《JavaScript高级程序设计》

图书基本信息

书名：《JavaScript高级程序设计》

13位ISBN编号：9787115152091

10位ISBN编号：7115152098

出版时间：2006年9月

出版社：人民邮电出版社

作者：Nicholas C. Zakas

页数：670

译者：曹力,张欣

《JavaScript高级程序设计》

内容概要

JavaScript最新经典教程 * Amazon超级畅销书 * AJAX程序员必备

随着国内的计算机图书市场越来越细化，各类引进版和原创图书在各自领域内如雨后春笋般展露头角，各类“经典”和“圣经”横空出世。但是有一个领域内却一直遗留了大片空白，就是JavaScript类图书。除了O'Reilly的《JAVASCRIPT权威指南》之外，国内一直没有出现过特别优秀的同类图书。就在这时，Wrox的《Professional JavaScript for Web Developers》走进了我们的视野，中文名是《JAVASCRIPT高级程序设计》

JavaScript是目前Web客户端开发的主要编程语言，也是Ajax的核心技术之一。本书从最早期Netscape浏览器中的JavaScript 开始讲起，直到当前它对XML和Web服务的具体支持，内容主要涉及JavaScript的语言特点、JavaScript与浏览器的交互、更高级的 JavaScript技巧，以及与在Web应用程序中部署JavaScript解决方案有关的问题，如错误处理、调试、安全性、优化/混淆化、XML和 Web服务，最后介绍应用所有这些知识来创建动态用户界面。
本书适合有一定编程经验的开发人员阅读，也可作为高校相关专业课程的教材。

新增DOM概念,如何实现正则表达式进行数据验证和字符串操作,把JavaScript联系到Web用户界面的事件处理方法； 数据验证、表排序和错误处理的方法等

目录

《JavaScript高级程序设计》

作者简介

Nicholas C.Zakas世界知名的JavaScript专家和Web开发人员。Nicholas拥有丰富的Web开发和界面设计经验，曾经参与许多世界大公司的Web解决方案开发，并与他人合作撰写了畅销书《Ajax高级程序设计》。

《JavaScript高级程序设计》

书籍目录

《JavaScript高级程序设计》

媒体关注与评论

书评“ 如果你像我一样，想学习或者熟练掌握今天最热门的Web开发技术，本书是一个绝佳的起点，适合在所有Ajax图书之前的阅读。 ”　　——J.Ambrose Little,Microsoft MVP　“ 本书作者显然非常了解读者的需要，切中要害，信息密集。单单对客户端通信、Web服务、正则表达式、DOM、XML处理等现代JavaScript技术的详细讲解，就已经物超所值。 ”　　——JavaScriptKit.com　JavaScript作为赋予网页活动与交互性的主要手段之一，早已经成为Web设计师和开发人员的必备技能。全世界无数网页每天都在依靠JavaScript完成各种关键任务。然而，JavaScript可能也是被人误解和误用最多的主流编程语言。很多人将它看作Java等面向对象编程语言的功能不全的小兄弟，甚至贬为雕虫小技，对它不屑一顾。　如今，随着越来越多的程序员转向浏览器/服务器模式开发，更加上Web 2.0和Ajax的兴起，JavaScript已经被推到了舞台中心。人们开始认识到，JavaScript绝非一种容易学习和掌握的技术，它同时具有面向对象、过程和函数型语言三类语言的特性，将灵活性与强大功能完美结合。迄今为止，它的惊人潜力还远远没有真正释放出来。　本书针对开发人员和有经验的Web设计师撰写，在简明扼要地讲述了JavaScript的语言核心ECMAScript，以及面向对象特性、BOM、DOM之后，很快转向高级主题：正则表达式、事件、数据验证、表排序、施放、错误处理、调试、XML、Web服务、安全、国际化、优化和知识产权保护，能够解决Web开发者目前面对的各种迫切问题。

《JavaScript高级程序设计》

编辑推荐

《Java Script高级程序设计》适合有一定编程经验的开发人员阅读，也可作为高校相关专业课程的教材。

《JavaScript高级程序设计》

精彩短评

1、很不错的一本js书，感谢作者的辛苦写作
2、很不错，对javascript和dom做了非常详细的介绍，只是书比较旧了。据说高程3快出来了，值得期待
3、已过时，不要再看这本，去看第三版吧。JS的东西真是“一日不见，如隔三秋”啊
4、虽然说是高级程序设计，但是我以为当作第一本入门书籍也未尝不可，因为写得挺浅显的。当然有的高深的章节，例如Ajax，HTML 5可能对于新手来说暂时用不到。对于每个api，每个浏览器的支持程度都讲得很详细。感觉各浏览器标准的不兼容平白无故的给前端工程师增加了好多工作量。如果能够使用编辑器编写代码，调试的操作说明，就更好了。是不是应该提及一下例如jQuery等JavaScript的增强呢？
5、最好的js书之一.不解释.
6、把Javascript的书都差不多看完了，回头发现就这一本才是正道。不过Javascript依然是难以理解的 - -!
7、好书
8、不喜欢
9、JS系Q2 读的第一版 第3、4章不错。 PS: 有曹力大拿啊！~
10、js这门语言真是让人恨，不知以后会不会爱上它
11、看了主要知道有这么个东西，到用时可以想起。
12、对于想学JS的人来说，相当好的书啊!但是对于翻译的书我总是觉得有点别扭
13、还行，把js的基础过了一遍
14、在我眼里，这是最好的javascrpt书，没有之一。
15、学习 JavaScript 很全面的一本书，第一遍看的粗，准备再来一遍。
16、怎么到现在第二版还没引进啊！出版社动作快点啊！希望赶快引进第二版的影印版，我可不想对着那些勘误列表来修改错误，尤其是翻译错误！还有，一定不要跟这书一样把索引给X了，查个东西很麻烦！
17、面试吃了亏，才想起来要钻研深一点，这本书讲的很基础，也很清楚，适合用于打好js语法，DOM，AJAX的基础
18、必备
19、4年前读书这本书，当时自己正在学习前端技术，对这种东西懵懵懂懂，现在回想，这确实一本书不错的基础书！
20、讲语法的，还行
21、讲的非常好，力荐
22、相对基础的较深,不过只要研究一番,还是有突破的!
23、这本书非常好，我前后看了4遍了
24、本书对javascript的语法和用法都提供了很多的说明，对模式也稍有带过，不过这本书实在是太厚了，读了两遍，现在都还不是很记得了，读这本书可以对javascript有很好的了解，推荐！
25、还全面的一本书
26、So ... 还得再来一遍~
27、入门必看
28、JavaScript高级程序设计要看红皮版的
29、面向对象的理解。。看了觉得受益匪浅
30、经典！
31、不错的JS进阶教程
32、最近重读了一遍。
33、学一门语言需要一个镐头书, 这就是
34、电子
35、Java Script高级程序设计不错，讲解很到位
36、买了挺久了，一直忘了评价。原本对这本书期待值挺高的，或许跟我目前水平有关吧。不过卓越的这本书质量还不错。

37、有深度
38、作者详细深入的解释javascript这门语言，对于很多概念都用很长的篇幅说明，但是本书很多概念较深，不适合初学者阅读，建议有一定javascript基础的朋友再作阅读
39、让自己的JS水平提高了不少，结合同时期在看的ajax in action，对前段有了一定的了解
40、很细致入味的讲解
41、翻译很好，经典的经典
42、不错的书，不过没有一本书是可以完全写完整的。对于这本书写成这样很不错了。
43、棒极了，看这个学js的，超级棒，还学了正则表达式
44、很多都是过时的内容，所以买了第二版哎
45、值得多次研读
46、不错，挺喜欢的，东西挺好
47、书是不错，内容也很详细，就是后面几章有点鸡肋。
48、入门大全，不解释
49、还不错的书~
50、不错！推荐！
51、中规中矩的书，很实用。
52、还好吧，外国人的书的通病，有点罗嗦翻译的还可以，起码不像有的书翻译成中文愣是看不懂总体来说还是不错的
53、东西很好，送货也很快，第二天就送到了
54、讲的挺深入，也挺通俗。
55、大部分段落是选读的，不过很多段落都看了好多遍。对于整个DOM结构叙述很清楚，DOM对象、继承方式等
56、书买了好几年了，看起来翻过很多次的样子，只是我不记得了。。现在重新看看。。
貌似内容很旧了，翻了一遍，决定再入第三版。。
57、内容较旧
58、花了两天看完，因为最近一直在碰css，所以想深入学习一下，入了几本书，这本虽有高级俩字，不过却是新手看的，看完后还是对很多东西更加明晰。
59、适合入门
60、极力推荐该作者的书
61、发货速度 很快啊 书的质量也还不错，
62、不喜欢翻译版
63、很好很强大，里面有些东西要反复思考才是
64、当读完这本书的时候，我发现我还是一头雾水~~~真正学会使用还是在后来的项目中历练~~Jquery+Css，才明白的Js的强大~
65、有Java基础、学习JavaScript的不二之选。

《JavaScript高级程序设计》

精彩书评

1、我郁闷，从DOM部分开始，因为兼容的问题，基本都是一个东西分成两节来讲，--！那么多的API，那么多的兼容问题，我不知道到何年何月才能够记下来。。。。看了半年多了，目前看到十三章表单部分，每天固定两小时，看得我是两眼直冒金星。。。值得庆幸的是大部分内容还能看懂~。。
2、值得静下来好好读的一本书，会加深对javascript的理解，最重要的是写的很容易让人理解，一段说明，然后配上样例代码，加深理解。很少有书能让我有想把书完全看完的念头，这本书不错，如果想好好加深一下javascript的理解，看一下这本书吧！
3、无论怎样，js都是前段开发人员最头疼的问题，兼容，兼容，再兼容，然后这本书耗费了很多笔墨在浏览器兼容性方面的差异阐述得非常仔细，当然这本书也涉及到了很多js的初级知识，其实，这本书不应该叫高级程序设计，把高级二字去掉更好。
4、书籍太多了,我们反而不能每本都详读。如果你需要全面的了解JavaScript,知道JavaScript的语法,DOM,正则表达式,事件,浏览器差异等等,那么仅仅看这本书就够了,剩下的就是需要一个api手册和不停的实践了。
5、一本算是通俗，但是不一定是循序渐进的书。你必须具备一定的js基础，或者需要具备一定的开发经验，才能够读懂或者比较详尽地理解。甚至有时候一遍过去还不一定能马上理解，需要再回过头来，再重新看某个章节，才能理解的这样一本书。不过对我来说意义不小。之前没有完整看过一本JS的书籍。过去的开发过程中，都是从网上零星获取即时需要的js知识，用查找到的知识马上进行实践。这一次看这本书，很多时候，我会恍然大悟，哦～～～，终于知道我以前为什么要这么写或者那么写才正确。前面4章对用JS进行模拟面向对象开发进行了很详尽细致的讲述。接下来的每一章都可以在以后的开发中再回过头来当作手册去查阅。一句话，我觉得是一本值得推荐的JS书。
6、语言非常简练、易懂，内容却很丰富，从JS的起源一直到对JS的未来展望（关于ECMAScript第四版）,一路走来，你的JS水平会在不知不觉中得到提高，尤其书中有很多例子写的非常实用，可以直接拿来应用到你的项目应用中，而且都是符合标准的支持多个浏览器，从这些简单实用的代码中可以学习到很多知识，比纯文字传授表达更加贴近你的真实思想，所以一定要认真读懂！！！
7、看过原版后看的译本，书绝对是本好书，翻译的也不错，对DOM和XML部分讲解比较深入，对JS进阶者来说，是个不错的选择。
8、这本书非常强调用OPP概念来写js。内容很多，够深的，看后觉得搞web标准的开发真是痛苦，应该说ECMAScript本身是不错的，但是，，，，所以从某些人的角度来说firefox和safair的崛起并不是好事。PS：不看懂这本书，《ajax实战》就别指望的
9、其实基础的东西讲的不多，高级的东西也不是很靠谱，而且是06年的书，对于互联网来说，有点旧了。关于ajax的东西基本没有，那是好像还不叫ajax，总之翻了翻，也就放下了，不是很实用...
10、如果我SAY這書比較淺，大家不會罵我吧？其實我只是覺得不太合適我看，我看開都是比較低版本的，用的是9i，這本書之所以現在還是在我家墊桌腳的最大原因哈。這是我比較給力厚道的一本書了。再有就是希望書中能多點圖說明就更好了。
11、个人感觉有点像做过山车的感觉，并不是建议有JavaScript基础的人看，应该是建议有编程基础的人看。一开始讲的是编程的基础问题，JavaScript的基础语法，马上就会谈到设计模式，和标准的兼容浏览器代码。中间涉及的很多概念，是必须要有编程经验的人才能读懂它的精髓。就书籍的涉及面和深入程度来说，此书是JavaScript进阶的必备书籍。此书我看了2遍了，准备再读一遍，此书作为JavaScript的标准教材，一点都不过分。
12、如果想深入的学习 js 买回来研读这本书可以帮助你学习js，这本书 我看了 四分之三，就没有看下去，觉得，对于现在的我，只要掌握这些，就可以了，毕竟如果你想完成手上的工作， 类似 jQuery，mootools，或者prototype这样的类库可以又快又好的帮组你做好工作。当然，如果你抱着研究语言的态度去看的，令当别论。
13、真正的想要精通JS实践才是王道，努力的自己试着去多多的写一写js，绝对会有事半功倍的效果的！
14、出版的时间比较长了，浏览器兼容的内容有些过时了，没有IE8，没有chrome。不过第二版里边都更新了，而且第二版整书的章节设置也有变化。在考虑要不要把第二版再看一下。
15、OKJavaScript高级程序设计JavaScript高级程序设计JavaScript高级程序设计

16、无论是对于初学还是进阶，从这本书中你都可以找到你想要的东西~每读一遍你都会被js有跟深入的了解~和应用~
17、从简单的地方到复杂的地方过度不是很好，讲解dom，和ecmascript基础的时候都比较简单，但一下就到了正则和排序等比较高级技术，而且有些东西让初学者一头雾水。总体来说翻译上有些地方让人觉得变扭，其他还不错，不过确实不容易看懂，必须先了解机制，然后再学 有个问题，这本是实战性不太强，大多数都只是介绍下就带过，有些东西但又很深入，讨论过多只是为了兼容性，没有切实可行的说明工作中会面临的问题，比如date对象，几乎是一笔带过，所以对于提高来说很不错，对于初学可能需要不少功夫
18、美好的语言，糟糕的实现。为了商业利益，置标准为草芥，浏览器大战的后遗症，就是再好的语言，它们都给你最糟糕的实现。这本书在一定程度上是解决浏览器之间的差异，并使用趋近“ 标准 ”的代码来教学。值得一读。
19、招聘兼职讲师一本好书可以帮助人成长，一套课程可以教会人一门技术。你的经验分享，也许可以帮助上万人，赢得上万个学生。我们源智天下科技有限公司，是一家定位于计算机图书创作和计算机技术教学的内容提供商。在过去的几年里，我们每年出版100多本专业的计算机图书，其中绝大多数都获得了比较好的销售成绩。同时，我们启动了一个网上的IT技术教学网站，融智技术学苑（www.rzchina.net），正通过互联网络，让我们的老作者给我们的百万读者提供增值服务。运行了一段时间后，我们的读者纷纷建议，希望开设一个项目开发频道，讲解如何从零开始，完成一个完整的IT项目的开发。所以，如果你有三年以上实际IT项目开发经验，并且善于讲解某项技术的应用，欢迎你能来总结你的开发经验，和我们的读者分享。请和我们联系合作。具体流程如下：（1）规划你要讲座的项目，发E-mail到：zhangjing8526@163.com（2）我们对你的课程满意的话，会发给你详细的报酬方案，已经如何录制视频的方案。（3）项目通过后，设计项目讲座PPT，签订合作合同；（4）验收视频合格后，支付基本报酬。（5）视频介绍上线，开始销售（6）每季度，根据销量结算一次教学报酬。

《JavaScript高级程序设计》

章节试读

1、《JavaScript高级程序设计》的笔记-第61页

正在找array的简写形式[]
第三行一定要使用括号

2、《JavaScript高级程序设计》的笔记-第20章 最佳实践（未完）

变量名应为名词，如car或person
函数名以动词开始，如getName()
返回布尔值得函数一般以is开头，如isEnable()
变量和函数名都应使用合乎逻辑的名字，不要担心长度；长度问题可以通过后处理和压缩来缓解。

变量类型应一目了然：
1）通过初始化指定变量类型（函数声明中的函数参数得用匈牙利法或者类型注释）
2）匈牙利法 bFound, iCount, sName, oPerson，一定程度上不易阅读
3）类型注释

……

3、《JavaScript高级程序设计》的笔记-第一章:JavaScript的实现

一个完整的javascript实现应该由下列三个部分组成:

核心(ECMAScript)-由ECMAScript-262定义,提供核心语言功能
文档对象模型(DOM)-提供访问和操作网页内容的方法和接口
浏览器对象模型(BOM)-提供与浏览器交互的方法和接口

ECMAScript

由ECMA-262定义的ECMAScript与web浏览器没有依赖关系.浏览器只是ECMAScript实现的宿主环境之一而已.宿主环境不仅提供基本的ECMAScript实现,同时也会提供该语言的扩展,以便语言和环境之间对接交互.而这些扩展-DOM,则利用ECMAScript的核心类型和语法提供更多更具体的内容,以便实现针对环境的操作.

既然ECMAScript-262标准没有参照web浏览器,那它都规定了些什么内容呢?大致来说,它规定了这么语言的写了组成部分:

语法//语言的用法
类型//数据类型
语句
关键字//有特殊用途的一些单词
保留字//将来可能有特殊用途的一些单词
操作符//操作数据值的符号
对象//你就是对象:)

文档对象模型(DOM)

文档对象模型(DOM)是针对XML但经过扩展用于HTML的应用程序编程接口(API).DOM把整个页面映射成一个多层节点结构.HTML或XML页面的每个组成部分都是某种类型的节点,这些节点有保护着不同类型的数据.看下面这个HTML页面
<html>//根节点,一个文档的最顶层节点
  <head>//html节点的儿子，body节点的兄弟
    <title>一个基本的网页(HTML文档)</title>//head节点的儿子,html节点的孙子
  </head>
  <body>//html节点的儿子,head节点的兄弟
    <p>你好呀:)</p>//body节点的儿子,html节点的孙子
  </body>
</html>
浏览器对象模型(BOM)

浏览器对象模型(BOM)就是可以对浏览器窗口进行访问和操作的功能,它至今没有相关标准,受限于浏览器的实现.

从根本上讲,BOM只处理浏览器窗口和框架,但人们习惯上也把所以真的浏览器的JavaScript扩展算作BOM的一部分.下面是一下这样的扩展:

  弹出新窗口的功能
  移动,缩放和关闭浏览器窗口的功能
  提供浏览器详细信息的navigator对象
  提供浏览器所加载页面的详细信息的location对象
  提供用户显示器分辨率详细信息的screen对象
  对cookies的支持
  想XMLhttpRequest和IE的ActiveXOBject这样的自定义对象

4、《JavaScript高级程序设计》的笔记-第229页

事件处理函数：
1、IE处理方式：attachEvent/detachEvent
2、DOM处理方式：addEventListener/removeEventListener

事件对象：
1、引发事件的对象；
2、事件发生时鼠标的信息；
3、事情发生时键盘的信息；

引发事件的对象：
```
IE: oDiv.onclick = function(){
  var oEvent = window.event;
}DOM:oDiv.onclick = function(){
  var oEvent = arguments[0];
}
//OR:
oDiv.onclick = function(oEvent){
  //...
```

}

事件类型:var sEventType = oEvent.type;按键代码:var iKeyCode = oEvent.keycode;特殊键:var bShift = oEvent.shiftKey;
var bAlt = oEvent.altKey;
var bCtrl = oEvent.ctrlKey;
事件类型分类:
鼠标事件：click/dbclick/mousedown/mouseout/mouseover/mouseup/mousemove
键盘事件：keydown/keyup/keypress(专指字符按键)
HTML事件：
突变事件：

5、《JavaScript高级程序设计》的笔记-第22页

6.引用类型，通常叫做类，处理的是对象。对象由new运算符加上要实例化的类的名字创建，如ar 0 = new object()。
object类：自身用处不大，所有类都由这个类继承而来。
属性有，constructor——对创建对象的函数的引用；
prototype——对该对象的对象原型的引用。
方法有，hansOwnPrototype(prototype)——判断对象是否有某个特定的属性；
isPrototypeOf(object)——判断该对象是否为另一个对象的原型；
prototypeIsEnumerable(prototype)——判断给定的属性是否可以用for...in语句进行枚举；
toString()——返回对象的原始字符串表示；valueOf()——返回最适合该对象的原始值。上述属性和方法都会被其他类覆盖。
Boolean类：Boolean原始值的引用类型，var oBooleanObject = new Boolean(true); 覆盖object类的valueOf()方法和tostring()方法返回原始值（true或false）。
Number类：var oNumberObject = new Number(55); 要得到其原始值，只需使用valueOf()方法。
一些专用方法：toFixed()方法——返回具有指定位数（0-20位）小数的数字的字符串表示，如oNumberObject.toFixed(2); 返回55.00。
toExponential()——返回用科学记数法表示的数字的字符串形式，如oNumberObject.toExponential(1); 返回9.9e+1表示5.5*10（1）。
toPrecision()方法——返回数字的预定形式或指数形式。
String类：var oStringObject = new String("hello world!"); 具有属性length,
方法：charAt() 和charCodeAt()——访问字符串中的单个字符，区别在于前者得到字符，后者得到字符代码；
concat()——把一个或多个字符串连接到String对象的原始值上，功能和+差不多；
indexOf()和lastIndexOf()——判定在一个字符串中是否包含某个字符，返回的都是指定的字串在另一个字符串中的位置（或-1如果没找到）。前者从字符串开头开始检索，后者从结尾开始检索；
localeCampare()——对字符串值进行排序，若string对象按照字母排序排在参数中的字符串之前就返回负数，等于参数中的字符串就返回0，排在之后就返回正数（区分大小写，大写字母在小写字母之后）；
slice()和substring()——返回要处理的字符串的子串。如oStringObject.slice(3); oStringObject.substring(3); //返回"lo world!" oStringObject.slice(3, 7); oStringObject.substring(3, 7); //返回“ low ”。两者只有在处理负数参数的时候不同，前者会用字符串的长度加上参数，而后者将其作为0处理；
toLowerCase()和toUpperCase()——把字符串转换成全部小写和全部大写；
toLocaleLowerCase()和toLocaleUpperCase()——基于特定区域实现的大小写转换（一般来说，如果不知道在以那种编码运行一种语言，则使用区域特定的方法比较安全）。

7.instanceof运算符，与typeof运算符相似用于识别正在处理的对象的类型，不同的是instanceof方法要求开发者明确的确认对象为某特定类型，如typeof oStringObject返回object， oStringObject instanceof string返回true，虽不如typeof方法灵活，但管用。

8.运算符：
一元运算符：只有一个参数。
delete——删除对以前定义的对象属性或方法的引用，如delete o.name，但不能删除对象尚未定义的属性或方法。
void——对任何值都返回undefined，通常用于避免输出不应该输出的值，如html里的一个Click Me被点击后即可看到屏幕上显示[object]，这是因为window.open返回了对新打开的窗口的引用，该对象将被转换成要显示的字符串，要避免这种情况就要使用void运算符调用window.open()函数: Click Me，此时函数返回的是undefined，不是有效的值，不会显示在浏览器窗口中。
前增量/前减量/后增量/后减量——
一元加法/一元减法——一元加法对数字无任何影响，但会将字符串转换成数字（方法与parserInt()相似，不要不同为一元运算符只会将0x开头的字符串转换成十进制的值，而对二进制或者八进制的数都会直接得到其数字部分的字面量）。一元减法是对数值求负，也可以将字符串转换成数字并求负值。
位运算符：每个整数都可以表示成二进制形式的32位数值，前31位表示数值，第32位表示符号，0为正数，1为负数。
not——由否定号~表示，步骤：把运算数转换成32位数字，把二进制形式转换成它的二进制反码，把二进制反码转换成浮点数。实质上是对数字求负然后减1。
and & ——
or | ——
XOR ^ ——
左移运算 << ——
有符号右移运算 >> ——
无符号右移运算 >> ——
(未完)

6、《JavaScript高级程序设计》的笔记-第1页

完整的Javascript包括核心ECMAScript, DOM和BOM。
ECMAScript：
ECMAScript描述了语法，类型，语句，关键字，保留字，运算符和对象。ECMAScript仅仅是一个描述，定义了脚本语言的所有属性、方法和对象。
DOM：
DOM是HTML和XML的应用程序接口（API）。DOM将把整个页面规划成由节点层级构成的文档。DOM通过创建树来表示文档。
DOM Level1由DOM core和DOM HTML构成。目标：规划文档的结构。
DOM Level2引入了DOM视图、DOM事件、DOM样式、DOM遍历和范围。扩展添加了对鼠标和用户界面事件、范围、遍历的支持，并通过对象接口添加了对CSS的支持。
DOM Level3引入了DOM Load和DOM Save，以统一的方式载入和保存文档。
目前Mozzila具有最好的DOM支持。
BOM：
BOM主要处理浏览器窗口和框架，有以下操作：
1.弹出新的浏览器窗口。
2.移动、关闭浏览器窗口以及调整窗口大小。

3.提供web浏览器详细信息的导航对象。
4.提供装载到浏览器中页面的详细信息的定位对象。
5.提供用户屏幕分辨率详细信息的屏幕对象。
6.对cookie的支持

7、《JavaScript高级程序设计》的笔记-第9页

1. 弱类型语言的同一变量可以存放不同类型的值。但在使用变量时，好的编程习惯是始终存放相同类型的值。
2. 变量可以存放两种类型的值，原始值和引用值。
原始值：存储在栈中的简单数据段，即，它们的值直接存储在变量访问的位置。有5中类型——Undefined、Null、Boolean、Number和String。（typeof运算符对于引用类型和null值都会返回object，null被认为是对象的占位符。typeof可以用于未定义的变量也不报错，将输出undefined。）
Undefined：变量未声明过，变量未初始化，以及函数无明确返回值时，其值均为undefined。
Null：值undefined是从值null派生出来的，因此在ECMAScript中把它们定义为相等，即null == undefined。但它们的含义不同，null用于表示尚未存在的对象。
Boolean：0和false之间可以相互转换。
Number：可以表示整数，浮点数，十进制，八进制(字面量的首数字必须是0)，十六进制（字面量首数字必须是0，后接x）等。函数isFinit()判断是否是无穷大的数。特殊值NaN，表示非数，它与自身不相等。函数isNaN()判断是否是非数。
String：\n换行，\t制表符，\b空格，\r回车，\f换页符，\\反斜杠，\'单引号，\“ 双引号，\0nnn八进制代码nnn（n是0到7中的一个八进制数字）表示的字符，\xnn十六进制代码nn（n是0到f中的一个十六进制数字）表示的字符，\unnnn十六进制代码nnnn表示发unicode字符。
3.转换成字符串：Boolean值、数字和字符串是伪对象，意味着它们具有属性和方法，如获得长度str.length。它们都有toString()方法。Boolean的toString()方法返回“ true ”或“ false ”；Number的toString()方法有默认模式和基模式两种，默认模式下返回都是数字的十进制表示，基模式可以用不同的基输出数字，例如二进制的基是2，number.toString(2)。
4.转换成数字：两种方法——parseInt()和parseFloat()。只能对string类型调用这些方法，否则返回NaN。
parseInt()方法从位置0处的字符开始查看，判断它是否是个有效数字，直到找到非有效数字的字符为止。parseInt()也有基模式，可以将二进制、八进制、十六进制或其他任何进制的字符串转换成整数，如要解析十六进制的值，需parseInt("AF", 16);
parseFloat()方法处理方式与parseInt()相似，第一个出现的小数点有效，后面再有小数点就无效。字符串必须以十进制形式表示浮点数。parseFloat()没有基模式。
5.强制转换：Boolean(value),Number(value), String(value)
当要转换的值是至少有一个字符的字符串、非0数字或对象时Boolean()函数将返回true，否则返回false。
Number()转换整个值，如果字符串值能被完整转换，Number()将判断是调用parseInt()还是parseFloat()方法，否则将返回NaN。
String()方法和toString()方法的唯一不同之处在于，前者对null或undefined值强制转换可以生成字符串而不引发错误。

8、《JavaScript高级程序设计》的笔记-第1页

1、getelementbyid：如果指定的id匹配某个元素的name的特性，ie6.0还会返回这个元素，这是一个bug，使用时一定要小心。

2、创建和操作节点比较常用的创建方法是：createdocumentfragment,createelement,createtextnode,其他的要不不用，要么不能跨平台。

3、ie在setattribute()上有个很大的问题，当时使用他时，变更并不是总是能刷新出来，如果你使用ie，尽可能使用属性。如：div.classname="test"。

4、NodeIterator对象展示了一种有序的自顶向下遍历整个DOM树的方式。如果想遍历DOM树的特定区域，再看看某个节点的兄弟节点或子节点，可以使用TreeWalter。

9、《JavaScript高级程序设计》的笔记-第69页

标记清除。

垃圾收集器首先会把在内存中的所有变量都添加上标记。
然后当执行进入到某个环境会把环境中使用过的变量的标记清楚。

最后垃圾回收回收了那些存在标记的变量。

以前没仔细看，一直以为是回收了那些不存在标记的变量，其实这里恰恰相反。

《JavaScript高级程序设计》

版权说明