

# 严格模式

本文将重点介绍ES5中严格模式相关的知识点，包括但不限于严格模式的介绍、写法、严格模式和非严格模式使用对比等内容。

## 严格模式简单介绍

在 **JavaScript** 中有严格模式和非严格模式两种运行环境。

### 严格模式的主要特点

- ❑ 消除Javascript语法的一些不合理、不严谨之处，减少一些怪异行为；
- ❑ 消除代码运行的一些不安全之处，保证代码运行的安全；
- ❑ 提高编译器效率，增加运行速度；
- ❑ 为未来新版本的Javascript做好铺垫。

**严格模式** 体现了Javascript更合理、更安全、更严谨的发展方向，包括IE 10在内的主流浏览器，都已经支持它。此外，同样的代码在严格模式中，可能会有不一样的运行结果甚至是无法运行。掌握严格模式相关的内容有助于更细致深入地理解Javascript，让你变成一个更好的程序员，而**实现严格模式** 只需要在脚本代码中添加上下面介绍的关键字即可。

**关键字** `"use strict";`

### 关键字具体说明

- (1) 注意区分大小写，必须全部都是小写的
- (2) 注意空格，整个字符串总共10个字符
- (3) 单引号和双引号无所谓，但是需要有引号
- (4) 必须写在作用域的最顶部，注意其具体的位置
- (5) 可以加分号，也可以不加，但是必须是一个字符串

HTML

**注意** 下面的关键字写法均是错误的

```
"USE strict";  
" use strict ";  
"("USE strict").toLowerCase();"
```

HTML

## 严格模式使用注意点

### ➡ 严格模式使用注意

○ 修正this的值

- ① 所有的变量必须使用var 关键字声明
- ② 不能使用 `delete` 关键字删除全局变量
- ③ 在对象中不允许有同名的属性
- ④ 函数的参数必须唯一(不能出现同名的参数)
- ⑤ `arguments` 对象的行为不同, 严格模式下和实参相对独立
- ⑥ 禁用了 `argument.callee` 和 `caller`函数
- ⑦ 不能在if语句中声明函数
- ⑧ 禁止使用 `eval` 和 `argument` 作为标识符
- ⑨ 禁用了 `with` 语句和八进制字面量

## 001 所有的变量都必须使用var关键字声明

```
a = 10;           //错误的演示
console.log(10);
```

## 002 不能使用delete关键字删除全局变量

```
//在非严格模式下,删除失败(静默失败)    失败了不吭声,严格模式下直接报错
var a = 10;
delete a;
console.log(a);
```

## 003 在对象中不允许有同名的属性

```
//在非严格模式下,会使用后面的属性赋值作为最终值,在严格模式下则直接报错
var obj = {
  name: "张三",
  name: "李四"
}
console.log(obj);
```

## 004 函数的参数必须唯一(不能出现同名的参数)

```
//在非严格模式下,如果函数在定义的时候,使用了多个同名的参数,则在函数内部形参的实际值为最后一个传入的实参值
//在严格模式下,直接报错
//    function func(a,a,a) {
//        console.log(a);
//        console.log(arguments);
//    }

function func(a,b,c) {
  console.log(a);
  console.log(arguments);
}
func(1,2,3);
```

## 005 arguments对象的行为不同

HTML

- (1) 严格模式下,在函数内部修改了对象的指向,对arguments的值不会产生影响
- (2) 在严格模式下,形参的值和arguments的值是相互独立的,在函数内部修改了形参的值对arguments不受影响
- (3) 在非严格模式下,修改了形参的值,arguments中的数据会跟着改变

```
//测试引用类型的值作为函数的参数
function funcName(obj) {
    console.log(obj);
    console.log(arguments[0]);

    //在函数内部修改形参的值
    obj = {age:20};
    console.log(obj);
    console.log(arguments[0]);
}

funcName({name:"张三"});
//测试基本类型数据作为函数的参数
function fun(str) {
    console.log(str);
    console.log(arguments[0]);
    str = "hello";
    console.log(arguments[0]);
}

fun("hi");
```

## 006 禁用了argument.callee和caller函数

argument.callee 是对函数自身的引用 argument.caller 是对调用函数的引用

```
var num = (function (n) {
    if (n ==1)
    {
        return 1;
    }

    return arguments.callee(n-1) + n;
})(10);

console.log(num); //55
```

## 007 不能在if语句中声明函数

```
//如果在if语句中声明函数,则会产生语法错误
if (true)
{
    console.log("_____");
    function demo() {
        console.log("呵呵呵呵");
    }
}
```

```
    ,  
    demo();  
}
```

## 008 禁止使用eval和argument作为标识符

```
var eval = "测试的字符串";  
console.log(eval);  
  
var arguments = "参数列表";  
console.log(arguments);
```

## 009 修正this的值

在严格模式下,函数this的值始终是指定的值,无论指定的是何值

```
var name = "测试的name";  
function demoTest() {  
  
    //在非严格模式下,打印出来的this为全局的对象window  
    console.log(this); //在严格模式下打印出来的this为undefined  
}  
demoTest();
```

## 010 禁用了with语句

```
var o = {name:"暂时干",age:20};  
with(o)  
{  
    name = "lisi";  
    age = 48  
}  
console.log(o);
```

## 11 禁用了八进制

```
//以0开头的数字常常引起混乱  
//var num = 023; //2*8 + 3 ==> 19  
//console.log(num); //19
```

## 严格模式书写格式和作用范围

### 严格模式书写格式

- 01 必须使用单引号或者是双引号括住字符串
- 02 必须使用小写,不能出现大写字符
- 03 必须是10个字符,不能多和也不能少
- 04 字符串后面的分号可以省略
- 05 必须写在当前作用域的最顶上

HTML

```
//"use strict";    //正确写法
//"use strict"      //正确写法    分号可以省略
//'use strict';     //正确写法    可以使用单引号
//"use strict";     //错误写法    必须是10个字符
//"use Strict";     //错误写法    所有的字符都必须小写

"use strict";
a = 10;
// "use strict";    //错误写法
b = 20;
console.log(a);
```

## 作用范围

- ① 函数的顶部(只对当前的函数有效)
- ② script标签的顶部,只对当前的标签有效,对页面中其他的script无效

```
//位置01    对func01和func02都有效
//"use strict";

function func01() {
    //位置02    对func01有效,对func02无效
    //"use strict";
    a = 10;
    console.log(a);
}

function func02() {
    //位置03    对func02有效,但对func01无效
    //"use strict";
    b = 20;
    console.log(b);
}

func01();
func02();
```

- Posted by 博客园·文顶顶 | 花田半亩
- 联系作者 简书·文顶顶 新浪微博·Coder\_文顶顶
- 原创文章，版权声明： 自由转载-非商用-非衍生-保持署名 | 文顶顶