



链滴

Javascript 学习笔记 ----- (一)

作者: [Re](#)

原文链接: <https://ld246.com/article/1578973234491>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



语言介绍

JavaScript 是能够在静态页面加动态效果的一种语言，与 Java 语言语法必将相似，当时 Java 语言正，这是这个名字的由来

基本语法

JavaScript 语言与 Java 类似，每个语句以";"结束，语句块用"{.....}"表示，但是 JavaScript 不强制要在每个语句后面加";"，它会自动帮你在行尾补充;

```
var a=1; var b=2;
```

这是两个语句，每个语句以";"结束。

语句块是一组语句的集合，例如，下面的代码先做判断，如果成立，则执行代码块。

```
if(a>2){  
    x=1;  
    y=2;  
    z=3;  
}
```

注释是代码中不运行的部分，能够启动提示开发者的功能，在 JavaScript 中注释分为两种，单行注释块注释，单行注释是//代码，块注释是/* 代码*/效果如下：

```
var x=1;  
// x=2;  
/* x=a+1;  
    y=22;*/
```

“x=2” 和 “x=a+1;y=22;” 都不会在代码中起作用。

注意： JavaScript 中严格区分大小写，如果弄错了大小写，程序将会报错或者不运行。

数据类型和变量

计算机顾名思义就是可以做数学计算的机器，因此，计算机程序理所当然地可以处理各种数值。但是计算机能处理的远不止数值，还可以处理文本、图形、音频、视频、网页等各种各样的数据，不同的数据，需要定义不同的数据类型。在 JavaScript 中定义了以下几种数据类型：

Number

Java 不区分整数和浮点数，统一用 Number 表示，以下都是 Number 的合法类型：

```
123; // 整数123
0.456; // 浮点数0.456
1.2345e3; // 科学计数法表示1.2345x1000，等同于1234.5
-99; // 负数
NaN; // NaN表示Not a Number，当无法计算结果时用NaN表示
Infinity; // Infinity表示无限大，当数值超过了JavaScript的Number所能表示的最大值时，就表示为Infinity
```

Number 可以直接做四则运算，规则和数学一致

```
1 + 2; // 3
(1 + 2) * 5 / 2; // 7.5
2 / 0; // Infinity
0 / 0; // NaN
10 % 3; // 1
10.5 % 3; // 1.5
```

字符串

字符串是指用单引号' 或双引号" 括起来的任意文本，这些学过 Java 的都懂。就不讲了。

布尔值

布尔值就两个 true 和 false，要么是 true，要不是 false。可以通过布尔运算得出来：

```
2 > 1; //true
2 == 1; //false
true && false; //false
true || false; //true
!true; //false
```

&&表示“且”，只有当两者同为 true 时它返回的时 true，“||”表示或，只有两者皆为 false 时返回的结果为 false。“!”表示取反，去后面布尔的相反布尔值。

比较运算符

当我们对 Number 作比较时，通常会得到一个布尔值，它和 Java 中类似，区别在于，它能够比较任类型的值，要区分两种比较 “==” 和 “===”：

"==" 严格度不高，他会将两边格式变为一致再进行比较，很多时候，会得到比较诡异的效果。

"===" 严格度较高，他不会转换数据类型比较，如果两边数据类型不一致，直接返回 false。

对于这个设计缺陷，我们在使用时，一定要去使用 "===" 。

另外一个意外：NaN 这个特殊的 Number 与任何值都不相同，包括他自己；

唯一能判断 NaN 的方法是通过 isNaN();函数

```
NaN===NaN; //false
isNaN(NaN); //true
```

null 和 undefined

null 表示一个“空”的值，它和 0 以及空字符串 ' '不同，0 是一个数值，而空字符串 ' '是长度为 0 的字符串。而 null 表示空。

在 JavaScript 中还有一个和“null”类似的 undefined，它表示“未定义”。

JavaScript 设计者希望用 null 表示一个空值，而 undefined 表示值未定义。事实证明，这并没有什卵用，区分两者意义不大，大多数情况下，我们都应该用 null。undefined 仅仅判断函数是否传递的情况下有用。

数组

数组是一组按顺序排列的集合，集合的每个值成为元素。JavaScript 的数组可以包括任意类型元素。[]表示

```
[1,2,6,'asd',null,true]
```

上面数组由 6 各元素构成，他们之间用 , 分开。

另一种创建数组的方法是通过 Array()函数实现：

```
new Array(1,2,3);//创建了数组[1,2,3]
```

然而，处于代码的可读性考虑，强烈建议直接使用[]。

数组的访问可以通过索引访问，他和 Java 数组一样起始索引是 0；

```
var arr=[1,2,3,4,"asda",null,true];
arr[0];//1
arr[1];//2
arr[5];//null
arr[8];//undefined
```

对象

JavaScript 的对象是由键-值构成的无序组合，例如：

```
var person={
  name:'Bob';
```

```
age:20;
tags:['js','web','mobile'];
city:'Beijing',
hasCar:true;
zipcode:null;
}
```

JavaScript 对象的键都是字符串类型，值可以是任意数据类型。上述 Person 对象一共定义了六个键对，其中每个键成为对象的属性，例如：Person 的 name 属性为 'Bob'，zipcode 属性为 null。

获取一个对象的属性值，可以用

```
对象变量.属性名
person.name; //'Bob'
person.zipcode;//null
```

变量

变量的概念基本和初中代数的方程变量是一致的，知识在计算机程序中，变量不仅可以是数字，还可以是任意数据类型。

变量在 JavaScript 中就是用一个变量名表示，变量名是大小写英文字母、数字、\$ 和_的组合，且不用数字开头。变量名也不能是 JavaScript 关键字，如 if while 等。申明一个变量用 var 语句，比如：

```
var a; // 申明了变量a, 此时a的值为undefined
var $b = 1; // 申明了变量$b, 同时给$b赋值, 此时$b的值为1
var s_007 = '007'; // s_007是一个字符串
var Answer = true; // Answer是一个布尔值true
var t = null; // t的值是null
```

变量名也可以用中文，但是，请不要给自己找麻烦。

在 JavaScript 中，使用等号 = 对变量进行赋值。可以把任意数据类型赋值给变量，同一个变量可以复赋值，而且可以是不同类型的变量，但是要注意只能用 var 申明一次，例如：

```
var a = 123; // a的值是整数123
a = 'ABC'; // a变为字符串
```

这种变量本身类型不固定的语言称之为**动态语言**，与之对应的是**静态语言**。静态语言在定义变量时必须指定变量类型，如果赋值的时候类型不匹配，就会报错。例如 Java 是**静态语言**，赋值语句如下：

```
int a = 123; // a是整数类型变量, 类型用int申明
a = "ABC"; // 错误: 不能把字符串赋给整型变量
```

和**静态语言**相比，动态语言更灵活，就是这个原因。