



JavaScript基础

Dr. Zhili Zhao

Email: zhaozhl@lzu.edu.cn
School of Information Science & Engineering
<http://xxxxy.lzu.edu.cn>



JavaScript是目前万维网上最流行的脚本语言（Script Language）。

JavaScript使Web浏览器及客户机端（Client-Side）具备了编写和运行程序的能力。使用JavaScript，可以向HTML文档写入HTML代码、验证表单数据、创建Cookie和为WWW页面添加动态特性。

- JavaScript放置
- 变量和类型
- 运算符与表达式
- 全局变量和局部变量
- 数据类型转换
- JavaScript中的流程控制语句

10.1 JavaScript放置



JavaScript既可以出现在HTML文档的**主体或头部**，也可以单独保存在**外部脚本文档**中。



使用script元素，可以在HTML文档主体直接插入JavaScript。**JavaScript可以向HTML文档写入包括元素标签和文本内容在内的HTML代码。**

【例10-1】在Web浏览器中输出乘法表



```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
  <title>在HTML文档主体直接插入JavaScript</title>
</head>

<body>
  <script type="text/javascript">
    var i,j
    //使用document对象的write方法，可以向HTML文档写入元素标签和文本内容
    document.write("<h1>乘法表</h1>")
    for(i=1;i<=9;i++){
      for(j=1;j<=i;j++){
        document.write(i+"*"+j+"="+i*j+" ")
      }
      document.write("<br/>")
    }
  </script>
</body>
</html>
```

10.1.2 在IE浏览器中调试JavaScript



从IE 8开始，微软在**IE浏览器**中**内置了一个JavaScript调试器**（Debugger），但在默认状态下该调试器是关闭的。可以按照以下步骤启动IE浏览器内置的JavaScript调试器。

在“开发人员工具”窗口中，可以在JavaScript中**设置断点、启动调试、停止调试和查看局部变量的当前值**。

在程序设计中，**函数（Function）**是一个具有**可重用性**的语句块，**可反复用来完成某个特定功能**。

通常将需要反复执行的多条语句组织在一个函数中，而在需要时通过函数调用执行其中的语句。因此，函数能够实现程序设计的模块化。

JavaScript既提供**系统预先定义的函数**，也允许**用户根据需要自行定义函数**。

在JavaScript中，可以按照以下语法定义函数：

```
function function_name(arg1,arg2,...,argN){  
    statements  
    [return expression ]  
}
```


10.1.4 在HTML文档头部定义JavaScript函数



兰州大学

除在HTML文档主体直接插入JavaScript外，也可以**首先在HTML文档头部定义JavaScript函数、然后在HTML文档主体通过函数调用执行JavaScript函数中的语句。**

【例10-2】在HTML文档头部定义函数
printMultiplicationTable，该函数能够在Web浏览器中输出乘法表，然后在HTML文档主体调用该函数

10.1.5 在外部脚本文档定义JavaScript函数



很多时候，需要在多个HTML文档中执行一组相同的JavaScript语句。此时，可以以函数形式将这些JavaScript语句组织和保存在以js为扩展名的脚本文档中。然后，在HTML文档中调用外部脚本文档中的JavaScript函数。这样，在多个HTML文档中即可轻松地执行相同的JavaScript语句。

【例10-3】在HTML文档中调用外部脚本文档中的JavaScript函数



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
    <title>在XHTML文档中调用在外部脚本文档中定义的JavaScript函数</title>

    <!--在script元素的开始标签中使用src属性指定外部脚本文档-->
    <script type="text/javascript" src="10-3.js"></script>
  </head>

  <body>
    <script type="text/javascript">
      //调用在外部脚本文档中定义的JavaScript函数printMultiplicationTable
      printMultiplicationTable()
    </script>
  </body>
</html>
```

- 同其它程序设计语言类似，在JavaScript中，**变量（Variable）**用于存储可以变化的数据。
- 在JavaScript中，可以使用**保留字var定义变量**。使用一个保留字var可以同时定义多个变量。
- 在JavaScript中，变量名的书写必须遵循以下两条**规则**：
 - 变量名是大小写敏感的。例如，x和X代表两个不同的变量。
 - 变量名必须以字母、下划线（_）或美元符号（\$）开始。

在JavaScript中，变量或数据有五种原始类型（Primitive Type）。每种类型代表不同性质的数据。

1. **undefined**类型。这种类型的值只有一个，即**undefined**。当一个变量仅仅被定义、但还没有被赋值时，该变量的类型和值都是undefined。
2. **object**类型。这种类型只有一个值，即**null**。
3. **boolean**类型。这种类型有两个值，即**true**和**false**。
4. **string**类型。这种类型是一种没有固定大小的原始类型，并且这种类型的值由一个字符串构成。
5. **number**类型。这种类型支持整数和浮点数，并且这种类型的变量或数据可以参与算术运算。

在JavaScript中，使用**typeof运算符**可以返回变量的类型。

【例10-4】使用typeof运算符测试变量的类型。

10.3 运算符与表达式

表 10-1 算术运算符与算术表达式

算术运算符	含义	算术表达式 (假设 $a=2$)	表达式的值
+	加	$a+3$	5
-	减	$a-1$	1
*	乘	$a*2$	4
/	除	$a/2$	1
%	取模 (求余)	$a\%2$	0
++	自增	$a++$, $++a$	2 , 3
--	自减	$a--$, $--a$	2 , 1

10.3 运算符与表达式

表 10-2 关系运算符与关系表达式

关系运算符	含义	关系表达式 (假设 $x=55$, $y=88$, $z="55"$)	表达式的值
<code>==</code>	是否等于	$x==y$, $x==z$	false , true
<code>===</code>	是否全等	$x===z$	false
<code>!=</code>	是否不等于	$x!=y$	true
<code>!==</code>	是否不全等于	$x!==z$	true
<code>></code>	是否大于	$x>y$	false
<code><</code>	是否小于	$x<y$	true
<code>>=</code>	是否大于等于	$x>=y$	false
<code><=</code>	是否小于等于	$x<=y$	true

10.3 运算符与表达式

表 10-3 逻辑运算符及逻辑表达式

逻辑运算符	说明	逻辑表达式 (假设 $x=6, y=3$)	表达式的值
<code>&&</code>	与	$(x < 10) \&\& (y > 1)$	true
<code> </code>	或	$(x == 5) (y == 5)$	false
<code>!</code>	非	$!(x == y)$	true

10.3 运算符与表达式



对于string类型的变量或数据，可以使用**字符串连接运算符“+”**进行连接（Concatenation）运算，这样可以将两个字符串前后连接起来。

例如，以下JavaScript代码

```
var s1="Java", s2="Script", s3;  
s3=s1+s2;  
document.write("s1="+s1+" s2="+s2+" s1+s2="+s3+"<br/>");
```

可以在Web浏览器中输出：

s1=Java s2=Script s1+s2=JavaScript

10.3 运算符与表达式

表 10-4 赋值运算符

赋值运算符	赋值语句 (x=12, y=5)	等价的赋值语句	运算结果
=	x=y		x=5
+=	x+=y	x=x+y	x=17
-=	x-=y	x=x-y	x=7
=	x=y	x=x*y	x=60
/=	x/=y	x=x/y	x=2.4
%=	x%=y	x=x%y	x=2

【例10-5】 验证运算符与表达式。

条件运算符 (?:) 是JavaScript中常见的运算符，并且经常和赋值运算符共同构成赋值语句，其基本形式为：

```
variable = boolean_expression ? true_value : false_value;
```

该赋值语句将根据布尔表达式boolean_expression的值有条件地给变量variable赋值。如果布尔表达式boolean_expression的值为true，就将表达式true_value的值赋给变量variable；如果布尔表达式boolean_expression的值为false，就将表达式false_value的值赋给变量variable。

10.3 运算符与表达式



以下赋值语句

```
var iMax = (iNum1 > iNum2) ? iNum1 : iNum2;
```

等价于如下if-else语句：

```
if (iNum1 > iNum2) iMax = iNum1;  
else iMax = iNum2;
```

10.4 全局变量和局部变量



在JavaScript中，变量可以分为全局变量和局部变量。在函数外部定义的变量属于全局变量，在函数内部定义的变量以及函数定义中的形式参数属于局部变量。

全局变量既可以在函数外部赋值和引用，也可以在函数内部赋值和引用。局部变量只能在函数内部定义、赋值和引用，而不能在函数外部赋值和引用。

【例10-6】理解全局变量以及局部变量



```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=gb2312"/>
    <title>全局变量和局部变量</title>
  </head>
  <body>
    <script type="text/javascript">
      var v1=0; //全局变量
      document.write("全局变量v1的初始值: "+v1+"<br/>");

      function f1() {
        var v2=100; //局部变量
        v1=v1+1;
        document.write("局部变量v2的值: "+v2+"<br/>");
      }

      f1();
      document.write("第一次函数调用后, 全局变量v1的值: "+v1+"<br/>");

      f1();
      document.write("第二次函数调用后, 全局变量v1的值: "+v1+"<br/>");
      //document.write("局部变量v2的值是 "+v2+"<br/>"); 此条语句将出错
    </script>
  </body>
</html>
```

在JavaScript中，有时候需要将保存在某些变量中的数据转换为另一种类型的数据。

10.5.1 隐式数据类型转换



例如,

```
var b=true,n1=100, n2=3.14;  
var s="JavaScript "+b+" "+n1+" "+n2;
```

此时, 运算符 “+” 会被Web浏览器解析为字符串连接运算符, 表达式中boolean或number类型的数据会被隐式地转换为string类型的数据, 然后参与字符串的连接运算。

执行第2条语句后, string类型变量s的值为 “JavaScript true 100 3.14”。

10.5.2 显式数据类型转换



在JavaScript中，**boolean**或**number**类型的变量也可以作为一个**对象**，此时可以使用这些对象的**toString**方法将其中的数据显式地转换为**字符串**。例如，

```
var b=true,n1=100, n2=3.14;  
var s="JavaScript "+b.toString()+" "+n1.toString()+" "+n2.toString();
```

执行第2条语句后，string类型变量s的值为"JavaScript true 100 3.14"。

【例10-7】 将boolean或number类型的数据转换为字符串。

10.5.2 显式数据类型转换



在JavaScript中，使用函数`parseInt`或`parseFloat`可以将字符串显式地转换为整数或浮点数。

`parseInt`函数会从左向右逐个字符解析整个字符串，具体过程如下：首先分析第一个字符，判断它是否是数字字符；如第一个字符不是数字字符，该函数将停止解析字符串，并返回NaN，表示“Not a Number”；如果第一个字符是数字字符，该函数将继续逐个分析后续字符……这一过程将持续到发现非数字字符或最后一个数字字符为止，此时该函数将把之前的数字字符串转换成相应的整数。

`parseFloat`函数和`parseInt`函数的工作原理类似。

【例10-8】将字符串转换为整数或浮点数



```
<body>
  <script type="text/javascript">
    var s1="16", s2="1.6";
    var n=parseInt(s1)+parseInt(s2);
    document.write("number类型变量n的值是: "+n+"<br/>");
    n=parseFloat(s1)+parseFloat(s2);
    document.write("number类型变量n的值是: "+n+"<br/>");

    document.write("parseInt的返回值是: "+parseInt("16sd")+"<br/>");
    document.write("parseInt的返回值是: "+parseInt("sd16")+"<br/>");

    s1="1.6.7";
    document.write("parseFloat(s1)的返回值是: "+parseFloat(s1)+"<br/>");
    s1=".16sd";
    document.write("parseFloat(s1)的返回值是: "+parseFloat(s1)+"<br/>");
    s1="sd1.6";
    document.write("parseFloat(s1)的返回值是: "+parseFloat(s1)+"<br/>");

    document.write("parseInt的返回值是: "+parseInt("016")+"<br/>");
    document.write("parseInt的返回值是: "+parseInt("0xA0")+"<br/>");
    document.write("parseInt的返回值是: "+parseInt("16", 8)+"<br/>");
    document.write("parseInt的返回值是: "+parseInt("A0", 16)+"<br/>");
    document.write("parseInt的返回值是: "+parseInt("A0")+"<br/>");
  </script>
</body>
```

与C和Java等程序设计语言类似，JavaScript支持结构化程序设计——不仅可以实现顺序结构（ Sequential Structure ）、选择结构（ Selection Structure ）和循环结构（ Repetition Structure ）等三种基本的程序结构，而且能够对程序流程进行选择控制和循环控制。

选择结构又称**分支结构**，或**选取结构**。使用选择结构，能够**在不同条件下**执行相应的数据处理任务。

选择结构又可进一步分为**单分支**、**双分支**和**多分支**等三种类型。

10.6.1 选择控制语句



使用if语句可以实现单分支选择结构，if语句的语法格式如下：

```
if ( boolean-expression )  
    statement | statement-block
```

使用if-else语句可以实现双分支选择结构，if-else语句的语法格式如下：

```
if ( boolean-expression )  
    statement | statement-block  
else  
    statement | statement-block
```

10.6.1 选择控制语句



使用**switch**语句可以实现**多分支选择结构**，**switch**语句的语法格式如下：

```
switch ( integral-expression ){  
    case integral-value1: statement1;  
        break;  
    case integral-value2: statement2;  
        break;  
    .....  
    case integral-valuen: statementn;  
        break;  
    default: statement;  
}
```

10.6.2 循环控制语句



在JavaScript中，循环控制语句主要有三种，分别是while、do-while和for语句。

while语句的语法格式如下：

```
while ( boolean-expression )  
    statement | statement-block
```

do-while语句的语法格式如下：

```
do {  
    statements;  
} while ( boolean-expression )
```

思考问题：while语句和do-while语句的区别在于?

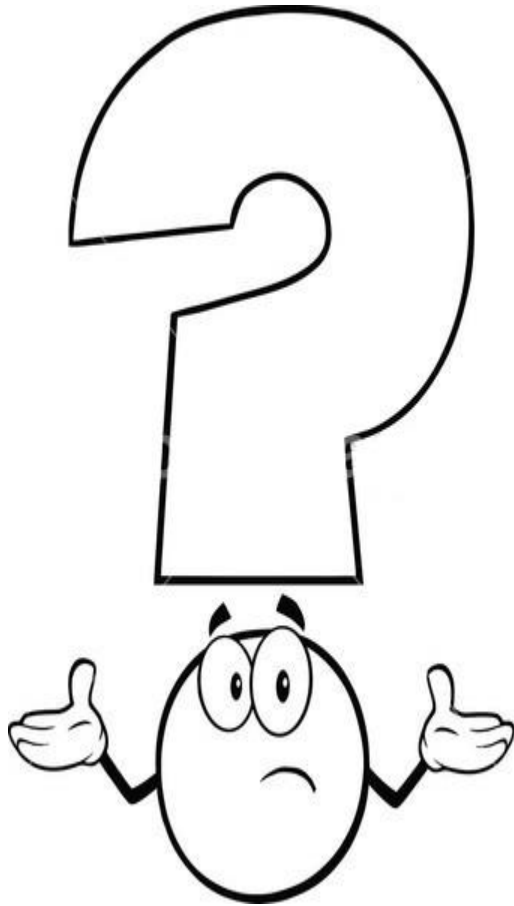
10.6.2 循环控制语句



for语句的语法格式如下：

```
for ( ForInit; boolean-expression; ForUpdate )  
    statement | statement-block
```

- JavaScript是一门轻量级的编程语言，具有短小精悍的特点，它使Web浏览器具备了运行程序的能力。
- JavaScript是在Web浏览器中运行的。
- JavaScript可以出现在HTML文档的主体或头部，但更多的是单独保存在外部脚本文档中。
- 定义在HTML文档头部的JavaScript函数只能在同一HTML文档中调用，定义在外部脚本文档中的函数可被多个HTML文档调用。
- JavaScript中使用var定义变量时，无须指明变量的类型。
- 全局变量既可以在函数外部赋值和引用，在可以在函数内部赋值和引用。局部变量只能在函数内部赋值和引用。



Questions?