**第一章javaScript概述**

在数百万张页面中，JavaScript 被用来改进设计、验证表单、检测浏览器、创建cookies，等等等等。

JavaScript 是因特网上最流行的脚本语言，并且可在所有主要的浏览器中运行，比方说 Internet Explorer、 Mozilla、Firefox、Netscape、和 Opera。

**什么是 JavaScript？**

* JavaScript 被设计用来向 HTML 页面添加交互行为。
* JavaScript 是一种脚本语言（脚本语言是一种轻量级的编程语言）。
* JavaScript 由数行可执行计算机代码组成。
* JavaScript 通常被直接嵌入 HTML 页面。
* JavaScript 是一种解释性语言（就是说，代码执行不进行预编译）。
* 所有的人无需购买许可证均可使用 JavaScript。

## 1.1．JavaScript 能做什么？

* **JavaScript 为 HTML 设计师提供了一种编程工具**
* HTML 创作者往往都不是程序员，但是 JavaScript 却是一种只拥有极其简单的语法的脚本语言！几乎每个人都有能力将短小的代码片断放入他们的 HTML 页面当中。
* **JavaScript 可以将动态的文本放入 HTML 页面**
* 类似于这样的一段 JavaScript 声明可以将一段可变的文本放入 HTML 页面：document.write("<h1>" + name + "</h1>")
* **JavaScript 可以对事件作出响应**
* 可以将 JavaScript 设置为当某事件发生时才会被执行，例如页面载入完成或者当用户点击某个 HTML 元素时。
* **JavaScript 可以读写 HTML 元素**
* JavaScript 可以读取及改变 HTML 元素的内容。
* **JavaScript 可被用来验证数据**
* 在数据被提交到服务器之前，JavaScript 可被用来验证这些数据。
* **JavaScript 可被用来检测访问者的浏览器**
* JavaScript 可被用来检测访问者的浏览器，并根据所检测到的浏览器，为这个浏览器载入相应的页面。
* **JavaScript 可被用来创建 cookies**
* JavaScript 可被用来存储和取回位于访问者的计算机中的信息。

## 1.2．javaScript简介

### 1.2.1.JavaScript：写入 HTML 输出

<script>

document.write("<h1>This is a heading</h1>");

document.write("<p>This is a paragraph.</p>");

</script>

### 1.2.2. JavaScript：对事件作出反应

<button type="button" onclick="alert('Welcome!')">点击这里</button>

### 1.2.3.JavaScript：改变 HTML 内容

<html>

<body>

<h1>我的第一段 JavaScript</h1>

<p id="demo">

JavaScript 能改变 HTML 元素的内容。

</p>

<script>

function myFunction()

{

x=document.getElementById("demo"); // 找到元素

x.innerHTML="Hello JavaScript!"; // 改变内容

}

</script>

<button type="button" onclick="myFunction()">点击这里</button>

</body>

</html>

### 1.2.4.JavaScript：改变 HTML 样式

<html>

<body>

<h1>我的第一段 JavaScript</h1>

<p id="demo">

JavaScript 能改变 HTML 元素的样式。

</p>

<script>

function myFunction()

{

x=document.getElementById("demo") // 找到元素

x.style.color="#ff0000"; // 改变样式

}

</script>

<button type="button" onclick="myFunction()">点击这里</button>

</body>

</html>

### 1.2.5.JavaScript：验证输入

<html>

<body>

<h1>我的第一段 JavaScript</h1>

<p>请输入数字。如果输入值不是数字，浏览器会弹出提示框。</p>

<input id="demo" type="text">

<script>

function myFunction()

{

var x=document.getElementById("demo").value;

if(x==""||isNaN(x))

{

alert("Not Numeric");

}

}

</script>

<button type="button" onclick="myFunction()">点击这里</button>

</body>

</html>

## 1.3．javaScript实现

HTML 中的脚本必须位于 <script> 与 </script> 标签之间。

脚本可被放置在 HTML 页面的 <body> 和 <head> 部分中。

<script>

alert("My First JavaScript");

</script>

### 1.3.2外部的 JavaScript

<html>

<body>

<script src="myScript.js"></script>

</body>

</html>

## 1.4javaScript输出

### 1.4.1操作 HTML 元素

<html>

<body>

<h1>My First Web Page</h1>

<p id="demo">My First Paragraph.</p>

<script>

document.getElementById("demo").innerHTML="My First JavaScript";

</script>

</body>

</html>

### 1.4.2写到文档输出

<h1>My First Web Page</h1>

<script>

document.write("<p>My First JavaScript</p>");

</script>

### 1.4.3警告

请使用 document.write() 仅仅向文档输出写内容。

如果在文档已完成加载后执行 document.write，整个 HTML 页面将被覆盖：

<html>

<body>

<h1>My First Web Page</h1>

<p>My First Paragraph.</p>

<button onclick="myFunction()">点击这里</button>

<script>

function myFunction()

{

document.write("糟糕！文档消失了。");

}

</script>

</body>

</html>

## 1.5 JavaScript 语句

### 1.5.1 JavaScript 代码块

<html>

<body>

<h1>My Web Page</h1>

<p id="myPar">I am a paragraph.</p>

<div id="myDiv">I am a div.</div>

<p>

<button type="button" onclick="myFunction()">点击这里</button>

</p>

<script>

function myFunction()

{

document.getElementById("myPar").innerHTML="Hello World";

document.getElementById("myDiv").innerHTML="How are you?";

}

</script>

<p>当您点击上面的按钮时，两个元素会改变。</p>

</body>

</html>

### 1.5.2JavaScript 对大小写敏感

JavaScript 对大小写是敏感的。

当编写 JavaScript 语句时，请留意是否关闭大小写切换键。

函数 getElementById 与 getElementbyID 是不同的。

同样同样，变量 myVariable 与 MyVariable 也是不同的。

## 1.6注释

单行注释以 // 开头。

多行注释以 /\* 开始，以 \*/ 结尾。

# 第二章 javaScript基础

## 2.1 变量

在js中可以使用var关键字声明一个变量，可以出现在任何位置。变量声明后才能使用。

其实不声明也可以使用，只是这样不合逻辑。

注意标识符、关键字问题。

变量名问题

var a,b,c=1;

声明（创建） JavaScript 变量

在 JavaScript 中创建变量经常被称为“声明”变量。

您可以通过 **var 语句**来声明 JavaScript 变量：

var x;

var carname;

在以上声明之后，变量并没有值，不过您可以在声明它们时向变量赋值：

var x=5;

var carname="Volvo";

**注释：**在为变量赋文本值时，请为该值加引号。

向 JavaScript 变量赋值

通过赋值语句向 JavaScript 变量赋值：

x=5;

carname="Volvo";

变量名在 = 符号的左边，而需要向变量赋的值在 = 的右侧。

在以上语句执行后，变量 **x** 中保存的值是 **5**，而 **carname** 的值是 **Volvo**。

向未声明的 JavaScript 变量赋值

如果您所赋值的变量还未进行过声明，该变量会自动声明。

这些语句：

x=5;

carname="Volvo";

与这些语句的效果相同：

var x=5;

var carname="Volvo";

重新声明 JavaScript 变量

如果您再次声明了 JavaScript 变量，该变量也不会丢失其原始值。

var x=5;

var x;

在以上语句执行后，变量 x 的值仍然是 5。在重新声明该变量时，x 的值不会被重置或清除。

<html>

<body>

<script>

var pi=3.14;

var name="Bill Gates";

var answer='Yes I am!';

document.write(pi + "<br>");

document.write(name + "<br>");

document.write(answer + "<br>");

</script>

</body>

</html>

例子二

<html>

<body>

<p>点击这里来创建变量，并显示结果。</p>

<button onclick="myFunction()">点击这里</button>

<p id="demo"></p>

<script>

function myFunction()

{

var carname="Volvo";

document.getElementById("demo").innerHTML=carname;

}

</script>

</body>

</html>

**2.2 数据类型**

对于javascript的变量来说是没有类型的。**变量的类型由它包含的数据决定**，**字符串、数字、布尔、数组、对象、Null、Undefined**

引用 Reference

列表 List

完成 Completion。

后三种仅仅作为Javascript运行时中间结果的数据类型，不能再代码中使用。

**字符串**

var answer="Nice to meet you!";

**数字**

var x1=34.00;

**布尔**

var x=true

**数组**

var cars=new Array();

cars[0]="Audi";

cars[1]="BMW";

cars[2]="Volvo";

var cars=new Array("Audi","BMW","Volvo");

**对象**

var person={firstname:"Bill", lastname:"Gates", id:5566};

1.未定义类型

此种类型只有一个变量值 undefined 。任何未被赋值的变量都是这个类型。undefined数据类型不能赋值给其他变量

2.空类型 null

仅有一个值 null。初始化时用

**一个对象类型的访问**

<html>

<body>

<script>

person=new Object();

person.firstname="Bill";

person.lastname="Gates";

person.age=56;

person.eyecolor="blue";

document.write(person.firstname + " is " + person.age + " years old.");

</script>

</body>

</html>

**2.3运算符**

和java基本一样。

多了几个：

1 恒等于

严格比较，不会发生数据转换的操作

2==”2”为true

2===“2”为false

2 逗号表达式

var a,b,c,d;

d=(a=1,b=a+2,c=b+3);

alert(c);

alert(d);都为6

Javascript中的运算符和Java大致相同。

只是运算过程中需要注意几点：

* 1. var x = 3120/1000\*1000; x = 3120;而不是3000。
  2. var x = 2.4+3.6 ; x = 6;而不是6.0
  3. var x = “12” + 1; x = “121”; x = “12” – 1 ; x = 11;

加号对于字符串是连接符

* 1. && || 是逻辑运算符 & | 是位运算符。
  2. 也支持三元运算符
  3. 特殊运算符 8 : 返回一个操作表达式的数据类型的字符串。

var x = 3;

var y = “123”;

var z = false;

typeof(x); //number

typeof(y); //string

typeof(z); //boolean

**算数运算符**

|  |  |  |  |
| --- | --- | --- | --- |
| **运算符** | **描述** | **例子** | **结果** |
| + | 加 | x=y+2 | x=7 |
| - | 减 | x=y-2 | x=3 |
| \* | 乘 | x=y\*2 | x=10 |
| / | 除 | x=y/2 | x=2.5 |
| % | 求余数 (保留整数) | x=y%2 | x=1 |
| ++ | 累加 | x=++y | x=6 |
| -- | 递减 | x=--y | x=4 |

**赋值运算符**

|  |  |  |  |
| --- | --- | --- | --- |
| **运算符** | **例子** | **等价于** | **结果** |
| = | x=y |  | x=5 |
| += | x+=y | x=x+y | x=15 |
| -= | x-=y | x=x-y | x=5 |
| \*= | x\*=y | x=x\*y | x=50 |
| /= | x/=y | x=x/y | x=2 |
| %= | x%=y | x=x%y | x=0 |

**用于字符串的 + 运算符**

+ 运算符用于把文本值或字符串变量加起来（连接起来）。

如需把两个或多个字符串变量连接起来，请使用 + 运算符。

txt1="What a very";

txt2="nice day";

txt3=txt1+txt2;

在以上语句执行后，变量 txt3 包含的值是 "What a verynice day"。

要想在两个字符串之间增加空格，需要把空格插入一个字符串之中：

txt1="What a very ";

txt2="nice day";

txt3=txt1+txt2;

或者把空格插入表达式中：

txt1="What a very";

txt2="nice day";

txt3=txt1+" "+txt2;

在以上语句执行后，变量 txt3 包含的值是：

"What a very nice day"

**对字符串和数字进行加法运算**

请看这些例子：

x=5+5;

document.write(x);

x="5"+"5";

document.write(x);

x=5+"5";

document.write(x);

x="5"+5;

document.write(x);

规则是：

**如果把数字与字符串相加，结果将成为字符串。**

**比较运算符**

|  |  |  |
| --- | --- | --- |
| **运算符** | **描述** | **例子** |
| == | 等于 | x==8 为 false |
| === | 全等（值和类型） | x===5 为 true；x==="5" 为 false |
| != | 不等于 | x!=8 为 true |
| > | 大于 | x>8 为 false |
| < | 小于 | x<8 为 true |
| >= | 大于或等于 | x>=8 为 false |
| <= | 小于或等于 | x<=8 为 true |

**逻辑运算符**

逻辑运算符用于测定变量或值之间的逻辑。

给定 x=6 以及 y=3，下表解释了逻辑运算符：

|  |  |  |
| --- | --- | --- |
| **运算符** | **描述** | **例子** |
| && | and | (x < 10 && y > 1) 为 true |
| || | or | (x==5 || y==5) 为 false |
| ! | not | !(x==y) 为 true |