JavaScript+jQuery前端开发基础教程

微课版

第1章 JavaScript基础

本章主要内容：

- JavaScript简介
- JavaScript编程工具
- 在HTML中使用JavaScript
- JavaScript基本语法

1.1 JavaScript简介

- JavaScript是一种轻量的解释型编程语言，具有面向对象的特点，在Web应用中得到广泛使用。所有现代Web浏览器（如Edge、Firefox、Chrome等，后文统称为浏览器）都包含了JavaScript解释器，所以都支持JavaScript脚本。
- 嵌入HTML文档的JavaScript称为客户端的JavaScript，通常简称为JavaScript。当然，JavaScript并不局限于浏览器客户端脚本编写，也可用于服务器、PC客户端和移动客户端等的应用编写。

1.1.1 JavaScript版本

- JavaScript最初由Netscape（网景通信）公司的Brendan Eich（布伦丹 • 艾希）研发。起初，该语言被称为Mocha，但在1995年9月更名为LiveScript，最后在12月，Netscape公司与Sun Microsystems（太阳微系统）公司联合发布的声明中，它被命名为JavaScript，也就是JavaScript 1.0。实现了JavaScript 1.0的Netscape Navigator浏览器2.0版几乎垄断了当时的浏览器市场。
- 因为JavaScript 1.0的巨大成功，Netscape公司在Netscape Navigator浏览器3.0版中实现了JavaScript 1.1。Microsoft公司在进军浏览器市场后，在Internet Explorer（简称IE）3.0中实现了一个JavaScript的克隆版本，并命名为JScript。

- 在Microsoft加入后，有3种不同的JavaScript版本同时存在：Netscape Navigator中的JavaScript、IE中的JScript以及CEnvi中的ScriptEase。这3种JavaScript的语法和特性并没有统一。
- 1997年，JavaScript 1.1作为一个草案提交给欧洲计算机制造商协会（European Computer Manufactures Association，ECMA）。之后，由来自Netscape、SunMicrosystems、Microsoft、Borland公司和其他一些对脚本语言感兴趣的程序员组成的TC39团体推出了JavaScript的“ECMA-262”标准，该标准将脚本语言名称定义为ECMAScript。该标准也被国际标准化组织（International Organization for Standardization，ISO）及国际电工委员会（International Electrotechnica Commission，IEC）采纳，作为各种浏览器的JavaScript语言标准。因此，JavaScript成了事实上的名称，ECMAScript代表了其语言标准。

1.1.2 JavaScript特点

- JavaScript具有下列主要特点。
  - 解释性：浏览器内置了JavaScript解释器。在浏览器中打开HTML文档时，其中的JavaScript代码直接被解释执行。
  - 支持对象：可自定义对象，也可使用各种内置对象。
  - 事件驱动：事件驱动使JavaScript能够响应用户操作，而不需要Web服务器端处理。例如，当用户输入E-mail地址时，可在输入事件处理函数中检查输入的合法性。

1.1.2 JavaScript特点

- 跨平台：JavaScript脚本运行于JavaScript解释器，配置了JavaScript解释器的平台均能执行JavaScript脚本。
- 安全性：客户端JavaScript脚本不允许访问本地磁盘，不能将数据写入服务器，也不能对网络文档进行修改和删除，只能通过浏览器实现信息的浏览和动态展示。Node.js作为一个服务器端JavaScript运行环境，可让JavaScript成为类似于PHP、Python、Ruby等的服务器端脚本语言，可让JavaScript程序实现读写文件、执行子进程以及网络通信等功能。

1.2 JavaScript编程工具

- JavaScript脚本需要嵌入HTML文档，可使用各种工具来编写JavaScript脚本。
- 最简单的工具是Windows的记事本。
- 常用的Web集成开发工具有Visual Studio Code（简称VS Code）、Adobe Dreamweaver、Eclipse和IntelliJ IDEA等。集成开发工具通常具有语法高亮、自动完成、错误检测等功能。
- 本书使用的VS Code是Microsoft推出的免费集成开发工具。

1.2.1　安装VS Code

- 演示过程

1.2.2　使用VS Code

- 演示：在VS Code中创建HTML文档

1.2.3　使用浏览器开发人员工具

- 演示

1.3.1　嵌入式JavaScript脚本

- 嵌入式JavaScript脚本指直接在HTML文档中包含JavaScript脚本，可使用下面的3种方法实现嵌入式JavaScript脚本。
  - 使用<script>标记。
  - 作为事件处理程序。
  - 作为URL。

在HTML中使用JavaScript

- 在HTML文档中，可通过两种方式使用JavaScript脚本：
  - 嵌入
  - 链接

1. 使用<script>标记嵌入JavaScript脚本

- HTML文档中的JavaScript脚本放在<script>和</script>标记之间。
- <script>标记放在 HTML 文档的<head>或<body>部分，当然也可放在其他位置。<script>标记内可包括任意多条JavaScript语句，语句按照先后顺序依次执行，
- 语句的执行过程也是浏览器加载HTML文档过程的一部分。
- 除了函数内部的代码外，浏览器在扫描到JavaScript语句时就会立即执行该语句。
- 函数内部的代码在调用函数时执行。

- 一个HTML文档可以包含任意多个<script>标记，<script>不能嵌套和交叉。
- 不管有多少个<script>标记，对HTML文档而言，它们包含的JavaScript语句组成一个JavaScript程序。
- 在一个<script>标记中定义的变量和函数，可在后续的<script>标记中使用。

language和type属性

- <script>标记的language和type属性（前者已被后者取代）可用于指定脚本使用的编程语言及其版本。
  - <script language="javascript"></script>
  - <script language="javascript 1.5"></script>
  - <script type="text/vbscript"></script>
- 脚本语言及其版本被指定后，如果浏览器不支持，则会忽略<script>标记内的脚本代码。

- 早期的脚本语言除了JavaScript外，还有VBScript。目前，绝大多数新的浏览器不再支持VBScript。
- JavaScript已成为事实上的唯一客户端HTML脚本编程语言。
- 所以，可不在<script>标记中指定脚本语言。

• </script>标记

- </script>标记表示一段脚本的结束。不管</script>标记出现在何处，浏览器均将其视为脚本的结束标记。
- <script>
- document.write("<script>")
- document.write("document.write('页面中输出脚本')")
- document.write("</script>")
- </script>

• defer属性

- 在<script>标记中使用defer属性时，文档加载完成后浏览器才执行脚本。
  - <script defer></script>
- 当然，如果在脚本中有内容输出到页面，defer属性会被忽略，脚本立即执行。

2．作为事件处理程序

- JavaScript脚本代码可直接作为事件处理程序代码。
- 例如：
  - <input type="button" value="请单击按钮" onclick="a = 1; b = 2
    ('单击按钮执行JavaScript语句弹出对话框\na+b='+(a+b))"/>

3．作为URL

- 在HTML文档中，使用“javascript”作为协议名称时，可将JavaScript语句作为URL使用。在访问该URL时，JavaScript语句被执行。
- 例如：
  - 请单击此链接

1.3.2　链接JavaScript脚本

- <script>标记的src属性用于指定链接的外部脚本文件。
- 通常，基于下列原因将JavaScript脚本放在外部文件中。
  - 脚本代码较长，移出HTML文档后，可简化HTML文档。
  - 脚本中的代码和函数需要在多个HTML文档间共享。将共享代码放在单个脚本文件中可节约磁盘空间，利于代码维护。
  - 多个HTML文档共享的函数在第1次被调用时，该函数被缓存，后续HTML文档可直接使用缓存中的函数，加快网页加载速度。
  - <script>标记的src属性值可以是任意的URL。这意味着可使用来自Web服务器的JavaScript脚本文件，或者是由服务器脚本动态输出的脚本。

- 独立的JavaScript脚本文件扩展名通常为“.js”，“.js”文件只包含JavaScript代码，没有<script>和HTML标记。
- 浏览器会将文件中的代码插入<script>和</script>标记之间。
- <script src="test1-5.js"></script>

1.4 JavaScript基本语法

1.4.1　区分大小写

- JavaScript对大小写敏感，使用过程中需要严格区分关键字、变量、函数以及其他标识符的大小写。

```
<script>
        a  =  100
        A  =  200
        document.write(a)
        document.write("<br>")
        document.write(A  )
    </script>
```

1.4.2　可忽略空格、换行符和制表符

- JavaScript会忽略代码中不属于字符串的空格、换行符和制表符。
- 通常，空格、换行符和制表符用于帮助代码排版，方便阅读程序。

```
<script>
                a   =
                        100
                document.
                        write(a)
</script>
```

1.4.3　不强制使用语句结束符号

- JavaScript并不强制要求语句末尾使用分号“;”来作为语句结束符号。
- JavaScript解释器可自动识别语句结束。
- 在某些时候，可使用分号将多条语句写在同一行。
  - <script>
  - 　　a　=100;　document.write(a)
  - </script>

1.4.4 注释

- 注释是程序中的说明信息，帮助理解代码。脚本执行时，注释内容会被忽略。
- JavaScript提供两种注释。
  - //：单行注释。//之后的内容为注释。
  - /*……*/：多行注释。在“/*”和“*/”之间的内容为注释，可以占据多个语句行。

```
<script>
        /*
        【例1-8】  在JavaScript脚本中使用注释
        下面的代码用于说明JavaScript对大小写敏感
        */
        a  =  100                                   //变量赋值
        A  =  200                                   //变量赋值
        document.write(a)                           //将变量值输出到页面
        document.write("<br>")                      //在页面中输出一个换行标记，将两个变量值分开
        document.write(A)                           //将变量值输出到页面
    </script>
```

1.4.5　标识符命名规则

- 标识符用于命名JavaScript中的变量、函数或其他对象。
- JavaScript标识符命名规则与Java相同：
  - 第1个字符必须是字母、下划线、美元符号或者汉字，后面的字符可以是字母、数字、下划线或者汉字。
  - JavaScript使用Unicode字符串，所以允许使用包含中文在内的各国语言字符。

- 例如，下面都是合法的标识符。

A

_data

$price

var1

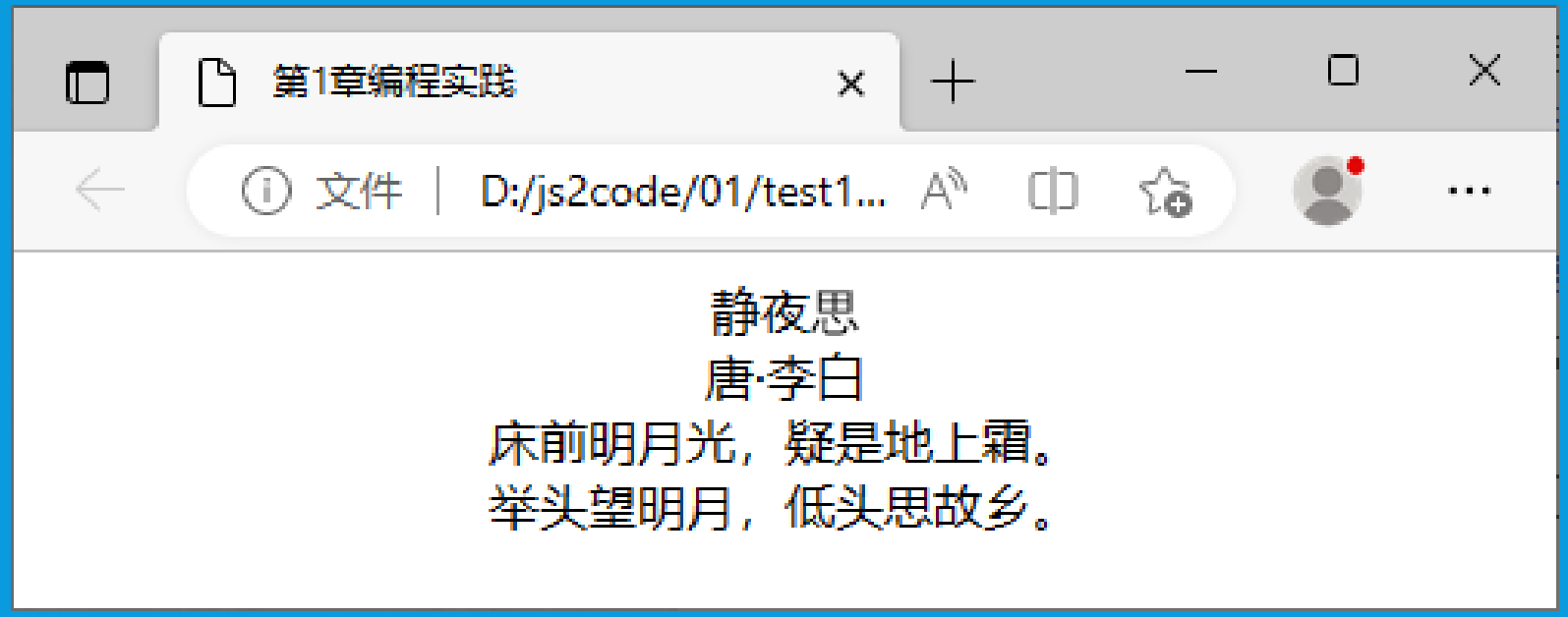
价格

1.4.6　输入和输出语句

- JavaScript常用的输入和输出语句如下。
  - document.write(msg)：将参数msg输出到Web页面的当前位置。
  - console.log(msg)：将参数msg输出到浏览器控制台。
  - alert(msg)：在浏览器中弹出警告对话框，参数msg作为警告信息显示。
  - prompt(msg)：在浏览器中弹出输入对话框，参数msg作为输入提示信息显示。

1.5 编程实践：在页面中输出唐诗

- 本节综合应用本章所学知识，使用JavaScript脚本在Web页面中输出唐诗《静夜思》，如图1-29所示。



JavaScript+jQuery前端开发基础教程

微课版

第2章 JavaScript核心语法基础

本章主要内容：

- 数据类型
- 变量
- 运算符和表达式
- 流程控制语句

2.1 数据类型和变量

- 程序中最基础的元素是数据和变量。数据类型决定了程序如何存储和处理数据，变量则是数据的“存储仓库”。

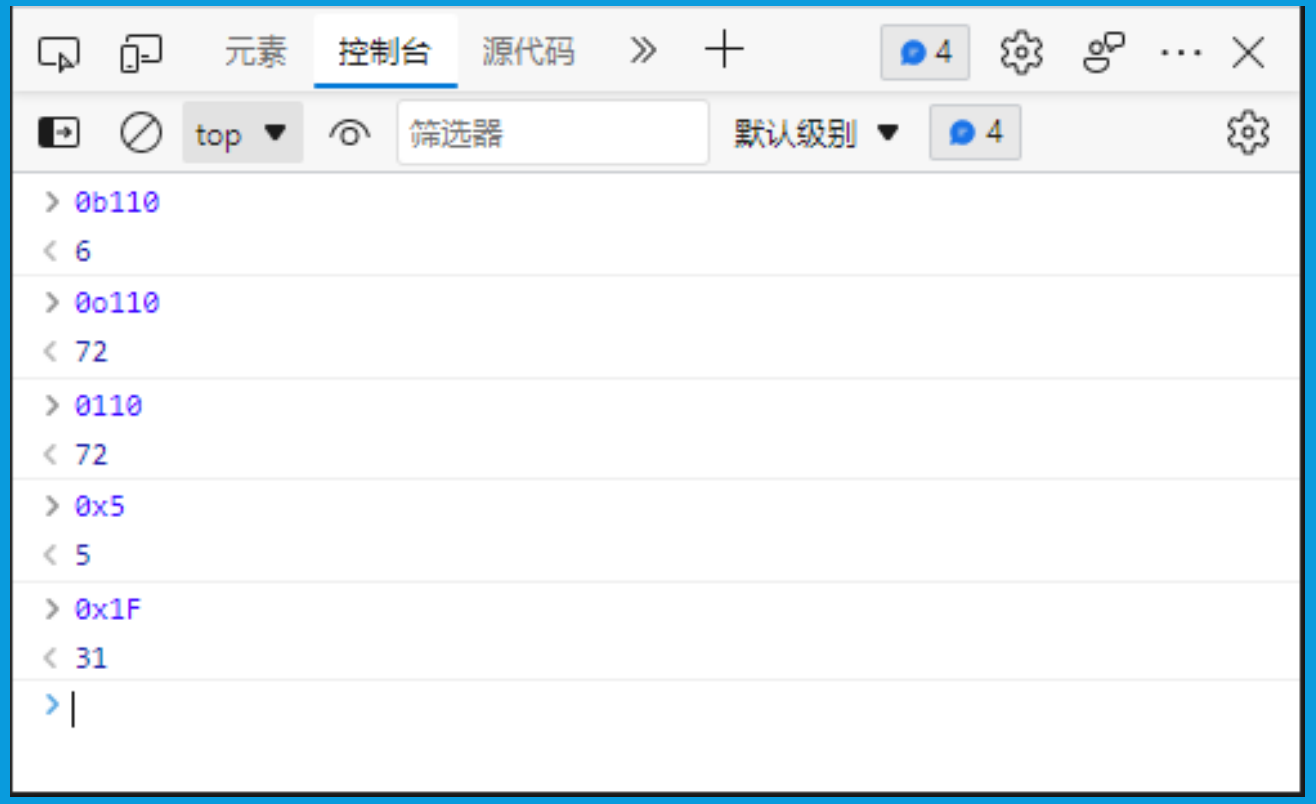
2.1.1　数据类型

- JavaScript数据类型可分为两类：基本类型和引用类型。
- 基本类型也称原始数据类型，包括：
  - number（数值）
  - string（字符串）
  - boolean（布尔值）
  - null（空值）
  - undefined（未定义）
  - symbol（符号）
- 引用类型也称复杂数据类型，包括object(对象）和function（函数）。函数实质上是对象的子类型。

1. 数值常量

- 在程序中直接使用的值称为字面量或常量。
- 数值常量支持十进制数、二进制数、八进制数和十六进制等记数形式。
- 十进制：人们常用的记数进制，使用0～9的数码表示数值。
- 二进制：以0b开头，使用0、1表示数值，例如：0b110、0b1001。
- 八进制：以数字0或0o开头，使用0～7等数码表示数值，例如05、0o10、017。
- 十六进制：以0x或0X开头，使用0～9、a～f、A～F等数码表示数值，例如0x5、0x1F。

- 在Edge浏览器控制台中输入各种进制数据，输出为对应的十进制数



- ES2020（即ECMAScript 2020）为JavaScript定义了一种新的数值类型bigint，用于表示64位整数。数值末尾的小写字母n表示这是一个bigint值。例如：10n、0b110n、0x1Fn。
- 数值常量包含小数，例如2.25、1.7。如果整数部分为0，JavaScript允许省略小数点前面的0，如0.25可表示为.25。
- 数值常量可用科学记数法表示，如1.25e-3、2.5E2。

JavaScript的特殊数值

- Infinity：Infinity表示正无穷大，-Infinity表示负无穷大。在非零数值除以0时就会出现无穷大。当一个正值超出JavaScript的表示范围时，其结果就是正无穷大。
- NaN：意思为“非数字”——Not a Number，表示数值运算时出现了错误或者未知结果。例如，0除以0的结果为NaN。
- Number.MAX_VALUE：最大数值。
- Number.MIN_VALUE：最小数值。
- Number.NaN：NaN。
- Number.POSITIVE_INFINITY：Infinity。
- Number.NEGATIVE_INFINITY：-Infinity。

2. 字符串常量

- JavaScript使用Unicode字符集。字符串常量指用英文的双引号（″）或单引号（’）括起来的一串Unicode字符，如″Java″或’15246’。
- 只能成对使用单引号或双引号作为字符串定界符，不能使用一个单引号和一个双引号。如果需要在字符串中包含单引号或双引号，则应用另一个作为字符串定界符或者使用转义字符。例如，″I like ’JavaScript’″。
- 字符串中可以使用转义字符，转义字符以“\”开始。例如，“\n”表示换行符，“\r”表示回车符。表2-1列出了JavaScript的转义字符。

表 2-1　JavaScript 的转义字符

转义字符	说明
\0	空字符，Unicode 编码为 0
\b	退格符
\n	换行符
\r	回车符
\t	制表符
\"	双引号
\'	单引号
\\	\
\XXX	XXX 为 3 位八进制数，表示字符的 Unicode 编码，如\101 表示字符 A
\xXX	XX 为两位十六进制数，表示字符的 Unicode 编码，如\x41 表示字符 A
\uXXXX	XXXX 为 4 位十六进制数，表示字符的 Unicode 编码，如\u0041 表示字符 A

- 3. 布尔型常量
- 布尔型常量只有两个：true和false（注意必须小写）。
- 4. null
- null在JavaScript中表示空值。

- 5. undefined
- 用var声明一个变量后，其默认值为undefined。
  - var a
  - document.write(a)　　//输出结果为undefined

6. 类型测试

- typeof运算符可测试数据的类型。
  - typeof(123)　　//结果为number
- 需要特别说明的是：typeof(null)结果为object，正确的结果应该是null。
- 这是JavaScript由来已久的一个bug，修复这个bug可能会产生更多的bug，导致现有的很多Web系统无法使用。所以，JavaScript一直未修复这个bug。

2.1.2　数据类型转换

- JavaScript中的数据类型转换包括隐式类型转换和显式类型转换。
- 1. 隐式类型转换
- 当JavaScript执行代码需要特定类型的数据，而提供的不是该类型的数据时，JavaScript就会根据需要转换数据的类型，这就是隐式类型转换。
  - `5 + 'x'`　　//结果为'5x'：数值5转换为字符串
  - `5 - '3'`　　//结果为2：字符串'3'转换为数值
  - `true + 'Abc'`　　//结果为'trueAbc'，布尔值true转换为字符串