软件工程师培养丛书

JavaScript技术应用

武汉厚溥教育科技有限公司　编著

“理论→总结→上机→习题”四阶段教学模式

- 理论结合实践，注重动手能力培养
- 任务驱动讲解，有效激发学习兴趣
- 典型项目案例，扎实培养专业素质
- 教学做一体化，极大提高教学效率

清华大学出版社

软件工程师培养丛书

JavaScript 技术应用

武汉厚溥教育科技有限公司　编著

清华大学出版社

北　　京

内 容 简 介

本书按照高等院校、高职高专计算机课程基本要求，以案例驱动的形式来组织内容，突出计算机课程的实践性特点。本书包含 8 章：JavaScript 简介、JavaScript 语句和函数、JavaScript 对象(一)、JavaScript 对象(二)、文档对象模型、JavaScript 事件及应用、JavaScript 特效制作(一)、JavaScript 特效制作(二)。

本书附赠 PPT 教学课件，可通过http://www.tupwk.com.cn/downpage下载。

本书内容安排合理，层次清楚，通俗易懂，实例丰富，突出理论和实践的结合，可作为各类高等院校、高职高专及培训机构的教材，也可供广大程序设计人员参考。

图书在版编目(CIP)数据

JavaScript 技术应用 / 武汉厚溥教育科技有限公司　编著. —北京：清华大学出版社，2014
(软件工程师培养丛书)
ISBN 978-7-302-36243-2

Ⅰ. ①J…　Ⅱ. ①武…　Ⅲ. ①JAVA 语言—程序设计　Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字(2014)第 076267 号

责任编辑：刘金喜　蔡　娟
封面设计：崔东方
版式设计：妙思品位
责任校对：成凤进
责任印制：

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn，010-62794504

印 刷 者：
装 订 者：
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张：**16.5　　**字　　数：**291 千字
版　　次：2014 年 6 月第 1 版　　**印　　次：**2014 年 6 月第 1 次印刷
印　　数：1～3000
定　　价：30.00 元

产品编号：

编 委 会

前　言

JavaScript 是一种由 Netscape(网景公司)的 LiveScript 发展而来的、原型化继承的、基于对象的、动态类型的、区分大小写的客户端脚本语言，主要目的是解决服务器端语言(如Perl)遗留的速度问题，为客户提供更流畅的浏览效果。当时服务端需要对数据进行验证，由于网络速度相当缓慢，只有 28.8Kbps，验证步骤浪费的时间太多。于是 Netscape 的浏览器 Navigator 加入了 JavaScript，提供了数据验证的基本功能。JavaScript 是一种基于对象和事件驱动并具有相对安全性的客户端脚本语言。同时也是一种广泛用于客户端 Web 开发的脚本语言，常用来给 HTML 网页添加动态功能，如响应用户的各种操作。它最初由 Netscape 的 Brendan Eich 设计，是一种动态、弱类型、基于原型的语言，内置支持类。

本书是“软件工程师培养丛书”中的一本专业教材，该丛书是由武汉厚溥教育科技有限公司开发，以培养符合企业需求的软件工程师应用开发、实施为目标的 IT 职业教育丛书。在开发该丛书之前，我们对 IT 行业的岗位序列做了充分的调研，包括研究从业人员技术方向、项目经验和职业素质等方面的需求，通过对面向的学生的特点、行业需求的现状以及实施等方面的详细分析，结合“厚溥”对软件人才培养模式的认知，按照软件专业总体定位要求，进行软件专业产品课程体系设计。该丛书集应用软件知识和多领域的实践项目于一体，着重培养学生的熟练度和规范性、集成和项目能力，从而达到预定的培养目标。

本书包含 8 章：JavaScript 简介、JavaScript 语句和函数、JavaScript 对象(一)、JavaScript 对象(二)、文档对象模型、JavaScript 事件及应用、JavaScript 特效制作(一)、

JavaScript 特效制作(二)。

我们对本书的编写体系做了精心的设计，按照“理论学习—知识总结—上机操作—课后习题”这一思路进行编排。“理论学习”部分描述了通过本案例要达到的学习目的与涉及的相关知识点，使学习目标更加明确；“知识总结”部分概括了案例所涉及的知识点，使知识点完整系统地呈现；“上机操作”部分对案例进行了详尽分析，通过完整的步骤帮助读者快速掌握该案例的操作方法；“课后习题”部分帮助读者理解章节的知识点。本书在内容编写方面，力求细致全面；在文字叙述方面，注意言简意赅、重点突出；在案例选取方面，强调案例的针对性和实用性。

本书凝聚了编者多年来的教学经验和成果，可作为各类高等院校、高职高专及培训机构的教材，也可供广大程序设计人员参考。

本书 PPT 教学课件可通过 http://www.tupwk.com.cn/downpage 下载。

本书由武汉厚溥教育科技有限公司编著，由翁高飞、王鹏等多名企业实战项目经理编写。本书编者长期从事项目开发和教学实施，并且对当前高校的教学情况非常熟悉，在编写过程中充分考虑到不同学生的特点和需求，加强了项目实战方面的教学。本书编写过程中，得到了武汉厚溥教育科技有限公司各级领导的大力支持，在此对他们表示衷心的感谢。

参与本书编写的人员还有：武汉商学院曹静老师、荆州职业技术学院方风波老师、武汉工程职业技术学院邹治伟和彭莉老师、湖北三峡职业技术学院李建利老师、武汉软件工程职业学院谢日星老师、黄冈职业技术学院吴金秀老师、湖北国土资源职业学院徐霞和明素华老师等。

限于编写时间和编者的水平，书中难免存在不足之处，希望广大读者批评指正。

服务邮箱：wkservice@vip.163.com。

编　者

2014 年 2 月

目　录

第1章

JavaScript简介

课程目标

- 了解 JavaScript 语言
- 掌握 JavaScript 数据类型
- 掌握 JavaScript 运算符

简　介

JavaScript 是一种基于对象的脚本语言，是网景公司(Netscape)最初在它的 Navigator 2.0 产品上设计并实现的，其前身叫做 LiveScript。语法上，JavaScript 和 C、C++、Java 等编程语言类似，它不仅具有条件分支结构、循环结构和函数等程序结构；而且支持 number、string 和 boolean 等原始数据类型，还包括数组对象、数学对象及正则表达式对象。自从 Sun 公司推出著名的 Java 语言之后，Netscape 公司采用了有关 Java 的程序概念，将自己原有的 LiveScript 重新进行设计，并改名为 JavaScript，这纯粹是一种商业上的市场策略，所以认为 JavaScript 是简化版的 Java 完全是一种误解。

JavaScript 是客户端脚本语言，也就是说，JavaScript 是在客户的浏览器上运行的，不需要服务器的支持。JavaScript 最初的动机是想要减轻服务器数据处理的负担，如表单数据的验证工作、在网页上显示时间、动态广告等。随着 JavaScript 所支持的功能日益增多，网页制作人员转而利用它来进行动态网页的设计。

JavaScript 是一种解释语言，其源代码在客户端执行之前不需经过编译，而是将文本格式的字符代码在客户端由浏览器解释执行。这就是说，JavaScript 是需要浏览器支持的。Microsoft 公司的 IE 浏览器在以前的版本中是不支持 JavaScript 语言的，从 IE 4.0 之后开始全面支持 JavaScript，这使得 JavaScript 成为两大浏览器的通用语言，也成为当前制作动态网页的利器。

1.1　什么是脚本语言

脚本语言的英文为 Scripting Language。脚本就是指通过记事本或其他文本编辑器创建，并保存为特定扩展名(如.reg、.vbs、.js、.inf 等)的文件。例如，*.reg 就是用于修改注册表的脚本文件，*.js 就是 JavaScript 的脚本文件。脚本文件一般都以文本形式存在，类似于一种命令。

我们知道，像用 C、C++、C#等编程语言编写的程序，必须先经过编译，将源代码

转换为二进制代码之后，以可执行文件的形式存在。脚本语言不需要编译，一般都有相应的脚本引擎来解释执行。像 Perl、JavaScript、VBScript 等脚本语言则不需要事先编译，只要使用合适的解释器便可以执行其代码。

脚本语言与编程语言也有很多相似地方，其函数与编程语言就比较相似，它也涉及变量。它与编程语言之间最大的区别是编程语言的语法和规则更为严格和复杂一些。

JavaScript 作为脚本语言的一种，在站点的开发中应用得相当广泛。编程人员可以使用 JavaScript 制作出各种页面特效。

JavaScript 的主要作用：

(1) 客户端页面表单数据验证。通过使用 JavaScript 可以对用户表单中所要提交到服务器的数据进行合法性和有效性验证。例如，验证电话号码不能包含数字和“-”以外的字符；验证身份证号码必须是 18 位，验证用户名是否为空，等等。通过数据的验证来保证提交到服务器的数据的完整性和合法性。

(2) 使用 JavaScript 动态改变网页元素或页面元素的样式。例如，在门户站点如新浪网站或搜狐网站上看到的动态广告，它可以显示或者关闭，也可以随着网页的滚动，广告也随之滚动。使用 JavaScript，能制作出很有观感性的界面效果。

(3) 响应客户端鼠标和键盘事件。

JavaScript 脚本语言的特点：

(1) JavaScript 的代码在客户端执行。

(2) JavaScript 由客户端的浏览器解释执行。

(3) JavaScript 语言是基于对象的语言。

JavaScript 和 Java 语言的区别：

(1) Java 是 Sun 公司推出的新一代面向对象的程序设计语言，特别适合于 Internet 应用程序开发。JavaScript 是 Netscape 公司的产品，其目的是扩展 Netscape Navigator 功能。

(2) 基于对象和面向对象。Java 是一种真正的面向对象的语言，即使开发简单的程序，必须设计类。JavaScript 是一种脚本语言，它是一种基于对象(Object Based)和事件驱动(Event Driver)的编程语言，它本身提供了非常丰富的内部对象供编程人员使用。

(3) 解释和编译。两种语言在其浏览器中所执行的方式不一样。Java 的源代码在传

递到客户端执行之前，必须经过编译，因而客户端上必须具有相应平台上的仿真器或解释器(称为 Java 虚拟机)。JavaScript 源代码在发往客户端执行之前不需经过编译，而是将文本格式的字符代码发送给客户端由浏览器解释执行。

(4) 强类型和弱类型。Java 采用强类型的变量检查，即所有变量在编译之前必须要声明。JavaScript 中的变量声明采用弱类型，即变量在使用前不需做声明，而是解释器在运行时检查其数据类型。

(5) 嵌入方式不一样。在 HTML 文档中，JavaScript 使用<script>...</script>来标识，而 Java 使用<applet>...</applet>来标识。

(6) 静态数据绑定和动态数据绑定。Java 采用静态数据绑定，即 Java 的对象引用必须在编译时进行，以使编译器能够实现强类型检查。JavaScript 采用动态绑定，即 JavaScript 的对象引用在运行时进行检查，如不经编译就无法实现对象引用的检查。

1.2 本书 JavaScript 开发和运行环境

开发环境：Dreamweaver 8.0。

运行环境：MS IE 6.0 sp3 或以上的版本。

1.3 在网页中使用 JavaScript

在网页中使用 JavaScript 有以下 4 种方法：

(1) 把 JavaScript 代码写在<script>…</script>标签里面，将标签插入到网页中。

(2) 在由<script>标签中使用的 src 属性指定使用外部脚本文件(.js)。

(3) 放置在事件处理程序中，该事件处理程序由 onclick 或 onmouseover 这样的 HTML 标签的属性值指定。

(4) 使用伪 URL，在浏览器的地址栏中使用特殊的 javascript:协议加上代码。例如，在浏览器的地址栏中输入 javascript: alert("hello")将显示“hello”消息框。

本章主要介绍前面两种在页面中使用 JavaScript 代码的方法。

1.3.1　使用<script>标签

使用<script>标签将 JavaScript 代码嵌入到页面中是最常见的方法。首先来看一段嵌入到网页中的 JavaScript 代码，这是第一个 JavaScript 程序，这段代码能实现弹出一个对话框。打开记事本，输入示例 1-1 所示的代码，保存为 helloworld.html。

示例 1-1：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>HelloWorld</title>
</head>
<body>
<script language="javascript">
<!--
      //使用 window 对象的 alert 方法弹出对话框
   alert("欢迎进入 JavaScript 世界！ ");
//-->
</script>
</body>
</html>
```

在 IE 浏览器中打开 helloworld.html，运行结果如图 1-1 所示。

图　1-1

如同 HTML 语言一样，JavaScript 程序代码是一些可用文本编辑器浏览和编辑的文本。JavaScript 程序代码由<script language="javascript">...</script>标签来说明。在标签之间就可加入 JavaScript 代码。W3C HTML 语法规范建议，language 属性最好不要使用，取而代之，使用属性 type 来标识 MIME 使用的语言类型。JavaScript 的 MIME 类型通常使用"text/javascript"。除了在<script>标签中使用 type 属性外，依照 HTML 的规范必须也

要指定<meta>元素。以后统一在<script>标签中使用 type 属性。

小知识

W3C 是指 World Wide Web 联盟，负责站点技术的标准化工作。例如，制定 HTML、XML 和 CSS 的规范和标准。W3C 的官方站点是 http://www.w3c.org，从那里可以得到有关 Web 方面的规范的标准信息。

例子中使用 HTML 的注释标记，通过<!--...//-->标签来说明，如果浏览器不支持 JavaScript 代码，则所有在其中的内容均被忽略。若浏览器支持，则执行其结果。使用注释是一个好的编程习惯，它使程序具有可读性。

从上面的实例分析中可以看出，编写一个 JavaScript 程序确实是非常容易的。

1.3.2 使用 JavaScript 外部文件

下面使用在网页中嵌入 JavaScript 源文件的方式来实现刚才的例子，打开 Dreamweaver 8.0，新建一个 JavaScript 源文件，文件名字是 hello.js，如图 1-2 所示。

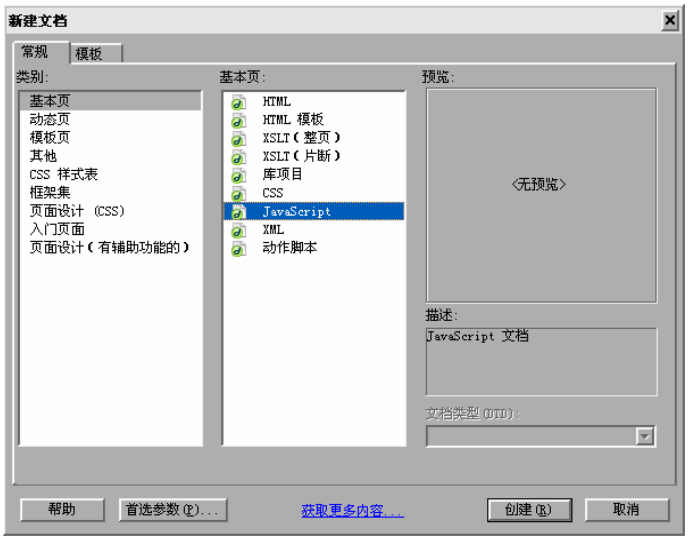

图 1-2

在创建的 hello.js 文件中输入以下代码：

```
alert("欢迎进入 JavaScript 世界！ ");
```

使用 Dreamweaver 新建一个 HTML 网页 helloworld1.hmtl，代码如示例 1-2 所示。

示例 1-2：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>HelloWorld</title>
</head>
<body>
<script    type="text/javascript" src="hello.js" ></script>
</body>
</html>
```

使用外部 js 文件的方式，同样可以实现弹出一个“欢迎进入 JavaScript 世界!”的对话框。

1.3.3　JavaScript 编写规范

当学习一门新的语言的时候，很重要的一点就是要知道它有哪些主要的特点。例如，代码是如何被执行的以及编写 JavaScript 代码通常要遵循的规范等。

(1) JavaScript 代码一行一行地被浏览器解释执行。把 JavaScript 代码的函数定义和变量的声明放在页面的<head>…</head>标签内比较好。

(2) JavaScript 代码大小写敏感。JavaScript 区分大小写，编写 JavaScript 代码时应正确处理好大小写字母。例如，result、Result 和 RESULT 是不同的标识符。这与 HTML 的标签和属性不区分大小写是不同的。例如，在 HTML 标签的属性中 onclick、ONCLICK 和 OnClick 作用是相同的。本书建议，HTML 标签和属性一律小写。

(3) 语句是 JavaScript 代码的基本单位，通常是以分号表示语句的结束。如果使用换行作为语句的结束，也是可以的。这就说明了 JavaScript 代码的不严格，本书要求，一条语句的结束必须使用分号作为结束标志。

(4) 和 C、Java 语言一样，使用{ }符号来标识由多条语句代码组成的代码块。在 JavaScript 代码中，块的开始符号“{” 和结束符号“}”必须是成对出现的。

(5) JavaScript 代码中的空格。JavaScript 会忽略多余的空白区域和空格。在 JavaScript

脚本中，出于编程的需要，可以添加额外的空格或制表符(Tab)使代码的格式工整，用来增强代码的可读性。

(6) 使用注释。用户可以使用行注释和块注释来说明脚本中语句的作用、函数的功能或其他信息。JavaScript 中的行注释使用双斜线(//)，块注释使用/*…*/。

1.4 JavaScript 核心构成

JavaScript 作为一种脚本语言，其核心构成包括数据类型、程序控制、内置对象和自定义对象。本章先介绍数据类型和运算，后面的章节将陆续介绍程序的控制和浏览器提供的内置对象及自定义的函数和对象。正是这些核心成分使得 JavaScript 功能强大，同时使得用户实现复杂的业务逻辑成为可能。

1.4.1 JavaScript 数据类型

JavaScript 的数据类型和 Java 相似，分为基本数据类型和引用数据类型。基本数据类型有下面几种：

(1) 数值数据类型(number)：JavaScript 支持整数和浮点数，占 8 个字节。整数可以为正数、0 或者负数；浮点数可以包含小数点，也可以包含一个 e(大小写均可，在科学记数法中表示“10 的幂”)，或者同时包含这两项。

(2) 布尔类型(boolean)：boolean 类型的取值可以是 true 或 false。

(3) 未定义数据类型(undefined)：undefined 表示一个未声明的变量，或已声明但没有赋值的变量，或一个并不存在的对象属性。

(4) 空数据类型(null)：null 值就是没有任何值，什么也不表示。

引用数据类型有下面几种：

(1) 字符串类型(string)：字符串是用单引号或双引号来说明的。例如：

```
"One World ! One Dream ! ";
```

(2) Array 数组类型：数组是数据元素的有序集合。

(3) 对象类型(Object)：对象是JavaScript中的重要组成部分，这部分将在后面章节详细介绍。

使用 typeof 方法可以查看数据的类型。例如，var str="你好！ ";alert(typeof(str));将显示 string。var obj = null ;alert(typeof(obj)); 将显示 object。

1.4.2　变量

变量就是所对应的值可能随程序的进行而变化的量。JavaScript 使用 var 关键字来声明一个变量。

1. JavaScript 中变量的命名规则

JavaScript 中变量的命名规则如下：

(1) 变量名的第一个字符只能是英文字母或下划线。

(2) 变量名从第二个字符开始，可以使用数字、字母和下划线。

(3) 变量名区分大小写。变量 A 和变量 a 是两个不同的变量。

(4) 不能使用 JavaScript 的关键字(保留字)。

2. 变量定义的方法

在定义 JavaScript 变量时，可以使用以下方式：

```
(1) var    name;                    //只声明变量，没有给初值
(2) var    answer = null;           //声明变量同时给变量赋空值
(3) var    price = 12.50;           //声明变量同时给变量赋数值的值
(4) var    str ="Hello!Mike";       //声明变量同时给变量赋字符串的值
(5) var    a, b, c;                 //使用逗号同时声明多个变量，没有给初值
(6) result = true;                  //省略 var 关键字来声明变量，赋布尔值
```

虽然 JavaScript 中使用数据类型来描述数据，但是由于 JavaScript 语言本身的特点是一种弱类型的语言，所以所定义的变量的数据类型决定于变量的值本身。例如，开始声明一个变量 var str ="this is test! "，变量 str 是字符串类型。在程序运行的过程中可以把一个布尔值重新赋给这个变量，str = true，这时，这个变量的数据类型就成为布尔型。这就说明了声明变量时不指定数据类型的原因。看看定义变量的示例 1-3。

示例 1-3：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>变量的定义</title>
</head>
<body>
<script type="text/javascript">
<!--
var salary; //定义变量
salary = 2000; //给变量赋值
var name = "布什";
var price = 2.5;
var isFamle = true ;
var obj = null ;
//使用 document.write()方法，可以将数据输出到页面
document.write("您的薪水是 : " +salary+ "元! salary 的类型是:"+typeof (salary)+"<br>");
document.write("我的名字是 : " + name + "! name 的类型是:"+typeof(name) +"<br>");
document.write("苹果价格是 : " + price + "元! price 的类型是:"+typeof (price)+"<br>");
document.write("isFamle 的类型是:"+typeof(isFamle)+"<br>" );
document.write("obj 的类型是:"+typeof(obj)+"<br>");
//-->
</script>
</body>
</html>
```

页面浏览结果如图 1-3 所示。

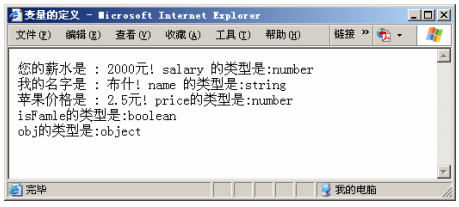

图 1-3

1.4.3 混合计算时的数据类型

各种数据类型混合在一起计算时，所计算出来的结果如下：

(1) 整数+小数结果是小数。

(2) 整数+字符串结果是字符串。

(3) 整数+布尔型结果是整数。

(4) 整数+空值结果是整数。

(5) 小数+字符串结果是字符串。

(6) 小数+布尔型结果是小数。

(7) 小数+空值结果是小数。

(8) 字符串+布尔型结果是字符串。

(9) 字符串+空值结果是字符串。

(10) 布尔型+空值结果是整数。

1.4.4　数据类型的转换

数据类型的转换在任何语言里都是一个至关重要的部分，如何将数据转换成程序中需要的数据类型，也是程序开发人员需要掌握的。JavaScript 中的数据类型转换分为自动类型转换和强制类型转换。

1. 自动类型转换

自动类型转换如表 1-1 所示。

表 1-1　自动类型转换

变量原来的值	转换成为下面的类型后的值			
var x;	string	number	boolean	object
未给初值	"undefined"	NaN	false	错误
null	"null"	0	false	object
非空字符串	字符串本身	字符串或 NaN	true	字符串对象
""	""	0	false	字符串对象
0	"0"	0	false	数值对象
不为 0 的数字	"数字本身"	数字本身	true	数值对象
NaN	"NaN"	/	false	数值对象
true	"true"	1	true	boolean 对象
false	"false"	0	false	boolean 对象

2. 强制类型转换

强制类型转换是指数字与字符串之间的转换。

(1) 转换成整数：用 parseInt()函数。

(2) 转换成小数：用 parseFloat()函数。

示例 1-4：

```
<html >
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>类型转换</title>
</head>
<body>
 <script type="text/javascript">
 <!--
 var address;
document.write("address 的值是:" +address+"<br>");
var phone = null;
document.write("phone 的值是:" +phone+"<br>");
 var sname = "HOPE"; //定义变量
 var snameResult = parseInt(sname);      //将变量转换成整型数字
 document.write("将姓名转换为数字的结果是:" + snameResult+"</br>");
 var age = "20";     //定义变量
 var ageResult = parseInt(age);  //将变量转换成整型数字
 document.write("将年龄转换为数字的结果是:" + ageResult);
 //-->
 </script>
</body>
</html>
```

页面浏览结果如图 1-4 所示。

图 1-4

1.5　JavaScript 表达式和运算符

JavaScript 中，表达式是由操作数和操作符组成的。表达式先按照某个规则计算，然后把值返回。JavaScript 中的运算符主要分为以下几种类型：赋值运算符、算术运算符、比较运算符、逻辑运算符、字符串运算符、求值运算符。

1.5.1　赋值运算符

同 C 和 Java 语言一样，JavaScript 里最基本的运算是赋值运算。使用赋值运算符“=”，把一个值赋给一个变量。例如，var name = "比尔.盖茨";var isTrue = true。也可以使用赋值运算给多个变量同时赋值，例如，var x = y = z = w = 10 ; 结果是所有的变量的值都是 10。学习中要注意“=”和“==”的使用，有人经常会把“==”写成“=”，这也是常犯的错误之一。

1.5.2　算术运算符

算术算符如表 1-2 所示。

表 1-2　算术运算符

运算符	说明	示例	结果
+	加法运算	x = 5, y = 7; sum = x+y;	sum 值 12
-	减法运算	x = 5, y = 7; sum = x-y;	sum 值-2
*	乘法运算	x = 5, y = 7; sum = x*y;	sum 值 35
/	除法运算	x = 5, y = 10; sum = x/y;sum1=y/x;	sum 值 0.5,sum1 值 2
%	取余运算	x = 5, y = 7; sum = x%y;sum1=y%x;	sum 值 5,sum1 值 2

示例 1-5：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>算术运算符</title>
</head>
```

```
<body>
<script type="text/javascript">
<!--
var num1 = 5 ;
var num2 = 4 ;
document.write("和是:"+(num1 + num2) +"<br />");
document.write("差是:"+(num1 - num2 ) +"<br />");
document.write("积是:"+(num1 * num2) +"<br />" );
document.write("商是:"+(num1 / num2 ) +"<br />");
document.write("余数是:"+(num1 %num2 ) +"<br />");
// -->
</script>
</body>
</html>
```

页面浏览结果如图 1-5 所示。

图 1-5

1.5.3 结合运算符

同 C 和 Java 语言一样，JavaScript 语言也支持结合运算。结合运算符如表 1-3 所示。

表 1-3 结合运算符

运算符	等价于	示例
x += y	x = x+y	var x = 5; x += 7; x 值是 12
x-= y	x = x-y	var x = 5; x-=7; x 的值是-2
x *= y	x = x * y	var x = 5; x *= 7; x 的值是 35
x /= y	x = x / y	var x = 5; x /= 2; x 的值是 2.5
x %= y	x = x % y	var x = 5; x %= 4; x 的值是 1

结合运算符中有两个特殊的运算符，自加运算符“++”和自减运算符“--”。使用

的过程中要注意是先加还是后加，先减还是后减。例如，var x = 3 ;var y ; y = x++和 y=++x 这两个语句执行后 y 的值是不同的，在程序中一定要注意使用。

1.5.4　比较运算符

比较运算符比较两个操作对象，并返回一个逻辑值。操作对象既可以是数字也可以是字符串值。比较运算符如表 1-4 所示。

表 1-4　比较运算符

运算符	说明	示例	结果
==	等于。如果两个操作数相等，则返回 true	2 = = 2	true
!=	不等于。如果两个操作数不等，则返回 true	2 != 5	true
>	大于。如果左操作数大于右操作数，则返回 true	3 > 2	true
>=	大于或等于。如果左操作数大于或等于右操作数，则返回 true	5>=3	true
<	小于。如果左操作数小于右操作数，则返回 true	3<2	false
<=	小于或等于。如果左操作数小于或等于右操作数，则返回 true	3<=2	false
===	绝对相等。如果操作对象相等并且类型相等，则返回 true	5 === 5	true
!==	绝对不等。如果操作对象不相等，并且不是同一类型，返回 true	5 !=== '5'	true

字符串的比较按照字母表顺序进行比较，考虑到字母有大小写，所以必须遵循下面的规则：

(1) 小写字母小于大写字母。

(2) 较短的字符串小于较长的字符串。

(3) 先出现在字母表的字符小于后面的字符。

例如，下面的例子均返回 true："b">"a" ; "thomas">"bush" ; "bbbbb">"b" ; "abC">"abc"。

1.5.5　逻辑运算符

逻辑运算符是对两个表达式进行处理，并返回一个布尔值，其真值表如表 1-5 所示。

表 1-5　真值表

表达式 1 值	表达式 2 值	&& 与运算结果	||或运算结果	!表达式 1 运算结果
true	true	true	true	false
true	false	false	true	false
false	true	false	true	true
false	false	false	false	true

问题：

请问下面的代码进行逻辑运算后的值是多少呢？

var str="test "; str && true 结果是什么呢？str || true 结果是什么呢？

var num =12 ;num && true 结果是什么呢？num || true 结果是什么呢？

参照数据类型自动转换表。

1.5.6　字符串运算符

字符串运算符：“+”对字符串进行连接处理。

示例 1-6：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>字符串运算符</title>
</head>
<body>
<script type="text/javascript">
<!--
var str1 = "北京，";
var str2 = "欢迎你！";
var str3 = str1 + str2 + "汤姆";
document.write("str3=" + str3 + "<br>");
var str4 = "请付"+ 50+"元的士费！";
document.write("str4=" + str4);
//-->
</script>
</body>
</html>
```

页面浏览结果如图 1-6 所示。

图　1-6

1.5.7　条件运算符

语法：(条件)？条件真的值: 条件假的值。

例如：status = (age >= 18) ? "adult" : "minor" ;

解释：如果 age >= 18，则将 adult 赋给 status，否则将 minor 赋给 status。

1.5.8　运算符的优先级

在表达式中，遇到多个操作符同时存在时，按优先级进行运算，如表 1-6 所示。

表 1-6　运算符的优先级

优先级	运算符
1	. [] () ++ -- ！ typeof
2	* / % + -
3	< <= > >= == != === !===
4	&& || ?: = *= /= %= += -=

【小结】

- JavaScript 语言是基于对象的语言
- 使用<script>标签在网页中使用 JavaScript
- 使用<script>标签的 src 属性引入外部文件.js 文件

- 掌握 JavaScript 中的数据类型
- 掌握 JavaScript 中各种运算符

【自测题】

1. 下列对 JavaScript 语言描述不正确的是(　　)。

   A. JavaScript 在客户端执行　　B. JavaScript 由客户端解释执行

   C. JavaScript 语言是基于对象的　　D. JavaScript 在服务器端执行

2. 在网页中引入 JavaScript 外部文件使用下面哪个标记？(　　)

   A. <body>　　B. <head>　　C. <script>　　D. <html>

3. 在 JavaScript 中，表达式 5 + “5”的计算结果是(　　)。

   A. 10　　B. 55

4. 在 JavaScript 中，document.write(6/5)的输出结果是(　　)。

   A. 1　　B. 1.2

5. 使用类型转换，var r = parseInt("A ") ; alert (r); 弹出的对话框显示的是(　　)。

   A. NaN　　B. A

上机部分

上机目标

- 掌握在网页中引入 JavaScript 的方式
- 掌握 alert 函数和 document.write 函数
- 掌握 JavaScript 数据类型及类型转换的方式
- 掌握 JavaScript 运算符

上机练习

◆第一阶段◆

练习 1：弹出对话框显示文字

问题描述：

在网页中使用 JavaScript 代码，动态地在页面加载时弹出一个对话框，显示“欢迎使用 JavaScript!”。

问题分析：

(1) 要在网页打开的时候弹出对话框，需要使用 JavaScript 的 alert()函数。

(2) 可以使用在网页中嵌入<script>标签的方式实现。

参考步骤：

(1) 创建新的 HTML 页面，取名 lesson1.html。

(2) 更改网页中<title>的值为“欢迎界面”，删除<!DOCTYPE>标签和<html>的 xmlns 属性。

(3) 修改 JavaScript 代码，如图 1-7 所示。

图　1-7

(4) 按下快捷键 F12，在 IE 浏览器中查看 lesson1.html 页面，结果如图 1-8 所示。

图 1-8

(5) 图 1-8 是将<script>标签放在<head>标签里面，修改上面的代码，把整个<script>标签拖动到<body>标签里去，同时在<body>标签里面加入一句静态文本“欢迎使用 JavaScript!”，代码如图 1-9 所示。按下 F12 键运行，查看与刚才的顺序编写的代码实现的结果有什么不同。

图 1-9

练习 2：在页面中使用外部 js 文件

问题描述：

在页面中引入 JavaScript 外部文件，并要使用 document.write()方法在网页中打印“欢迎使用 JavaScript!”，将文字放在层标签中，文字的大小是 36，文字的颜色是红色。

问题分析：

(1) 要引入外部文件，必须首先创建*.js 文件。

(2) 要使用层，必须要用 标签，并将文字放在其中。

(3) 使用 的 style 属性来控制文字大小和颜色。

(4) 注意双引号和单引号的使用。

参考步骤：

(1) 创建 lesson2.js 文件。

(2) 在 lesson2.js 文件中输入下面的代码。

```
document.write("<div style='font-size:36px; color:red'>");
document.write("欢迎使用 JavaScript ！ ");
document.write("</div>");
```

(3) 在 lesson2.js 文件的相同路径下创建一个网页文件 lesson2.html。

(4) 在 lesson2.html 编写如下代码。

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=gb2312" />
<title>使用外部 *.js 文件</title>
</head>
<body>
<script type="text/javascript" src="lesson2.js"></script>
</body>
</html>
```

(5) 按下 F12 键，在 IE 中浏览 lesson2.html。运行结果如图 1-10 所示。

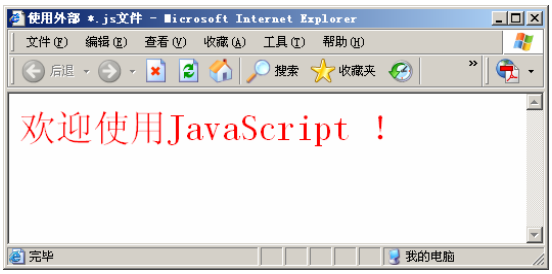

图　1-10

◆第二阶段◆

练习 3：在页面中使用外部 js 文件

问题描述：

编写一个页面，以及一个 js 文件，使用引入外部文件的方式在网页中加入链接，并

给链接一个样式。链接本身是红颜色，当鼠标移动到链接上时变为黄色，访问后的链接为绿色，并将这个 JavaScript 代码引入到页面中。

问题分析：

(1) 实现一个网页。

(2) 实现一个 JavaScript 源文件。

(3) 使用 document.write()函数。

(4) 注意代码中的单引号和双引号的使用。

【课后作业】

(1) 分别在网页的<head>和<body>中嵌入 JavaScript 代码，弹出对话框显示“你好”。

(2) 分别在网页的<head>和<body>中使用外部.js 文件，弹出对话框显示“你好”。

(3) 掌握自动类型转换的知识点，看看下面的几个 alert()方法的返回值各是多少，体会数据类型的自动转换。

① alert(2= =true);

② alert(2== =true) ;

③ alert("2"&&true);

④ alert(0 == "");

⑤ alert(null= =false);

(4) 在网页中打印 5/4 和 20/4 的结果。

(5) 在网页中查看 alert(10/0)的结果。(数据类型 Infinity)

(6) 在网页里查看结果: var w = 3 , q = 2 ,t = 5 ,e = 8 ; w = q = t = e ;请问最后的结果 w,q,t,e 的值各是多少？(体会赋值运算)