# 第五章 JavaScript上

本节所讲内容：

5.1 JavaScript介绍

5.2 JavaScript嵌入页面的方式

5.3 JavaScript变量

5.4 JavaScript运算符

5.5 JavaScript流程控制语句

## 5.1 JavaScript介绍

JavaScript是运行在浏览器端的脚本语言，JavaScript主要解决的是前端与用户交互的问题，包括使用交互与数据交互。 JavaScript是浏览器解释执行的，前端脚本语言还有JScript（微软，IE独有），ActionScript( Adobe公司，需要插件)等。

前端三大块：

1、HTML：页面结构  
 2、CSS：页面表现：元素大小、颜色、位置、隐藏或显示、部分动画效果  
 3、JavaScript：页面行为：部分动画效果、页面与用户的交互、页面功能



## 5.2 JavaScript嵌入页面的方式

### 5.2.1 嵌入页面的方式

1、行间事件（主要用于事件）

<input type="button" name="" onclick="alert('ok！');">

2、页面script标签嵌入

<script type="text/javascript">

var a = '你好！';

alert(a);

</script>

3、外部引入

<script type="text/javascript" src="js/index.js"></script>

### ****5.2.2 javascript语句与注释****

1、一条javascript语句应该以“;”结尾

<input type="button" name="" onclick="*fn*()" value="提交">  
<script>  
 var *a* = 123;  
 var *b* = 'str';  
 function *fn*() {  
 *alert*(*a*);  
 }

2、javascript注释

<script type="text/javascript">

// 单行注释

var a = 123;

/\*

多行注释

1、...

2、...

\*/

var b = 'str';

</script>

## 5.3 变量

JavaScript 是一种弱类型语言，javascript的变量类型由它的值来决定。 定义变量需要用关键字 'var'

var a = 123;

var b = 'asd';

//同时定义多个变量可以用","隔开，公用一个‘var’关键字

var c = 45,d='qwe',f='68';

5.3.1变量类型：

5种基本数据类型：

number、string、boolean、undefined、null

1. JavaScript 字符串

字符串是存储字符（比如 "for teacher"）的变量。

字符串可以是引号中的任意文本。您可以使用单引号或双引号：var carname='for teacher';

2.JavaScript Number类型

JavaScript 只有一种数字类型。数字可以带小数点，也可以不带：

var x1=34.00; //使用小数点来写

var x2=34; //不使用小数点来写

NaN，即非数值(Not a Number)是一个特殊的值，这个数值用于表示一个本来要返回数值的操作数未返回数值的情况*。*

var box = 0 / 0; //NaN

3.JavaScript 布尔类型

布尔（逻辑）只能有两个值：true 或 false。

var x=true;

var y=false;

4.JavaScript 数组

下面的代码创建名为 cars 的数组：var cars=new Array();

5.JavaScript 对象

对象由花括号分隔。在括号内部，对象的属性以名称和值对的形式 (name : value) 来定义。属性由逗号分隔：

var person={firstname:"John", lastname:"Doe", id:5566};

6.Undefined类型

Undefined类型只有一个值，即特殊的undefined。在使用var声明变量，但没有对其初始化时，这个变量的值就是undefined。

var box;

alert(box);

7.Null类型

Null类型是一个只有一个值的数据类型，即特殊的值null。它表示一个空对象引用(指针)，而typeof操作符检测null会返回object。

JavaScript描述了一组具有特定用途的关键字，一般用于控制语句的开始或结束，或者用于执行特定的操作等。关键字也是语言保留的，不能用作标识符。

ECMAScript全部关键字

|  |  |  |  |
| --- | --- | --- | --- |
| break | else | new | var |
| case | finally | return | void |
| catch | for | switch | while |
| continue | function | this | with |
| default | if | throw |  |
| delete | in | try |  |
| do | instanceof | s |  |

ECMAScript定义的全部保留字

|  |  |  |  |
| --- | --- | --- | --- |
| abstract | enum | int | short |
| boolean | export | interface | static |
| byte | extends | long | super |
| char | final | native | synchronized |
| class | float | package | throws |
| const | goto | private | transient |
| debugger | implements | protected | volatile |
| double | import | public |  |

变量是用于存储数据的"容器"。定义变量时要使用var操作符（var是关键），后面跟一个变量名（变量名是标识符）。

变量、函数、属性、函数参数命名规范：

1. 区分大小写
2. 第一个字符必须是字母、下划线（\_）或者美元符号（$）
3. 其他字符可以是字母、下划线、美元符或数字
4. 变量名称对大小写敏感（y 和 Y 是不同的变量）

5.4. JavaScript运算符

5.4.1.表达式

表达式是用于JavaScript脚本运行时进行计算的式子，可以包含常量变量运算符。ECMAScript定义了5个算术运算符，加减乘除，取余。

1. 加法

var box = 1 + 2; //等于3

var box = 1 + NaN; //NaN，只要有一个NaN就为NaN

1. 减法

var box = 100 - 70; //等于30

var box = -100 - 70 //等于-170

1. 乘法

var box = 100 \* 70; //7000

var box = 100 \* NaN; //NaN，只要有一个NaN即为NaN

1. 除法

var box = 100 / 70; //1.42....

var box = 100 / NaN; //NaN

1. 取余

var box = 10 % 3; //1，余数为1

var box = 100 % NaN; //NaN

6. 递增++和递减--

var box = 100;

++box; //把box累加一个1，相当于box = box+1

--box; //把box累减一个1，相当于box = box-1

box++;

box--;

7.前置和后置的区别

在没有赋值操作时，前置和后置是一样的。但在赋值操作时，如果递增或递减运算符前置，那么前置的运算符会先累加或累减再赋值，如果是后置运算符则先赋值再累加或累减。

var box = 100;

var age = ++box; //age值为101

var height = box++; //height值为100

PS：+号的两种作用

var a = 1+1; //纯数值 就是加法

alert(a);

var a = 1+'123abc'; //数值和字符串 连接符作用 =1123abc

alert(a);

5.4.2 运算符

1.比较运算符：小于(<)、大于(>)、小于等于(<=)、大于等于(>=)、相等(==)、不等(!=)、全等(恒等)(===)、不全等(不恒等)(!==)。

2.赋值运算符：等于(=) 加等于(+=) 减等于（-=） 乘等于（\*=） 除等于（/=） 取于等于（%=）

3.JavaScript逻辑运算符

逻辑与(AND) ：&&

var box = (5 > 4) && (4 > 3) //true，两边都为true，返回true

4.逻辑或(OR)：||

var box = (9 > 7) || (7 > 8); //true，两边只要有一边是true，返回true

逻辑非(NOT)：!

5.逗号运算符

逗号运算符可以在一条语句中执行多个操作。

var box = 100, age = 20, height = 178; //多个变量声明

6.三元条件运算符

三元条件运算符其实就是后面将要学到的if语句的简写形式。

var box = 5 > 4 ? '对' : '错'; //对，5>4返回true则把'对'赋值给box，反之。

7. 运算符优先级

在一般的运算中，我们不必考虑到运算符的优先级，因为我们可以通过圆括号来解决这种问题。比如：

var box = 5 - 4 \* 8; //-27

var box = (5 - 4) \* 8; //8

但如果没有使用圆括号强制优先级，我们必须遵循以下顺序：

|  |  |
| --- | --- |
| 运算符 | 描述 |
| . [] () | 对象成员存取、数组下标、函数调用等 |
| ++ -- ~ ! delete new typeof void | 一元运算符 |
| \* / % | 乘法、除法、去模 |
| + - + | 加法、减法、字符串连接 |
| < <= > >= instanceof | 关系比较、检测类实例 |
| == != === !== | 恒等(全等) |
| & | 位与 |
| ^ | 位异或 |
| | | 位或 |
| && | 逻辑与 |
| || | 逻辑或 |
| ?: | 三元条件 |
| = x= | 赋值、运算赋值 |
| , | 多重赋值、数组元素 |

==和===的区别:

== equality 等同，=== identity 恒等。

==， 两边值类型不同的时候，要先进行类型转换，再比较。

==，不做类型转换，类型不同的一定不等。

<!DOCTYPE html>  
<html>  
<head>  
 <meta charset="utf-8">  
 <title>js 运算符</title>  
 <script>  
 *//表达式  
 //var box = 1+2;  
 //alert(box);  
  
 //算术运算符* var *ad* = 1+3; *//4* var *ae* = 2\*3; *//6* var *af* = 6/3; *//2* var *ah* = 6-4; *//2* var *ai* = 9%4; *//1  
 //alert(ai);  
 //任何计算包含了NaN 结果都是NaN* var *box* = 3;  
 *//var aj = box++; //3 先赋值 在运算  
 //var ak = box--; //3  
 //var al = ++box; //3+1 先运算 在赋值* var *am* = --*box //3-1  
 //alert(am);  
  
 //var a = 1+'123abc'; //数值和字符串 连接符作用 =1123abc  
 //alert(a);* var *a* = true; *//null 隐式转化* if(*a*===1){  
 *alert*(*a*);  
 }else{  
 *alert*('不相等');  
 }  
 *// = 赋值 == 相等 === 恒等 数据类型相同  
 /\*  
 var ad = 6;  
 var ae = 6;  
 //alert(ad+=ae);  
 //alert(ad\*=ae);  
 //alert(ad/=ae);  
 //alert(ad%=ae);  
 alert(ad-=ae);  
  
 var box = (5>4)&&(4>5); //程序并行上  
 //alert(box);  
 var box2 = (5>4)||(4>3); //程序选择上  
 //alert(box2);  
 var box3 = !NaN; //程序取反上  
 //alert(box3);  
  
 var box4=100,box5=200,box6=300;  
 alert(box5);  
   
 var ad = 2>4?'对':'错';  
 alert(ad);  
\*/* </script>  
</head>  
<body>  
  
</body>  
</html>

## 5.5 字符串处理方法

1. 字符串合并操作：“ + ”
2. parseInt() 将数字字符串转化为整数
3. parseFloat() 将数字字符串转化为小数
4. split() 把一个字符串分隔成字符串组成的数组
5. charAt() 获取字符串中的某一个字符
6. indexOf() 查找字符串是否含有某字符
7. substring() 截取字符串 用法： substring(start,end)（不包括end）
8. toUpperCase() 字符串转大写
9. toLowerCase() 字符串转小写

字符串反转：

var str = 'asdfj12jlsdkf098';

var str2 = str.split('').reverse().join('');

alert(str2);

## 5.6 类型转换

1. 直接转换 parseInt() 与 parseFloat()

alert('12'+7); //弹出127

alert( parseInt('12') + 7 ); //弹出19

alert( parseInt(5.6)); // 弹出5

alert('5.6'+2.3); // 弹出5.62.3

alert(parseFloat('5.6')+2.3); // 弹出7.8999999999999995

alert(0.1+0.2); //弹出 0.3000000000000004

alert((0.1\*100+0.2\*100)/100); //弹出0.3

alert((parseFloat('5.6')\*100+2.3\*100)/100); //弹出7.9

1. 隐式转换 “==” 和 “-”

if('3'==3)

{

alert('相等');

}

// 弹出'相等'

alert('10'-3); // 弹出7

1. NaN 和 isNaN

alert( parseInt('123abc') ); // 弹出123

alert( parseInt('abc123') ); // 弹出NaN

5.7 JavaScript数组

5.7.1 数组的概述

数组就是一组数据的集合，javascript中，数组里面的数据可以是不同类型的。



1.使用new关键字创建数组

var box = new Array(); //创建了一个数组

var box = new Array(10); //创建一个包含10个元素的数组

var box = new Array('for','web','学神'); //创建一个数组并分配好了元素

2.以上三种方法，可以省略new关键字。

var box = Array(); //省略了new关键字

3使用字面量方式创建数组

var box = []; //创建一个空的数组

var box = ['for','web','学神']; //创建包含元素的数组

4.使用索引下标来读取数组的值

alert(box[2]); //获取第三个元素

box[2] = '学生'; //修改第三个元素

box[4] = '计算机编程'; //增加第五个元素

5.7.2 操作数组中数据的方法

1、获取数组的长度：aList.length;

var aList = [1,2,3,4];

alert(aList.length); // 弹出4

2、用下标操作数组的某个数据：aList[0];

var aList = [1,2,3,4];

alert(aList[0]); // 弹出1

3、join() 将数组成员通过一个分隔符合并成字符串

var aList = [1,2,3,4];

alert(aList.join('-')); // 弹出 1-2-3-4

4、push() 和 pop() 从数组最后增加成员或删除成员

var aList = [1,2,3,4];

aList.push(5);

alert(aList); //弹出1,2,3,4,5

aList.pop();

alert(aList); // 弹出1,2,3,4

5、unshift()和 shift() 从数组前面增加成员或删除成员

var aList = [1,2,3,4];

aList.unshift(5);

alert(aList); //弹出5,1,2,3,4

aList.shift();

alert(aList); // 弹出1,2,3,4

6、reverse() 将数组反转

var aList = [1,2,3,4];

aList.reverse();

alert(aList); // 弹出4,3,2,1

7、indexOf() 返回数组中元素第一次出现的索引值

var aList = [1,2,3,4,1,3,4];

alert(aList.indexOf(1));

8、splice() 在数组中增加或删除成员

var aList = [1,2,3,4];

aList.splice(2,1,7,8,9); //从第2个元素开始，删除1个元素，然后在此位置增加'7,8,9'三个元素

alert(aList); //弹出 1,2,7,8,9,4

5.7.3多维数组 ：

多维数组指的是数组的成员也是数组的数组。

var aList = [[1,2,3],['a','b','c']];

alert(aList[0][1]); //弹出2;

创建一个稍微复杂一点的数组

var box = [

{ //第一个元素是一个对象

name : 'for',

age : 28,

run : function () {

return 'run了';

}

},

['马云','李彦宏',new Object()],//第二个元素是数组

'江苏', //第三个元素是字符串

25+25, //第四个元素是数值

new Array(1,2,3) //第五个元素是数组

];

alert(box);

总结：

5.1 JavaScript介绍

5.2 JavaScript嵌入页面的方式

5.3 变量

5.4 JavaScript运算符

5.5 字符串处理方法

5.6 类型转换

5.7 JavaScript数组