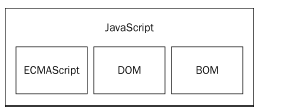
Javascript简介

# JavaScript简史

JavaScript诞生于1995年，开始只是作为简单的输入验证器。随着1996年微软推出其JavaScript 实现意味着有了两个不同的JavaScript 版本：Netscape Navigator 中的JavaScript、Internet Explorer 中的JScript。为了解决两个版本的兼容性问题，1997年ECMA（European Computer Manufacturers Association 欧洲计算机制造商协会）制定了ECMA-262标准（JavaScript1.1）。

# JavaScript的组成

 JavaScript的完整实现是由：核心（ECMAScript），文档对象模型（DOM），浏览器对象模型（BOM）组成。

## ECMAScript（核心）

宿主环境：web浏览器（客户端JavaScript平台），Node（服务器JavaScript平台）。宿主环境只要提供ECMAScript的实现和提供语言的扩展，如DOM（文档对象模型）。

ECMAScript组成部分：语法，类型，语句，关键字，保留字，操作符，对象。

ECMAScript的版本：第一版本和JavaScript1.1相同。第二版本为了和ISO/IEC-16262 保持一致，没有做任何的增加修改和删除.第三版本标志着ECMAScript成为了一门真正的编程语言：增加了正则表达式，新控制语句，try-catch处理异常等。第四个版本进行了全面的修改，不仅包含了强类型变量、新语句和新数据结构、真正的类和经典继承，还定义了与数据交互的新方式。由于修改跨越太大了，被弃用了。第五版也称为3.1版本，2009年12月3日发布。增加了JSON对象、继承方法和高级属性的定义，还定义一种严格模式。

ECMAScript的兼容性：不同浏览器以及相同浏览器不同版本对ECMAScript版本支持程度不同。（IE8开始支持第五版，IE9完全支持第五版；FireFox4.0+支持第五版。Chrome1+支持第三版）

## DOM（文档对象模型：提供访问和操作网页内容的方法和接）

使用DOM的原因：无需重新加载加载页面就可以修改HTML的外观和内容（DHTML(Dynamic HTML)）。

DOM的兼容性：由于Netscape和微软公式对DHTML方面各执己见。为了解决其兼容性问题，由W3C开始制定DOM的标准。DOM1：DOM核心（用于映射XML文档结构）和DOM HTML（用于HTML对象和方法）两个模块。DOM2：扩充了 鼠标和用户界面时间、范围，遍历，样式。DOM3：引入了统一方式加载和保存文档的方法，增加了验证文档的方法等。

其他DOM标准：DOM 并不只是针对JavaScript 的，很多别的语言也都实现了DOM。例如：SVG，MathML，SMIL（都是W3C推荐的基于XML标准的DOM）。

## BOM（浏览器对象模型：提供与浏览器交互的方法和接口）

BOM用于控制浏览器显示页面以外的内容。不同的浏览器对其实现都是不同的，故存在兼容性问题。HTML5致力于把很多BOM功能写入正是的规范。

# HTML中使用JavaScript

## HTML文档模式

文档模式：主要影响CSS内容的呈现，但在某些情况下也会影响到JavaScript 的解释执行。主要有三种模式：混杂模式，标准模式和准标准模式。准标准模式又有过渡型和框架集型。准标准模式与标准模式非常接近，它们的差异几乎可以忽略不计。

### 标准模式

|  |
| --- |
| <!-- HTML 4.01 严格型 -->  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  "http://www.w3.org/TR/html4/strict.dtd">  <!-- XHTML 1.0 严格型 -->  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">  <!-- HTML 5 -->  <!DOCTYPE html> |

### 准标准模式

|  |
| --- |
| <!-- HTML 4.01 过渡型 -->  <!DOCTYPE HTML PUBLIC"-//W3C//DTD HTML 4.01Transitional//EN"  "http://www.w3.org/TR/html4/loose.dtd">  <!-- HTML 4.01 框架集型 -->  <!DOCTYPE HTML PUBLIC"-//W3C//DTD HTML 4.01 Frameset//EN"  "http://www.w3.org/TR/html4/frameset.dtd">  <!-- XHTML 1.0 过渡型 -->  <!DOCTYPE html PUBLIC"-//W3C//DTD XHTML 1.0 Transitional//EN"  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  <!-- XHTML 1.0 框架集型 -->  <!DOCTYPE html PUBLIC"-//W3C//DTD XHTML 1.0 Frameset//EN"  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd"> |

### 混杂模式

混杂模式：文档开始处如果没有发现文档类型声明，则默认是混杂模式。

## JavaScript导入方式

|  |
| --- |
| 1. js代码写入事件：<input type="button" name="test " value="确定" onclick="document.write('hello world')"/>  2. <head>中添加script元素：<script type="text/javascript"></script>  3. 导入外部js文件：<script type="text/javascript" src="test.js"></script>  注意：一般使用导入外部js方式，它具有可维护性，可缓存的特点。 |

## <script>元素及其属性

HTML 4.01 为<script>定义了下列6 个属性：

* async：可选。表示应该立即下载脚本，但不应妨碍页面中的其他操作，比如下载其他资源或等待加载其他脚本。只对外部脚本文件有效。
* charset：可选。表示通过src 属性指定的代码的字符集。由于大多数浏览器会忽略它的值，因此这个属性很少有人用。
* defer：可选。表示脚本可以延迟到文档完全被解析和显示之后再执行</html>。只对外部脚本文件有效。IE7 及更早版本对嵌入脚本也支持这个属性。
* language：已废弃。原来用于表示编写代码使用的脚本语言（如JavaScript、JavaScript1.2或VBScript）。大多数浏览器会忽略这个属性，因此也没有必要再用了。
* src：可选。表示包含要执行代码的外部文件。
* type：可选。可以看成是language 的替代属性；表示编写代码使用的脚本语言的内容类型（也称为MIME 类型）。虽text/javascript,text/  
  ecmascript 都已经不被推荐使用，但人们一直以来使用的都还是text/javascript。实际上，服务器在传送JavaScript 文件时使用的MIME 类型通常是application/x–javascript，但在type 中设置这个值却可能导致脚本被忽略。另外，在非IE浏览器中还可以使用以下值：application/javascript 和application/ecmascript。考虑到约定俗成和最大限度的浏览器兼容性，目前type 属性的值是text/javascript。不过，这个属性并不是必需的，如果没有指定这个属性，默认值text/javascript。

注意：在使用<script>嵌入JavaScript 代码时，记住不要在代码中的任何地方出现"</script>"字符串。可以通过"<\/script>"解决。

XHTML 文档中使用：<script type="text/javascript" src="example.js" />方式引入。但是不符合HTML规范，可以使用通用的方式：

<script type="text/javascript" src="http://www.somewhere.com/afile.js"></script>

带有src 属性的<script>元素不应该在其<script>和</script>标签之间再

包含额外的JavaScript 代码。如果包含了嵌入的代码，则只会下载并执行外部脚本文件，嵌入的代码会被忽略。

head中引入的CSS和JavaScript文件过多时候，页面加载和解析是按顺序执行，无疑会导致浏览器在呈现页面时出现明显的延迟，而延迟期间的浏览器窗口中将是一片空白。为了解决这个问题，将引入JavaScript放在body标签中。

|  |
| --- |
| <body>  <!-- 这里放内容 -->  <script type="text/javascript" src="example1.js"></script>  <script type="text/javascript" src="example2.js"></script>  </body> |

XHTML中嵌入JavaScript 代码时，XHTML是将HTML 作为XML 的应用而重新定义的一个标准。不支持>,<等符号。使用内嵌JavaScript代码时候，应该：

|  |
| --- |
| 对于其他类型HTML页面中应该注释掉<!CDATA[ ]]>  <script type="text/javascript"><![CDATA[  …  ]]></script> |

## <noscript>元素的使用

Noscript中的内容会在不支持JavaScript 的浏览器或者浏览器支持脚本，但脚本被禁用的情况中显示。

|  |
| --- |
| <body>  <noscript>  <p>本页面需要浏览器支持（启用）JavaScript。</p>  </noscript>  </body> |

# 项目中引入javaScript

|  |
| --- |
| <body>  ……  <script type="text/javascript" src="example.js"></script>  </body> |