Atitit 前端 dom 的艺术 attilax著

目录

[1. 概念 1](#_Toc4709)

[2. 发展历程 1](#_Toc692)

[2.1. 厂商各自为政 2](#_Toc4835)

[2.2. 1.4　制定标准 标准化 w3cdom 2](#_Toc687)

[2.3. 1.4.1　浏览器以外的考虑 析XML文档的时候，你获得的DOM 3](#_Toc18226)

[3. DOM 　3.1 文档：DOM中的“D”　3.2 对象：DOM中的“O”　3.3 模型：DOM中的“M” 3](#_Toc7973)

[3.1. 动态创建标记 4](#_Toc17223)

[3.2. 第8章 充实文档的内容 4](#_Toc15777)

[3.3. CSS-DOM 5](#_Toc6726)

# 概念

1.2　DOM  
　　什么是DOM？简单地说，DOM是一套对文档的内容进行抽象和概念化的方法。  
　　在现实世界里，人们对所谓的“世界对象模型”都不会陌生。例如，当用“汽车”、“房子”和“树”等名词来称呼日常生活环境里的事物时，我们可以百分之百地肯定对方知道我们说的是什么，这是因为人们对这些名词所代表的东西有着同样的认识。于是，当对别人说“汽车停在了车库里”时，可以断定他们不会理解为“小鸟关在了壁橱里”。  
　　我们的“世界对象模型”不仅可以用来描述客观存在的事物，还可以用来描述抽象概念。例如，假设有个人向我问路，而我给出的答案是“左边第三栋房子”。这个答案有没有意义将取决于那个人能否理解“第三”和“左边”的含义。如果他不会数数或者分不清左右，则不管他是否理解这几个概念，我的回答对他都不会有任何帮助。在现实世界里，正是因为大家对抽象的世界对象模型有着基本的共识，人们才能用非常简单的话表达出复杂的含义并得到对方的理解。具体到这里的例子，你可以相当有把握地断定，其他人对“第三”和“左边”的理解和我完全一样。  
　　这个道理对网页也同样适用。JavaScript的早期版本向程序员提供了查询和操控Web文档某些实际内容（主要是图像和表单）的手段。因为JavaScript预先定义了“images”和“forms”等术语，我们才能像下面这样在JavaScript脚本里引用“文档中的第三个图像”或“文档中名为‘details’的表单”：  
　　现在的人们通常把这种试验性质的初级DOM称为“第0级DOM”（DOM Level0）。在还未形成统一标准的初期阶段，“第0级DOM”的常见用途是翻转图片和验证表单数据。Netscape和微软公司各自推出第四代浏览器产品以后，DOM开始遇到麻烦，陷入困境。

# 发展历程

## 厂商各自为政

　本书第1版面世的时候，做一名Web设计师是件很让人很兴奋的事。5个年头过去了，这个职业依然保持着强大的吸引力。特别是JavaScript，经历了从被人误解到万众瞩目的巨大转变。Web开发呢，也已从混乱无序的状态，发展成一门需要严格训练才能从事的正规职业。无论设计师还是开发人员，在创建网站的过程中都积极地采用标准技术，Web标准已经深入人心。  
　　当网页设计人员谈论起与Web标准有关的话题时，HTML（超文本标记语言）和CSS（层叠样式表）通常占据着核心地位。不过，W3C（万维网联盟）已批准另一项技术，所有与标准相兼容的Web浏览器都支持它，这就是DOM（文档对象模型）。我们可以利用DOM给文档增加交互能力，就像利用CSS给文档添加各种样式一样。  
　　在开始学习DOM之前，我们先检视一下使网页具备交互能力的程序设计语言。这种语言就是JavaScript，它已经诞生相当长的时间了。

接下来，只需利用JavaScript改变myelement元素的left和top样式，就可以让它在页面上随意移动。不过，这只是理论而已。  
　　不幸的是，NN 4和IE4浏览器使用的是两种不兼容的DOM。换句话说，虽然浏览器制造商的目标一样，但他们在解决DOM问题时采用的办法却完全不同。  
　　1.3.2　浏览器之间的冲突  
　　Netscape公司的DOM使用了专有元素，这些元素称为层（layer）。层有唯一的ID，JavaScript代码需要像下面这样引用它们：  
　　而在微软公司的DOM中这个元素必须像下面这样引用：  
　　这两种DOM的差异并不止这一点。假设你想找出myelement元素的left位置并把它赋值给变量xpos，那么在NetscapeNavigator 4浏览器里必须这样做：  
　　而在IE 4浏览器中，需要使用如下所示的语句才能完成同样的工作：  
　　这就导致了一种很可笑的局面：程序员在编写DOM脚本代码时必须知道它们将运行在哪种浏览器环境里，所以在实际工作中，许多脚本都不得不编写两次，一次为NetscapeNavigator，另一次为IE。同时，为了确保能够正确地向不同的浏览器提供与之相应的脚本，程序员还必须编写一些代码去探查在客户端运行的浏览器到底是哪一种。  
　　DHTML打开了一个充满机会的新世界，但想要进入其中的人们却发现这是个充满苦难的世界。因此，没多久，DHTML就从一个大热门变成了一个人们不愿提起的名词，而对这种技术的评价也很快地变成了“宣传噱头”和“难以实现”。

## 1.4　制定标准 标准化 w3cdom

　　就在浏览器制造商以DOM为武器展开营销大战的同时，W3C不事声张地结合大家的优点推出了一个标准化的DOM。令人欣慰的是，Netscape、微软和其他一些浏览器制造商们还能抛开彼此的敌意而与W3C携手制定新的标准，并于1998年10月完成了“第1级DOM”（DOMLevel 1）。  
　　回到刚才的例子，我们已经用  
　　标签定义了一个ID为myelement的页面元素，现在需要找出它的left位置并把这个值保存到变量xpos中。下面是使用新的标准化DOM时需要用到的语法：  
　　乍看起来，这与刚才那两种非标准化的专有DOM相比并没有明显的改进。但事实上，标准化的DOM有着非常远大的抱负。  
　　浏览器制造商们感兴趣的只不过是通过JavaScript操控网页的具体办法，但W3C推出的标准化DOM却可以让任何一种程序设计语言对使用任何一种标记语言编写出来的任何一份文档进行操控。

## 1.4.1　浏览器以外的考虑 析XML文档的时候，你获得的DOM

　　DOM是一种API（应用编程接口）。简单地说，API就是一组已经得到有关各方共同认可的基本约定。在现实世界中，相当于API的例子包括（但不限于）摩尔斯码、国际时区、化学元素周期表。以上这些都是不同学科领域中的标准，它们使得人们能够更方便地交流与合作。如果没有一个统一的标准，事情往往会演变成为一场灾难。别忘了，因混淆英制度量衡与公制度量衡至少导致过一次火星探测任务的失败。  
　　在软件编程领域中，虽然存在着多种不同的语言，但很多任务却是相同或相似的。这也正是人们需要API的原因。一旦掌握了某个标准，就可以把它应用在许多不同的环境中。虽然语法会因为使用的程序设计语言而有所变化，但这些约定却总是保持不变的。  
　　因此，虽然本书的重点是教会你如何通过JavaScript使用DOM，当你需要使用诸如PHP或Python之类的程序设计语言去解析XML文档的时候，你获得的DOM新知识将会有很大的帮助。  
　　W3C对DOM的定义是：“一个与系统平台和编程语言无关的接口，程序和脚本可以通过这个接口动态地访问和修改文档的内容、结构和样式。”W3C推出的标准化DOM，在独立性和适用范围等诸多方面，都远远超出了各自为战的浏览器制造商们推出的各种专有DOM。

# DOM 　3.1 文档：DOM中的“D”　3.2 对象：DOM中的“O”　3.3 模型：DOM中的“M”

　3.4 节点  
　　3.4.1 元素节点  
　　3.4.2 文本节点  
　　3.4.3 属性节点  
　　3.4.4 CSS  
　　3.4.5 获取元素  
　　3.4.6 盘点知识点  
　3.5 获取和设置属性  
　　3.5.1 getAttribute  
　　3.5.2 setAttribute

## 动态创建标记

1. 7.1 一些传统方法  
   　　7.1.1 document.write  
   　　7.1.2 innerHTML属性  
   　7.2 DOM方法  
   　　7.2.1 createElement方法  
   　　7.2.2 appendChild方法  
   　　7.2.3 createTextNode方法  
   　　7.2.4 一个更复杂的组合  
   　7.3 重回图片库  
   　　7.3.1 在已有元素前插入一个新元素  
   　　7.3.2 在现有方法后插入一个新元素  
   　　7.3.3 图片库二次改进版  
   　7.4 Ajax  
   　　7.4.1 XMLHttpRequest对象  
   　　7.4.2 渐进增强与Ajax  
   　　7.4.3 Hijax  
   　7.5 小结

## 第8章 充实文档的内容

1. 8.1 不应该做什么  
   　8.2 把“不可见”变成“可见”  
   　8.3 内容  
   　　8.3.1 选用HTML、XHTML还是HTML5  
   　　8.3.2 CSS  
   　　8.3.3 JavaScript  
   　8.4 显示“缩略语列表”  
   　　8.4.1 编写displayAbbreviations函数  
   　　8.4.2 创建标记  
   　　8.4.3 一个浏览器“地雷”  
   　8.5 显示“文献来源链接表”  
   　8.6 显示“快捷键清单”  
   　8.7 检索和添加信息  
   　8.8 小结

## CSS-DOM

1. 9.1 三位一体的网页  
   　　9.1.1 结构层  
   　　9.1.2 表示层  
   　　9.1.3 行为层  
   　　9.1.4 分离  
   　9.2 style属性  
   　　9.2.1 获取样式  
   　　9.2.2 设置样式

《JavaScript DOM编程艺术(第2版)（html+css+javascript教程精粹，JavaScrip权威指南！）》(（英）基思(Keith)【简介\_书评\_在线阅读】 - 当当图书.html