字符串保护编码的字符串  
 \* @return {String} 被保护处理过后的字符串  
\*/  
function escapeReg(str) {  
 return str.replace(new RegExp("([.\*+?^=!:\x24{}()|[\\]\/\\\\])", "g"), "\\\x241");  
}  
  
/\*\*  
 \* 删除URL字符串中指定的 Query  
 \* @author:meizz  
 \* @version:2010/12/16  
 \* @param {String} url URL字符串  
 \* @param {String} key 被删除的Query名  
 \* @return {String} 被删除指定 query 后的URL字符串  
\*/  
  
function delUrlQuery(url, key) {  
 key = escapeReg(key);  
 var reg = new RegExp("((\\?)("+ key +"=[^&]\*&)+(?!"+ key +  
 "=))|(((\\?|&)"+ key +"=[^&]\*)+$)|(&"+ key +"=[^&]\*)", "g");  
 return url.replace(reg, "\x241")  
}  
  
// 应用实例  
var str = "http://www.xxx.com/?pn=0"; // 删除指定字符 pn=0  
delUrlQuery(str, "pn");

这段代码相对于层次4《入微》有什么区别吗？从代码实现上来说没有太大的区别，但是从思路上来说却有着本质的区别：1、不再是就事论事，头疼医头，而是把一类问题抽象理论化，一招破万招；2、有封装的概念，不再是每次从零开始，而是站在半山腰开始爬。

在WEB前端研发队伍里也有很大一部分人《入室》层次时就自我感觉良好，直接跨跃到《化蝶》，积累自己的代码库，抽象化问题。但没有基础，缺少强大 的后劲，即使能够破茧也经受不了风吹雨打。一份不成熟的架构设计对团队开发带来的危害远大于它带来的好处，这种例子在业界屡见不鲜。不要拔苗助长，不要不 会走就想着跑，夯实基础，水到渠成地成长，厚积薄发，强力地破茧而出。

【进阶之路】

你已经从原始积累，到厚积薄发，到破茧而出之后，你所关注的应该不再是一招一式、一个项目、一个模块，而应该是一种思路，一种理论。你可以做以下几 个步骤以突破到更高层次：再仔细看几遍HTML/CSS/JavaScript接口帮助文档；选择一门强语言（C++/C#/Java等）观察理解这些语 言的组织结构，语言设计；看看原型链，链式语法编程，泛型，接口编程，DOM遥控器等等；仔细阅读成熟的WEB前端开发框架的设计文档，看他们为什么要这 样设计。

### ****六.【大侠】****

这里所说的大侠，不是大家互相吹捧的“大侠”，而是实至名归的高手。这个级别的人完全有能力写出不差于Bindows/jQuery/Ext/YUI/Dojo的同等级别规模的前端开发框架。**应用成熟的开发框架指导、解决问题。**

// 库文件 /mz/string/escapeReg.js  
/\*\*  
 \* 在拼接正则表达式字符串时，消除原字符串中特殊字符对正则表达式的干扰  
 \* @author:meizz  
 \* @version: 2010/12/16  
 \* @param {String} str 被正则表达式字符串保护编码的字符串  
 \* @return {String} 被保护处理过后的字符串  
\*/  
mz.string.escapeReg = function (str) {  
 return str.replace(new RegExp("([.\*+?^=!:\x24{}()|[\\]\/\\\\])", "g"), "\\\x241");  
}  
  
// 库文件 /mz/url/delQuery.js  
/// include mz.string.escapeReg;  
/\*\*  
 \* 删除URL字符串中指定的 Query  
 \* @author:meizz  
 \* @version:2010/12/16  
 \* @param {String} url URL字符串  
 \* @param {String} key 被删除的Query名  
 \* @return {String} 被删除指定 query 后的URL字符串  
\*/  
mz.url.delQuery = function (url, key) {  
 key = mz.string.escapeReg(key);  
 var reg = new RegExp("((\\?)("+ key +"=[^&]\*&)+(?!"+ key +  
 "=))|(((\\?|&)"+ key +"=[^&]\*)+$)|(&"+ key +"=[^&]\*)", "g");  
 return url.replace(reg, "\x241")  
}  
  
// 应用实例  
/// include mz.url.delQuery;  
var str = "http://www.xxx.com/?pn=0"; // 删除指定字符 pn=0  
mz.url.delQuery(str, "pn");

自成体系，有基础，也有理论高度。知道为什么这样设计，也知道什么样的设计最好。比如这个例子可以有这样的封装：

// 库文件 /mz/url/delQuery.js  
/// include mz.string.escapeReg;  
/\*\*  
 \* 删除URL字符串中指定的 Query  
 \* @author:meizz  
 \* @version:2010/12/16  
 \* @param {String} url URL字符串  
 \* @param {String} key 被删除的Query名  
 \* @return {String} 被删除指定 query 后的URL字符串  
\*/  
String.prototype.delQuery = function ( key) {  
 key = mz.string.escapeReg(key);  
 var reg = new RegExp("((\\?)("+ key +"=[^&]\*&)+(?!"+ key +  
 "=))|(((\\?|&)"+ key +"=[^&]\*)+$)|(&"+ key +"=[^&]\*)", "g");  
 return this.replace(reg, "\x241")  
}  
  
// 应用实例  
/// include mz.url.delQuery;  
var str = "http://www.xxx.com/?pn=0"; // 删除指定字符 pn=0  
str.delQuery("pn");

而为什么不采用下面的那种封装呢？经过了《知微》和《化蝶》你就懂了。

【进阶出路】

道法自然，从根上去寻找突破的契机。你可以研读HTML解析引擎设计与实现，JS解析引擎设计与实现，其它语言的代码解析与编译实现等等。

或者出些书。低级别的人写的书要么是一大抄，空无一物；要么是害人。

### ****七.【宗师】****

这个级别的人已然到了无招胜有招的境界。项目开发中的难题？没有难题！运行平台的差异？从根本上搞定！代码规范、开发模式，早已经被抛在身后。这个级别的人已经**不再关注于某个前端开发框架，而是应对具体的环境给出最佳的理论指导。**

这个级别的人所注意的应该是以最合理的系统架构引领着整个团队的进步，在什么样的场景下该用什么样的架构设计。3个、10个、50个、100个人的团队最应该用哪种模式？等你到了宗师级别，你再来回答吧。

【进阶出路】

每一个宗师就是一个高山，就是一个领域里的神，但是你仅满足于在一群比你弱的群体展现你的强大吗？如果还你是止步原地，那总会有人乘着飞机、宇宙飞船从你的头领掠过，高处不胜寒！

要突破这片领域，那就必须跳出这片领域。要想突破WEB前端研发的宗师级，那就跳出WEB前端吧，上面还有WEB开发。即使你是WEB前端的宗师， 但没有快速的数据响应，没有高速的网络架构，没有优美的系统支持，你又能如何？所以突破之路就是把目光投到整条WEB开发的链条中去。

### ****八.【飞升】****

其实严格来说，飞升已经不是原领域的范围了。在WEB研发领域，对于这个层次的有一个很好听的称谓：架构师。当然那些“伪架构师”另当别论。

一法通，万法通。在其它的技术领域，也可以按照《入门》《登堂》《入室》《入微》《化蝶》《大侠》《宗师》来划分等级；一样也可以按照我这里所写的每个级别的**【进阶之路】**来快速提升。

祝贺你再获辉煌！

黄方荣

2011/01/30

原文链接：<http://hszy00232.blog.163.com/blog/static/430227532011119105237357/>