

人工智能技术专业群系列教材

主 编◎王向华

副主编◎杜海颖 谢莉莉

Web 前端开发

——JavaScript+jQuery 活页式教程

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是 Web 前端开发的入门教材，主要介绍了 JavaScript 和 jQuery 的基础知识和基本技能。采用活页式教材的编写理念，将各部分知识模块化，整体内容循序渐进，精简了理论知识，更加注重案例与实践练习，可逐步加强学习对于 Web 前端开发技术的理解和掌握。

本书共包含 12 个学习活动。学习活动 1~9 围绕 JavaScript 进行讲解，主要包括输入输出数据、控制分支结构程序、控制程序循环执行、存储并编辑数组数据、JavaScript 代码模块化、JavaScript 对象、掌握 JavaScript 中 BOM 操作、使用 DOM 控制页面元素、JavaScript 事件处理。学习活动 10~11 围绕 jQuery 进行讲解，主要包括使用 jQuery 框架处理元素、基于 jQuery 的事件处理与动画。学习活动 12 围绕正则表达式进行讲解。

本书内容通俗易懂、突出实践，可作为本科、高职院校计算机相关专业的 Web 前端开发课程的教材，也可供广大 IT 技术初学者和编程爱好者自学使用。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

Web 前端开发: JavaScript+jQuery 活页式教程 / 王向华主编. —北京: 电子工业出版社, 2022.12

ISBN 978-7-121-44817-1

I. ①W… II. ①王… III. ①网页制作工具—教材②JAVA 语言—程序设计—教材 IV. ①TP393.092.2
②TP312.8

中国版本图书馆 CIP 数据核字 (2022) 第 253669 号

责任编辑: 关雅莉 文字编辑: 张志鹏

印 刷:

装 订:

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×1 092 1/16 印张: 20.25 字数: 518.4 千字

版 次: 2022 年 12 月第 1 版

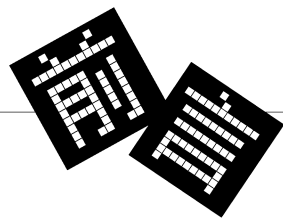
印 次: 2022 年 12 月第 1 次印刷

定 价: 68.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: (010) 88254576, zhangzhp@phei.com.cn。



党的二十大报告指出：“教育、科技、人才是全面建设社会主义现代化国家的基础性、战略性支撑”。职业教育作为教育的一个重要组成部分，其目的是培养应用型人才和具有一定文化水平和专业知识技能的社会主义劳动者和建设者，职业教育侧重于实践技能和实际工作能力的培养。在职业院校中，“Web 前端开发”是人工智能专业的一门重要专业课。

HTML、CSS 和 JavaScript 是 Web 前端开发常用的 3 种技术，通过这 3 种技术衍生出了各种框架和解决方案。随着移动互联网的发展，大量高性能移动终端设备的应用对 Web 前端开发提出了更高的要求，JavaScript 作为 Web 前端开发中必不可少的一种技术，越来越展现出其重要性，也逐渐成为 Web 前端开发人员的必备技能。

本书适用于具有 HTML 和 CSS 基础的读者，主要介绍了 JavaScript 与 HTML、CSS 相结合，开发交互式页面的相关知识，同时对 jQuery 框架和正则表达式的基础知识进行了介绍。本书的编者多年从事 Web 前端开发的教学工作，在内容的编排和案例的设计上更加注重学生的学习习惯和岗位能力的要求。本书对理论知识进行了合理规划，突出培养学生的实践动手能力，引导学生自主学习，增强学生解决问题的能力。

本书结构

本书共包含 12 个学习活动。学习活动 1~9 围绕 JavaScript 进行讲解，主要包括输入输出数据、控制分支结构程序、控制程序循环执行、存储并编辑数组数据、JavaScript 代码模块化、JavaScript 对象、掌握 JavaScript 中 BOM 操作、使用 DOM 控制页面元素、JavaScript 事件处理。学习活动 10~11 围绕 jQuery 进行讲解，主要包括使用 jQuery 框架处理元素、基于 jQuery 的事件处理与动画。学习活动 12 围绕正则表达式进行讲解。

本书中每个学习活动都由不同的学习任务组成，在介绍学习任务之前，设计了“学习目标”“建议学时”“学习活动简介”3 个环节，引导学生了解本次学习目标的主要内容，并为授课教师提供建议学时参考。学习任务包含“任务目标”“知识介绍”“实践练习”“知识盘点”4 个环节，有些学习任务还包含“知识拓展”环节。“知识介绍”环节主要介绍核心的理论知识；“知识拓展”环节进一步介绍拓展知识，扩展知识面，增加知识储备；“实践练习”环节提供了丰富的练习题；“知识盘点”环节提供了知识梳理与汇总表，可供学生自评、同学互评和教师评价使用。

本书内容通俗易懂、突出实践，可作为本科、高职院校计算机相关专业的 Web 前端开

发课程的教材，也可供广大 IT 技术初学者和编程爱好者自学使用。

本书特色

本书采用活页式的编写理念，将各部分知识模块化，依据“Web 前端开发”职业技能等级标准中的有关 JavaScript 和 jQuery 相关要求编写，整体内容循序渐进，在阐述理论的同时，每个学习任务都设计了实践练习题目。编者具有多年 Web 前端开发教学经验，在编写本书的过程中，一方面注重知识的融会贯通，另一方面注重学生的学习习惯，能够提升学生的学习兴趣，使学生能够及时消化所学的知识和技能。

代码说明

本书案例和实践练习的重点是 Web 前端交互式编程，为了叙述简洁，书中省略了大部分 CSS 样式和部分 HTML 代码，完整的代码见本书提供的源代码。

众所周知，注释在 Web 前端开发中是必不可少的。由于篇幅限制，书中只对必要的地方进行了注释，本书源代码中提供了比较完整的注释。

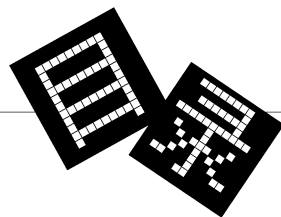
学习资源

作为一本新形态教材，本书还精心设计了 PPT、视频讲解、例题和练习的源文件，练习答案等，可扫描本书中的二维码或登录华信教育资源网注册后免费进行下载。

本书编者

本书的编者都是长期从事 Web 前端开发的一线教师，在网站、软件研发等领域积累了丰富的经验。本书由王向华老师主编，负责编写本书学习活动 1~4、学习活动 9~12，并负责全书内容的规划和统稿。谢莉莉老师参与了本书学习活动 5、学习活动 6 的编写工作，杜海颖老师参与了本书学习活动 7、学习活动 8 的编写工作。

尽管我们为编写本书付出了最大的努力，但是书中难免存在不足之处，欢迎专家和读者提出宝贵的意见。如果发现任何问题或有不认同之处，可以通过发送电子邮件与我们取得联系。电子邮箱：10151916@qq.com。



学习活动 1 输入输出数据	001
学习任务 1 引入 JavaScript 脚本	003
学习任务 2 声明 JavaScript 变量	009
学习任务 3 输入输出简单数据	013
学习任务 4 输入输出表达式	019
学习活动 2 控制分支语句程序	025
学习任务 1 使用单分支语句模拟话费充值	027
学习任务 2 使用双分支语句判断成绩	031
学习任务 3 使用多分支语句判断年龄区间	037
学习任务 4 使用 switch 语句控制图片显示	043
学习活动 3 控制程序循环执行	049
学习任务 1 使用 while 循环语句输出表格	051
学习任务 2 使用 do-while 循环语句计算水仙花数	057
学习任务 3 使用 for 循环语句输出图片	061
学习任务 4 控制循环流程	065
学习活动 4 存储并编辑数组数据	069
学习任务 1 创建与遍历一维数组	071
学习任务 2 创建、引用和遍历关联数组	077
学习任务 3 创建与遍历二维数组	081
学习任务 4 使用数组相关的方法	085
学习任务 5 数组排序	089

学习活动 5 JavaScript 代码模块化	093
学习任务 1 声明和调用函数	095
学习任务 2 定义变量作用域	101
学习任务 3 使用匿名函数	107
学习任务 4 使用回调函数	111
学习任务 5 使用闭包函数	117
学习活动 6 JavaScript 对象	121
学习任务 1 String 对象属性与方法	123
学习任务 2 Math 对象属性与方法	129
学习任务 3 Date 对象属性与方法	135
学习任务 4 自定义对象	141
学习活动 7 掌握 JavaScript 中 BOM 操作	149
学习任务 1 使用 window 对象控制浏览器	151
学习任务 2 使用 location 对象控制操作 URL	157
学习任务 3 使用 history 对象控制浏览历史	161
学习任务 4 使用 screen 对象获取屏幕信息	165
学习任务 5 使用 document 对象操作文档	169
学习活动 8 使用 DOM 控制页面元素	173
学习任务 1 获取页面 DOM 节点	175
学习任务 2 添加与删除页面节点	179
学习任务 3 通过 DOM 编辑节点内容	185
学习任务 4 使用 DOM 属性控制节点样式	191
学习活动 9 JavaScript 事件处理	199
学习任务 1 绑定事件	201
学习任务 2 常见页面事件	207
学习任务 3 常见鼠标事件	213
学习任务 4 常见键盘事件	221
学习任务 5 事件冒泡与事件委托	227
学习活动 10 使用 jQuery 框架处理元素	233
学习任务 1 引入和使用 jQuery	235
学习任务 2 使用 jQuery 选择器	239

学习任务 3 使用 jQuery 添加或删除节点	251
学习任务 4 使用 jQuery 设置元素属性	257
学习任务 5 使用 jQuery 设置元素样式	263
学习任务 6 使用 jQuery 设置元素内容	269
学习活动 11 基于 jQuery 的事件处理与动画效果	273
学习任务 1 绑定与解绑事件	275
学习任务 2 处理 jQuery 复合事件与事件委托	281
学习任务 3 使用 jQuery 实现预定义动画效果	285
学习任务 4 实现自定义动画效果	291
学习活动 12 正则表达式	297
学习任务 1 创建与使用正则表达式	299
学习任务 2 使用元字符控制正则表达式	303
学习任务 3 其他方法匹配正则表达式	309

电子工业出版社版权所有
盗版必究



扫一扫，观看 PPT

学习活动 1

输入输出数据

【学习目标】

知识目标

1. 学习引入 JavaScript 脚本的方法。
2. 学习 JavaScript 语言中数据、表达式的输入输出方法。
3. 学习简单的 JavaScript 语言程序设计方法。

技能目标

1. 能够在 HTML 脚本中正确引入 JavaScript 脚本。
2. 能够编写 JavaScript 脚本并进行简单的程序设计。
3. 能够利用工具调试 JavaScript 脚本。

【建议学时】

8 学时。

【学习活动简介】

在 Web 前端开发中，使用 HTML 和 CSS 可以做出精美的页面，但是用户和页面之间的交互需求却无法被满足。为了减轻服务器端的负担，同时增强页面的交互性，使用 JavaScript 是非常好的解决方法。同时，JavaScript 也是 jQuery 框架的基础。

本次学习活动的目标是能够正确地引入 JavaScript（JS）脚本，并能进行简单的 JavaScript 语言程序设计，可以完成简单的输入输出操作，同时能够调试、纠正程序中出现的错误。学习活动 1 的知识导图如图 1-1 所示。

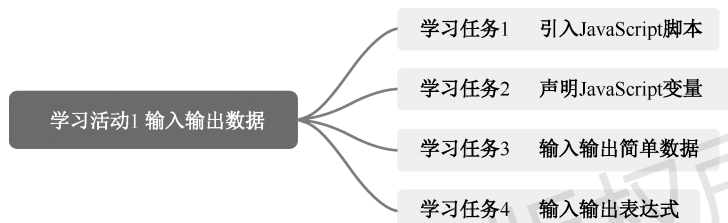


图 1-1 学习活动 1 的知识导图



扫一扫，阅读例题

学习任务 1

引入 JavaScript 脚本

任务目标

- ◆ 掌握采用行内式引入 JavaScript 脚本的方法
- ◆ 掌握采用内嵌式引入 JavaScript 脚本的方法
- ◆ 掌握采用外链式引入 JavaScript 脚本的方法



扫一扫，观看视频

知识介绍

JavaScript 语言是一种直译式脚本语言，不能单独运行，必须引入 HTML 脚本中才能被解释、执行。将 JavaScript (JS) 脚本引入 HTML 脚本中的方法有行内式、内嵌式和外链式 3 种。

1. 行内式

将简短的 JavaScript 脚本写在 HTML 标签的事件属性中，这种方法称为行内式。以下 HTML 脚本能够实现单击【确定】按钮后弹出提示框。

```
<button onclick="alert('行内式JS')">行内式</button>
```

HTML 标签的事件属性一般都以“on”开头，如 onclick、onchange 等。运行此语句所在的 HTML 脚本，单击页面中的【确定】按钮，弹出提示框。行内式 JavaScript 脚本的显示效果如图 1-2 所示。



图 1-2 行内式 JavaScript 脚本的显示效果

2. 内嵌式

当 JavaScript 脚本有多行语句时，可以采用内嵌式引入 JavaScript 脚本。这种方法是将 JavaScript 脚本包含在<script>标签中。<script>标签可以写在<head>标签或<body>标签中。示例 1-1 演示了如何在<body>标签中嵌入<script>标签。

【示例 1-1】demo1_1.html

```
1 <body>
2     <script type="text/JavaScript">
3         alert("内嵌式JS演示");
4     </script>
5 </body>
```

以上代码中，第 2 行和第 4 行分别是<script>标签的开始和结束，JavaScript 脚本写在中间。<script>标签的 type 属性值“text/JavaScript”表示代码为 JavaScript 脚本，这是 HTML5 中<script>标签的默认属性，可以省略。

运行以上代码，会弹出提示框。内嵌式 JavaScript 脚本的显示效果如图 1-3 所示。

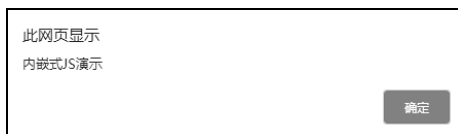


图 1-3 内嵌式 JavaScript 脚本的显示效果

3. 外链式

外链式也称为外部 JavaScript，是指将 JavaScript 脚本写在单独的文件中，文件一般以.js 作为扩展名。当 JavaScript 脚本的代码量很大，或者多个页面需要实现相同的交互效果时，一般需要采用外链式。采用外链式引入 JavaScript 脚本需要注意以下 3 点。

- (1) .js 文件中不需要写<script>标签，直接写 JavaScript 脚本。
- (2) 在 HTML 脚本中使用以下标签引入外部的.js 文件。

```
<script src=".js文件名"></script>
```

- (3) <script>标签不能写成单标签的形式，必须以“</script>”结束。

示例 1-2 演示了采用外链式引入 JavaScript 脚本的方法。

【示例 1-2】demo1_2.html

- (1) 创建 test.js，输入以下代码。

```
alert("外链式JS");
```

- (2) 创建 demo1_2.html，引入 test.js。

```
1 <body>
2     <script src="./test.js"></script>
3 </body>
```

- (3) 运行以上代码，会弹出提示框。外链式 JavaScript 脚本的显示效果如图 1-4 所示。

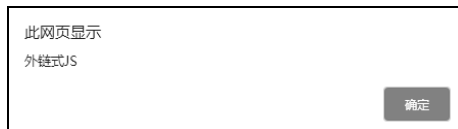


图 1-4 外链式 JavaScript 脚本的显示效果



1. JavaScript 介绍

1994 年,网景通信公司(Netscape Communications Corporation)发布了历史上第一款比较成熟的网络浏览器——Navigator,轰动一时。但是,这个时期的浏览器都是静态的,只能用来浏览,不具备与用户互动的能力,也不具备今天的浏览器的各种动态效果,如图片轮播、鼠标悬浮切换等。

1995 年,网景通信公司的系统程序员布兰登·艾奇(Brendan Eich),花了不到两周的时间设计了一种能在网页上实现动态效果的编程语言。当时,太阳微系统公司的 Java 非常有名气,网景通信公司希望借用 Java 的名气进行推广,便将这种编程语言命名为 JavaScript。实际上,JavaScript 除了语法上有点像 Java,其他都跟 Java 没有任何关系。

1996 年,包括微软在内的不同厂商都模仿网景通信公司,陆续推出了不同的 JavaScript 语言,导致 JavaScript 的语法和特性日益混乱,其标准化问题被提上日程。最终由欧洲计算机制造商协会(European Computer Manufacturers Association)以 JavaScript 1.1 为蓝本,制定了 ECMA-262 标准,并由此标准定义了一种新的脚本语言——ECMAScript。随后,国际标准化组织(ISO)也以 ECMAScript 作为标准,各个浏览器厂商开始将 ECMAScript 作为各自使用 JavaScript 语言的基础,JavaScript 也由此逐步发展完善起来。

JavaScript 是一种基于对象(Object)和事件驱动(Event Driven)的脚本语言。通过 JavaScript 能够实现页面的交互功能。

JavaScript 语言以 ECMAScript 为核心,还包括了文档对象模型(DOM)和浏览器对象模型(BOM)两部分。JavaScript 的组成如图 1-5 所示。

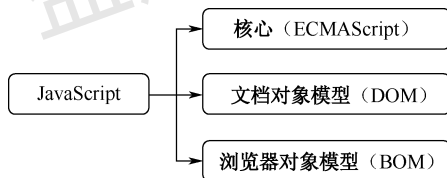


图 1-5 JavaScript 的组成

(1) 核心 (ECMAScript)。

ECMAScript 是 JavaScript 的核心标准,规定了 JavaScript 的组成,具体内容包括语法、类型、语言、关键字、保留字、操作符、对象等。

(2) 文档对象模型 (DOM)。

文档对象模型(Document Object Model, DOM)将页面映射为一个多层次节点结构。通过使用 DOM,开发者可以自由地对网页元素进行删除、添加、替换或修改等操作,实现网页的动态效果。

(3) 浏览器对象模型 (BOM)。

浏览器对象模型(Browser Object Model, BOM)用于处理浏览器的窗口和框架。通过使用 BOM,开发者可以对浏览器窗口进行操作。例如,弹出提示框、前进、后退、页面刷新、关闭浏览器窗口等。

2. JavaScript 的特点

(1) 脚本语言。

JavaScript 是由文本命令组成的脚本语言 (Script)，与非脚本语言 (如 C 语言) 相比，不需要先编译成可执行文件后才能运行，其程序在运行过程中就能逐行被解释执行。

(2) 基于对象。

JavaScript 也是一种基于对象的脚本语言，它不仅能创建对象，也能使用现有的对象。例如，使用 jQuery 框架提高了开发效率，使得 Web 前端开发更加的快捷高效。

(3) 简单。

JavaScript 语言采用弱类型的变量类型，对使用的数据类型未做严格的要求，其设计简单紧凑，灵活易用。

(4) 跨平台性。

JavaScript 脚本不依赖操作系统，仅依赖浏览器的支持。因此，一个 JavaScript 脚本在编写后可以在任意支持 JavaScript 脚本的浏览器中运行。目前，JavaScript 脚本已被大多数的浏览器所支持。

3. JavaScript 开发工具的选择

基于 JavaScript 的 Web 前端开发对软件环境的要求相对较低，只需使用编辑器和浏览器即可。常用的编辑器有 EditPlus、Notepad++、Sublime_text、HBuilder、WebStorm 等。用于 Web 前端开发与测试的浏览器一般为 Chrome 浏览器和火狐浏览器，本书默认使用 HBuilder 编辑器和 Chrome 浏览器。



实践练习



扫一扫，观看视频

1. 问答题

(1) JavaScript 语言由哪几部分组成？其核心是什么？

(2) 可以使用哪些方式将 JavaScript 脚本引入 HTML 脚本中？

(3) 在 HTML 脚本中引入当前目录中的 function.js 的代码是什么？

(4) <script>标签可以放在 HTML 脚本中的什么位置？

(5) 查阅资料，简述 JavaScript 语言的优点。

学习活动 1

输入输出数据

2. 编程练习

分别使用 JavaScript 脚本的 3 种引入方法编写程序，使得页面加载后提示框中显示“Hello World!”。请将主要代码写在下面。



知识盘点

请对本次学习任务的知识点进行梳理与汇总，填写表 1-1。

表 1-1 知识点梳理与汇总

学习任务名称：引入 JavaScript 脚本			
知识点描述：引入 JavaScript 脚本是 Web 前端开发的前提，使用正确的方式引入 JavaScript 脚本将会给后续的编程工作带来很多的便利。			
知识点	内容盘点	权重	
1. JavaScript 语言的组成及各组成部分的功能	（简述 JavaScript 语言的组成及各组成部分的功能）	20%	
2. 行内式 JavaScript 脚本	（举例说明行内式 JavaScript 脚本）	25%	
3. 内嵌式 JavaScript 脚本	（举例说明内嵌式 JavaScript 脚本）	25%	
4. 外链式 JavaScript 脚本	（举例说明外链式 JavaScript 脚本）	30%	
学习评价			
评价内容	自评	互评	师评
1. 知识点掌握情况（40%）			
2. 实践练习完成情况（60%）			
本人签名：	组长签名：	教师签名：	

学习任务 2

声明 JavaScript 变量

任务目标

- ◆ 掌握 JavaScript 语言中变量的声明与赋值方法
- ◆ 掌握 JavaScript 语言中数据类型的组成
- ◆ 了解 JavaScript 语言中常见的关键字



扫一扫，观看视频

知识介绍

1. 变量的声明与赋值

变量指的是在程序运行期间，值可以改变的量。为变量起的名字称为变量名。变量实际上是程序运行过程中，在内存中分配的一段存储空间。

JavaScript 语言中使用关键字“var”声明变量，语法如下。

```
var 变量名;
```

示例代码如下。

```
var name, age;
```

变量声明之后是没有值的，为其赋值后才有实际的意义，示例代码如下。

```
name = "Tom";
```

```
age = 20;
```

变量赋值之后，可以通过变量名进行访问，示例代码如下。

```
alert(name);
```

```
alert(age);
```

也可以在声明变量的同时进行变量赋值，这个过程称为变量初始化，示例代码如下。

```
var name = "Tom";
```

进行变量命名时，必须遵循 JavaScript 语言的命名规范，具体要求如下。

- (1) 变量名由数字、字母、下画线和符号“\$”组成。
- (2) 变量名不能以数字开头。例如，变量名“1abc”是错误的。
- (3) 变量命名时应尽量使用有意义的标识符，如 age、name、gender 等。
- (4) 命名变量建议采用规范的驼峰命名法，即当变量名由多个单词组成时，除了第一个单词，其他单词的首字母须大写，如 firstTime、endTime 等。

JavaScript 属于弱类型语言，变量可以不通过声明关键字“var”直接使用，但是因为变量作用域的问题，出现问题时排错会比较困难，因此不推荐使用这种方法。

2. JavaScript 常见关键字

JavaScript 变量名不能与保留字、关键字重名。JavaScript 常见关键字如表 1-2 所示。

表 1-2 JavaScript 常见关键字

abstract	arguments	boolean	break	byte	case	catch	char
class*	const	continue	debugger	default	delete	do	double
else	enum*	eval	export*	extends*	false	final	finally
float	for	function	goto	if	implements	import*	in
instanceof	int	interface	let	long	native	new	null
package	private	protected	public	return	short	static	super*
switch	synchronized	this	throw	throws	transient	true	try
typeof	var	void	volatile	while	with	yield	

注：表 1-2 中的每个关键字都有特殊的含义和作用。其中，“*”标记的关键字是 ECMAScript5 中新添加的。

3. JavaScript 数据类型

JavaScript 是弱类型语言，系统根据变量的值自动判断数据类型，并分配相应的存储空间。JavaScript 数据类型如图 1-6 所示。

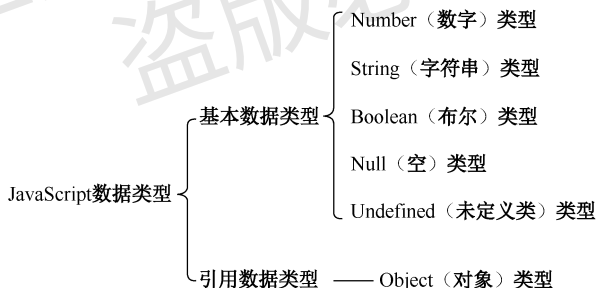


图 1-6 JavaScript 数据类型

关于 JavaScript 数据类型，需要注意以下 6 点。

- (1) Number 类型既可以表示整数，也可以表示浮点数。
- (2) Number 类型中有一个特殊值 NaN（Not a Number），表示非数字。可以使用 isNaN() 函数来判断数据或变量是否为非数字，示例代码如下。

```
document.write(isNaN('123')); //显示结果: false
document.write(isNaN(123)); //显示结果: false
document.write(isNaN("abc")); //显示结果: true
```

- (3) String 类型是一组用单引号或者双引号括起来的文本。
- (4) Boolean 类型只有两个值：true 和 false。
- (5) Null 类型只有一个值 null，表示空对象的指针。
- (6) Undefined 类型只有一个值 undefined，表示声明的变量没有被赋值。

🔗 实践练习



扫一扫，观看视频

1. 问答题

(1) JavaScript 语言中如何表示八进制数和十六进制数？

(2) 对于以下变量定义语句，请问 `isNaN(num)` 和 `isNaN('num')` 的值分别是多少？

```
var num = 876.23;
```

2. 编程练习

(1) 分别定义 `Null` 类型的变量和 `Undefined` 类型的变量，通过函数 `alert()` 输出。请写出程序代码。

(2) 定义变量 “greeting”，赋值为 “hello world”，通过函数 `alert()` 输出。请将主要代码写在下面。

(3) 定义变量 “introduce”，初始化为个人信息介绍，通过函数 `alert()` 输出。请将主要代码写在下面。

(4) 定义两个变量 “x” “y” 并进行赋值，然后交换它们的值，通过函数 `alert()` 输出交换结果。请将程序的主要代码写在下面。



知识盘点

请对本次学习任务的知识点进行梳理与汇总，填写表 1-3。

表 1-3 知识点梳理与汇总

学习任务名称：声明 JavaScript 变量			
知识点描述：因为是弱类型语言，JavaScript 变量使用灵活，能够随时改变变量的数据类型。使用过程中要注意不能与关键字重名，并且要按照软件编程规范对变量进行命名。			
知识点	内容盘点		权重
1. 声明 JavaScript 变量的方法	(简述如何声明 JavaScript 变量)		20%
2. 变量命名规范	(简述声明变量时，变量命名的规范)		20%
3. JavaScript 数据类型	(简述 JavaScript 数据类型有哪些)		20%
4. Number 类型的特殊值及判断数据为非数字的方法	(Number 类型的特殊值是什么？如何判断一个数据是否为非数字？)		20%
5. Null 类型和 Undefined 类型的数据	(简述什么是 Null 类型和 Undefined 类型的数据)		20%
学习评价			
评价内容	自评	互评	师评
1. 知识点掌握情况 (40%)			
2. 实践练习完成情况 (60%)			
本人签名：_____ 组长签名：_____ 教师签名：_____			

学习任务 3

输入输出简单数据



任务目标

- ◆ 掌握使用函数 alert() 输出信息的方法
- ◆ 掌握使用函数 confirm() 输出信息的方法
- ◆ 掌握使用函数 prompt() 输入输出信息的方法
- ◆ 掌握使用函数 document.write() 在页面中输出信息的方法
- ◆ 掌握 JavaScript 中数据类型转换的方法



扫一扫，观看视频



知识介绍

1. JavaScript 简单输入输出语句

在页面交互过程中，常见的信息输入输出方式是通过表单控件进行的。对于一些简单的信息输出既可以通过消息框（警告框、确认框或提示框）完成，也可以通过在页面中输出的方法实现。JavaScript 简单输入输出语句如表 1-4 所示。

表 1-4 JavaScript 简单输入输出语句

语句	说明
alert("text")	警告框，用于提示用户的某些信息
confirm("text")	确认框，用于验证是否接受用户的操作
prompt("text","defaultvalue")	提示框，用于提示用户输入某个值
document.write("text")	用于在页面中输出信息，常用于输出 HTML 脚本

【示例 1-3】demo1_3.html

```
1  <script>
2      alert("警告框，用于提示用户信息");
3      confirm("确认框，用于用户确认信息");
4      prompt("提示框，用于提示用户输入信息");
5      document.write("<h1>在页面中输出信息</h1>");
6  </script>
```

示例 1-3 代码运行后，函数 alert() 生成的警告框如图 1-7 所示，函数 confirm() 生成的确认框如图 1-8 所示，函数 prompt() 生成的提示框如图 1-9 所示；函数 document.write()

生成的输出信息如图 1-10 所示。

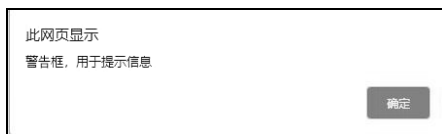


图 1-7 函数 alert()生成的警告框

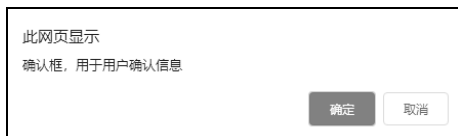


图 1-8 函数 confirm()生成的确认框



图 1-9 函数 prompt()生成的提示框



图 1-10 函数 document.write()生成的输出信息

在函数 `confirm()` 生成的确认框中, 用户可以单击【确定】或者【取消】按钮。单击【确定】按钮返回 `true`, 单击【取消】按钮返回 `false`。在函数 `prompt()` 生成的提示框中, 用户需要输入一个值, 再单击【确定】或【取消】按钮才能继续操作, 单击【确定】按钮返回输入的值, 单击【取消】按钮返回 `null`。

2. JavaScript 数据类型检测

在 Web 前端开发过程中, 有时需要根据不同的数据类型进行不同的操作。JavaScript 提供了 4 种常用的数据类型检测方法, 分别是 `typeof`、`instanceof`、`constructor` 和 `Object.prototype.toString.call()`。

(1) `typeof` 用于检测变量或者值的数据类型, 基本语法如下。

```
typeof (表达式)      // 检测表达式结果的数据类型
typeof (变量名)      // 检测变量的数据类型
```

`typeof` 运算符的返回值为字符串, 表明数据类型。`typeof` 运算结果分类如表 1-5 所示。

表 1-5 typeof 运算结果分类

返回值的数据类型	含义
undefined	表示 Undefined 类型的变量或值
boolean	表示 Boolean 类型的变量或值
string	表示 String 类型的变量或值
number	表示 Number 类型的变量或值
object	表示 Object 类型的变量或值，或者是 null
function	表示 Function 类型的变量或值

示例代码如下。

```
console.log(typeof "");           //String类型
console.log(typeof 1);            //Number类型
console.log(typeof true);        //Boolean类型
console.log(typeof null);        //Object类型
console.log(typeof undefined);   //Undefined类型
console.log(typeof []);          //Object类型
console.log(typeof function(){}); //Function类型
console.log(typeof {});          //Object类型
```

需要说明的是，`typeof` 只能检测基本的数据类型，如果遇到引用数据类型（如 `Array` 类型或者 `Object` 类型），返回值都是 `object`。

(2) `instanceof` 用于检测一个实例是否属于某个类。使用 `instanceof` 可以弥补 `typeof` 不能检测对象数据类型的局限性，示例代码如下。

```
console.log([] instanceof Array);           //true
console.log(function(){} instanceof Function); //true
console.log({} instanceof Object);          //true
console.log(new String('abc') instanceof String) //true
```

(3) `constructor` 是函数原型上的属性，该属性指向的是构造函数本身。此处仅做简单的了解，示例代码如下。

```
console.log(("1").constructor === String); //true
console.log([] .constructor === Array);    //true
console.log({} .constructor === Object);   //true
```

(4) `Object.prototype.toString.call()` 是最全面地检测数据类型的方式。返回值的类型为 `String` 类型，示例代码如下。

```
Object.prototype.toString.call(1);           //'[object Number]'
Object.prototype.toString.call({});          //'[object Object]'
Object.prototype.toString.call([]);          //'[object Array]'
Object.prototype.toString.call(null);        //'[object Null]'
```

3. JavaScript 数据类型转换

运算过程中如果出现操作数的数据类型与预期不符，系统就会自动进行数据类型转换，这种转换对用户是不可见的。如果无法进行自动类型转换，就需要进行强制类型转换。

(1) 强制类型转换。

强制类型转换主要包括强制转换为 String 类型、强制转换为 Number 类型、强制转换为 Boolean 类型。强制类型转换如表 1-6 所示。

表 1-6 强制类型转换

转换分类	转换语法
强制转换为 String 类型	方式 1: 使用 “+” 运算符与字符串拼接 方式 2: 使用 toString() 方法, 注意 null 和 undefined 这两个值没有 toString() 方法 方式 3: 使用 String() 方法
强制转换为 Number 类型	方式 1: 使用 Number() 方法 方式 2: 使用 parseInt() 或 parseFloat() 方法
强制转换为 Boolean 类型	使用 Boolean() 方法

(2) 自动类型转换。

自动类型转换是基于强制的数据类型转换之上的优先转换方式。遇到以下 3 种情况时, 系统会进行自动类型转换。

① 自动转换为 Boolean 类型。系统遇到预期为 Boolean 类型的地方 (如 if 语句的条件部分) 就会将非 Boolean 类型的数据自动转换为布尔值, 系统会自动调用函数 Boolean()。

② 自动转换为 Number 类型。算数运算符 (+、-、*、/ 等) 与非 Number 类型的值进行运算时, 除了 “+” 运算符用于字符串拼接, 系统会将这些值自动转换为 Number 类型, 然后再进行运算。

③ 自动转换为 String 类型。主要发生在字符串的 “加法” 运算时。当一个值为字符串, 另一个值为非字符串时, 后者自动转换为 String 类型。

实践练习

1. 问答题

(1) 其他数据类型转换为 Boolean 类型时, 哪些数据会转换为 false? 哪些数据会转换为 true?



扫一扫, 观看视频

(2) 将以下表达式的结果写在横线处。

typeof age;	_____	typeof true;	_____
typeof 'abc';	_____	typeof 100;	_____
typeof NaN;	_____	typeof null;	_____
var fn = function(){}; typeof(fn);	_____		

(3) 将以下表达式的结果写在横线处。

Number('674');	_____	Number('674abc');	_____
Number(' ');	_____	Number(true);	_____
Number(null);	_____	parseInt("9.54");	_____
parseFloat('2.31');	_____	parseFloat('a3b');	_____
parseFloat("11.22.33");	_____	parseFloat("4.3abc");	_____

(4) 将以下表达式的结果写在横线处。

'15' - '2';	_____	'15' * '2';	_____
true - 1;	_____	'1' - 1;	_____
'15' + 1;	_____	'15' + true;	_____

2. 编程练习

通过提示框输入两个操作数和一个操作符, 实现简单的数据计算功能, 并将运算结果显示在页面中。请将程序的主要代码写在下面。



知识盘点

请对本次学习任务的知识点进行梳理与汇总，填写表 1-7。

表 1-7 知识点梳理与汇总

学习任务名称：输入输出简单数据			
知识点描述：本次学习任务介绍了几种常见的输出方式，各种输出方式有各自不同的用途。			
知识点	内容盘点		权重
1. 警告框	（举例说明警告框的用法）		10%
2. 确认框	（举例说明确认框的用法）		10%
3. 提示框	（举例说明提示框的用法）		10%
4. 在页面中输出信息	（举例说明如何在页面中输出信息）		10%
5. JavaScript 数据类型检测常用方法	（举例说明数据类型检测函数的用法）		10%
6. 强制转换为 Number 类型的方法	（举例说明如何将其他类型数据转换为 Number 类型数据）		15%
7. 强制转换为 String 类型的方法	（举例说明如何将其他类型数据转换为 String 类型数据）		20%
8. 强制转换为 Boolean 类型的方法	（举例说明如何将其他类型数据转换为 Boolean 类型数据）		15%
学习评价			
评价内容	自评	互评	师评
1. 知识点掌握情况（40%）			
2. 实践练习完成情况（60%）			
本人签名：			

学习任务 4

输入输出表达式

任务目标

- ◆ 掌握 JavaScript 语言中算术运算符及表达式
- ◆ 掌握 JavaScript 语言中比较运算符及表达式
- ◆ 掌握 JavaScript 语言中逻辑运算符及表达式
- ◆ 掌握 JavaScript 语言中赋值运算符及表达式
- ◆ 掌握 JavaScript 语言中三元运算符的使用方法
- ◆ 了解 JavaScript 语言中运算符的优先级



扫一扫，观看视频

知识介绍

在 JavaScript 语言中，运算符也称为操作符，用于算术运算、比较运算、逻辑运算、赋值运算等。JavaScript 常用的运算符如表 1-8 所示。使用不同的运算符可以组成不同类型的表达式。

表 1-8 JavaScript 常用的运算符

运算符类别	运算符或语法格式
算术运算符	+, -, *, /, % (取余), ** (乘方), ++, --
比较运算符	==, ===, !=, !==, >, <, >=, <=
逻辑运算符	&& (与), (或), ! (非)
赋值运算符	=, +=, -=, *=, /=, %=
三元运算符	条件表达式 ? 表达式 1 : 表达式 2

对于这些运算符，有以下 5 点需要注意。

1. 算术运算符

算术运算符中，取余运算符“%”计算结果的正负号由第一个操作数的正负决定，示例代码如下。

```
document.write(-5% 2);           //显示结果： -1
document.write(5 % -2);          //显示结果： 1
```

2. 比较运算符

比较运算符中，“==”“===”是有区别的。“==”是相等运算符，会自动转换数据类型，变成相同的数据类型后再比较两个数据的值是否相等；“===”是严格相等运算符，不会自动转换数据类型，所以它既比较数值，又比较数据类型，用来比较两个操作数的值和类型是否都相同。

3. 逻辑运算符

(1) 逻辑运算符中，与运算符“&&”的运算规则是：如果第一个表达式的布尔值为 true，则返回第二个表达式的值（注意是值，不是布尔值）；如果第一个表达式的布尔值为 false，则直接返回第一个表达式的值，且不再对第二个表达式求值，示例代码如下。

```
document.write('t' && 'f');           //结果: "f"
document.write('t' && (1 + 2));        //结果: 3
```

(2) 逻辑运算符中，或运算符“||”的运算规则是：如果第一个表达式的布尔值为 true，则返回第一个表达式的值，且不再对第二个表达式求值；如果第一个表达式的布尔值为 false，则返回第二个表达式的值，示例代码如下。

```
document.write('' || 'f');           //结果: "f"
document.write(true || (3 == 2));     //结果: true
```

4. 三元运算符

三元运算符是 JavaScript 语言中唯一需要 3 个操作数的运算符，能够基于条件表达式的结果执行语句，语法如下。

```
条件表达式 ? 表达式1 : 表达式2
```

如果条件表达式的结果为 true，则返回表达式 1 的值，否则返回表达式 2 的值，示例代码如下。

```
var score = prompt("请输入一个分数值");
document.write(score >= 60 ? "通过":"未通过");
```

上述示例代码表达的含义是：当输入大于或等于 60 的数值时，页面上显示“通过”，当输入小于 60 的数值时，页面上显示“未通过”。

5. 运算符的优先级

当表达式中包含了多种运算符时，应首先明确所有参与运算的运算符优先级，才能计算出正确的结果。运算符优先级如表 1-9 所示，优先级自上而下递减。

表 1-9 运算符优先级

优先级	运算符	结合性
1	[]、.、()	从左向右
2	++、--、~、!、delete、new、typeof、void	从右向左
3	*, /、%	从左向右
4	+, -	从左向右
5	<, <=, >, >=, instanceof	从左向右

续表

优先级	运算符	结合性
6	==、!=、===、!==	从左向右
7	&&	从左向右
8		从左向右
9	?:	从右向左
10	=、+=、-=、*=、/=、%=、&=、 =、^=、<、<=、>、>=、>>=	从右向左
11	,	按优先级计算，从右向左

 实践练习

扫一扫，观看视频

1. 问答题

(1) 在横线处填写下列代码的显示结果。

`document.write(-5% 2);` _____`document.write(5 % -2);` _____`document.write('cat' > 'dog');` _____`document.write('cat' > 100);` _____`document.write("123" == 123);` _____`document.write("123" === 123);` _____`document.write("123" !== 123);` _____(2) 假设 $x=10$, $y=20$, 填写以下表格中表达式的运算结果。

表达式	结果
$x < 20 \ \&\& \ y > 1$	
$x == 20 \ \ y == 5$	
$!(x == y)$	

2. 编程练习

(1) 输入 3 个整数 a 、 b 和 c , 使用三元运算符计算最大值, 输出到页面中。请将主要代码写在下面。

(2) 输入圆的半径, 计算出圆的面积和周长, 输出到页面上。请将主要代码写在下面。

(3) 编写一个收银台收款程序, 输入商品的单价、购买数量和收款金额, 输出应收金额和找零。将主要代码写在下面。



知识盘点

请对本次学习任务的知识点进行梳理与汇总，填写表 1-10。

表 1-10 知识点梳理与汇总

学习任务名称：输入输出表达式			
知识点描述：表达式是程序的基本组成部分，熟练掌握各种表达式的运算规则对于编写分支、循环等语句的程序非常重要。			
知识点	内容盘点		权重
1. JavaScript 常见的运算符	（简述 JavaScript 常见的运算符）		20%
2. “==” “===” 的区别	（简述==运算符与===运算符之间的区别）		20%
3. 与运算和或运算的运算规则	（简述与运算和或运算的运算规则，尤其是“短路”问题）		30%
4. 三元运算符	（举例说明三元运算符的用法）		30%
学习评价			
评价内容	自评	互评	师评
1. 知识点掌握情况（40%）			
2. 实践练习完成情况（60%）			
本人签名：			

