

第7章 JavaScript

本章讲述的主要内容

- 7.1 JavaScript的基本概念
- 7.2 JavaScript程序设计基础
- 7.3 JavaScript对象
- 7.4 BOM和DOM对象
- 7.5 实训 设计浮动图片效果

7.1 JavaScript的基本概念

7.1.1 JavaScript概述

JavaScript是一种通用的、跨平台的、基于对象和事件驱动的客户端脚本语言，其主要特点有以下几个方面。

- (1) 弱数据类型
- (2) 跨平台
- (3) 基于对象Transformations, 扩展样式表转换) 进行数据交换及操作。
- (4) 基于事件驱动

7.1 JavaScript的基本概念

7.1.1 JavaScript概述

JavaScript代码格式不够严谨，使用比较灵活，但过于随意将会导致代码的可读性降低，不易于后期维护和升级。因此，在编写JavaScript程序时应遵守以下规范。

- ① 书写代码时应注意，浏览器解析JavaScript代码时会忽略标识符与运算符之间多余的空格。
- ② 书写代码时每条语句一般应独占一行，并以英文分号“;”为结束符。
- ③ 代码要使用缩进格式编写，以增强其层次感和可读性。
- ④ 代码中可以使用“//”表示单行注释，使用“/*……*/”表示多行注释。

7.1 JavaScript的基本概念

7.1.2 JavaScript的代码编写规范

(1) 行内JavaScript语句

直接将JavaScript代码写在HTML元素的开始标记中。

(2) 嵌入式JavaScript语句块

将JavaScript代码写在<script>……</script>中。

(3) 使用独立的.js文件

```
<script type="text/javascript" src="JavaScript 文件的 URL"></script>
```

7.2 JavaScript程序设计基础

7.2.1 数据类型和变量

1. 数据类型

表 7-1 JavaScript 常用数据类型及说明

数据类型	说明
String	字符串型，表示由 0~N 个 Unicode 字符组成的字符序列，字符串数据可以用英文单引号或双引号表示。
Number	数值型，可以是 32 位的整数或 64 位的浮点数，整数可以是十进制、八进制或十六进制的形式。
Boolean	布尔型，只包括 true 和 false 两个值，表示表达式的“真”或“假”。
Null	Null 是一种特殊类型，用来表示不存在的对象，或者说它表示了一个空对象引用。
Undefined	未定义类型，通常用 Undefined 表示变量不包含任何有效值。
Array	数组类型，表示一系列变量或对象的集合，这些变量或对象的类型可以是不相同的。
Function	函数类型，它是由事件驱动的或者被调用时执行的，可重复使用的代码块。
Object	对象类型，在 JavaScript 中，对象是指拥有属性和方法的数据。

7.2 JavaScript程序设计基础

7.2.1 数据类型和变量

2. 变量和运算符

(1) 声明变量和为变量赋值

```
var username, userpwd; → //声明 2 个变量。  
username = "zhangsan"; → //为变量赋值。  
userpwd = "123456";  
var age = 21; → //声明变量的同时为变量赋值
```

(2) 运算符

JavaScript使用的运算符与C#语言的运算符大部分相同。如，**算术运算符**+、-、*、/；**比较运算符**<、>、==、>=、<=、!=；**布尔运算符**&&、||、! 等。

7.2 JavaScript程序设计基础

7.2.1 数据类型和变量

3. 变量的作用域

变量的作用范围是指可以访问该变量的代码区域。

JavaScript中按变量的作用范围分为全局变量和局部变量。

① **全局变量**：可以在整个HTML文档范围中使用的变量，这种变量通常都是在函数体外定义的变量。

② **局部变量**：只能在局部范围内使用的变量，这种变量通常都是在函数体内定义的变量，所以只在函数体内部有效。

省略关键字var声明的变量（未定义直接使用的变量），无论在函数体内部还是外部，都是全局变量。

7.2 JavaScript程序设计基础

7.2.2 流程控制语句

1. 分支结构

if…else if…else、switch语句

2. 循环结构

JavaScript的循环结构主要由for语句、while语句和**for in**语句构成。其中，for语句和while语句与C#中的语法格式及使用方法完全相同，这里不再赘述。C#中的foreach语句在JavaScript中稍有变化，其语法格式如下所示。

```
for(循环变量·in·数据集)
{
    ……循环体语句;
}
```


7.2 JavaScript程序设计基础

7.2.3 JavaScript函数

1. JavaScript内置函数

表 7-2 JavaScript 常用内置函数

函数名	说·明
<u>alert</u>	用于显示一个弹出信息框。
<u>confirm</u>	用于显示一个确认对话框，包括“确定”和“取消”两个按钮。语法格式： <code>confirm("提示文本");</code>
<u>prompt</u>	用于显示一个输入对话框，提示等待用户输入。语法格式： <code>prompt("提示文本","缺省值");</code>
<u>eval</u>	用于计算表达式的结果。
<u>isNaN</u>	表示“is Not a Number”用于测试变量是否为一个数字，返回值为 true 表示不是一个数字。

7.2 JavaScript程序设计基础

7.2.3 JavaScript函数

2. 自定义函数

JavaScript除了可以使用预定义函数外，还可以根据需要自定义用于实现特定功能的函数。由于JavaScript是弱类型脚本程序设计语言，故在定义函数时无需声明函数的参数类型和返回值类型。JavaScript的自定义函数分为命名函数、匿名函数、对象函数和自调用函数四种。

自定义函数代码可以书写在<script></script>标记之间，也可以书写在.js文件中。在同一个<script></script>标记中，函数定义可以书写在调用语句之前或之后；但在不同的<script></script>标记中函数定义语句只能书写在调用语句之前。

7.2 JavaScript程序设计基础

7.2.3 JavaScript函数

2. 自定义函数

(1) 命名函数

```
function 函数名([形参列表]) → //存在多个参数时应使用英文逗号分隔  
{  
    ... 函数体语句;  
    ... [return 返回值;] → //可选项，函数可以有，也可以没有返回值  
}
```

7.2 JavaScript程序设计基础

7.2.3 JavaScript函数

2. 自定义函数

(2) 匿名函数

```
function([形参列表])  
{  
    ... 函数体语句;  
    ... [return 返回值;] → //可选项  
}
```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/605301244103011131>