

ES5严格模式



jCodeLife



0.137

2020.08.14 06:32:15 字数 1,412 阅读 36

[编辑文章](#)

除了正常运行模式，ECMAScript 5添加了第二种运行模式：“严格模式”（strict mode）。顾名思义，这种模式使得JavaScript在更严格的条件下运行。

为什么提出严格模式

1. 消除JavaScript语法的一些不合理、不严谨之处，减少一些怪异行为；
2. 消除代码运行的一些不安全之处，保证代码运行的安全；
3. 提高编译器效率，增加运行速度；
4. 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，许多大项目已经开始全面拥抱它。

另一方面，同样的代码，在“严格模式”中，可能会有不一样的运行结果；一些在“正常模式”下可以运行的语句，在“严格模式”下将不能运行。所以有必要知道里面具体有哪些要求！

首先，如何启用严格模式

使用 `use strict` 这条语句即可

```
1 <script>
2   "use strict";
3   console.log("这是严格模式。");
4 </script>
```

需要注意的是，严格模式的声明必须放在脚本的第一行，否则整个脚本将会以正常模式运行。

除了这种全局的严格模式，还可以声明在函数里面，即局部的严格模式，很多时候局部的严格模式还比较常见。

注意：在函数里面声明严格模式，严格模式只对此函数内的代码起作用

```
1 function strict(){
2   "use strict";
3   return "这是严格模式。";
4 }
5 function notStrict() {
6   return "这是正常模式。";
7 }
```

接下来我们一一介绍严格模式都做了哪些规范：

1. 变量必须先声明后使用

不存在暗示全局变量这种东西了

```
1 <script>
2   "use strict";
3   a = 1; //报错
4 </script>
```

推荐阅读

Vue干货小技巧——学会再也不做加班狗

阅读 1,314

超级基础却又超级容易出错的前端面试题（1）

阅读 89

学会这6个强大的CSS选择器，将真正帮你写出干净的CSS代码！

阅读 258

面试了几个前端，给爷整哭了！

阅读 29,572

vue优化页面

阅读 818

语法错误。

```
1  "use strict";
2  function f(a, a, b) { // 语法错误
3      return ;
4  }
```

3. 不能使用with

with用于改变代码块内的的作用域，将其设置为某个对象

```
1  function Lakers() {
2      this.name = "kobe bryant";
3      this.age = "28";
4      this.gender = "boy";
5  }
6  var people=new Lakers();
7  with(people){
8      var str = "姓名: " + name + "<br>"; //直接使用了people对象中的name属性
9      str += "年龄: " + age + "<br>";
10     str += "性别: " + gender;
11     document.write(str);
12 }
```

4. 不能使用arguments.callee

callee是arguments的一个属性，用于返回正在执行的函数本身的引用

```
1  function fn(num){
2      if(num<=1){
3          return 1;
4      }else{
5          return num*arguments.callee(num-1);
6      }
7  }
```

5. 不能使用fn.caller和fn.arguments获取函数调用的堆栈

caller是函数上面的一个属性，保存着调用当前的函数的函数引用，如果是全局作用于中调用当前的函数就返回null

```
1  function testFunc() {
2      'use strict';
3      console.log(testFunc.caller); //Uncaught TypeError: ...
4      console.log(testFunc.arguments); //Uncaught TypeError: ...
5  }
6  function main() {
7      testFunc();
8  }
9  main();
```

6. 新增一些保留字

ES5本质上来讲只是一个语言优化的过渡，约束了晦涩和不安全的语法，为以后的高级语法铺平道路。所以在ES5中新增了一些保留字，严格模式下，不能使用他们作为变量名或参数名，包括：

implements, interface, let, package, private, protected, public, static, yield

7. 禁止this指向全局对象

在严格模式中，this不被允许指向全局，如果运行时试图指向全局，this将会变为undefined

推荐阅读

[Vue干货小技巧——学会再也不做加班狗](#)

阅读 1,314

[超级基础却又超级容易出错的前端面试题（1）](#)

阅读 89

[学会这6个强大的CSS选择器，将真正帮你写出干净的CSS代码！](#)

阅读 258

[面试了几个前端，给爷整哭了！](#)

阅读 29,572

[vue优化页面](#)

阅读 818

8. arguments不再追踪参数变化

常规模式下，在执行函数时如果我们更改参数的值，操作结果会立即映射到arguments对象中，更改arguments对象中的值，也会立即映射到参数上！

严格模式约束了这种行为，将arguments对象与参数分离，更改只会影响自己，不会对彼此都产生影响

```
1 'use strict';
2 var fn = function(a, b, c) {
3   a = 10;
4   console.log(arguments[0]); // 1
5
6   arguments[1] = 20;
7   console.log(b);           // 2
8 };
9
10 fn(1, 2, 3);
```

9. eval不会在它的外层作用域引入变量(强制为eval创建新作用域)

常规模式下，使用eval函数可能会影响当前作用域或全局作用域，给程序的运行结果带来不确定性，严格模式为JavaScript程序创建了第三种作用域：eval作用域。eval函数中的字符串只能在eval作用域内运行，所以严格模式下执行eval函数不会对当前作用域产生作用

```
1 'use strict';
2 eval("var a = 1;");
3 console.log(a); //Uncaught ReferenceError: a is not defined
4
5 //or
6
7 eval("'use strict'; var b = 3;");
8 console.log(b); //Uncaught ReferenceError: b is not defined
```

10. eval和arguments不能被重新赋值

常规模式下的JavaScript的随意性较大，eval和arguments可以有很多稀奇古怪的用法，程序虽然可以运行，但这给代码的可读性、可维护性也带来了一些问题。所以严格模式禁止对eval和arguments做非法操作

11. 禁止使用前缀数字零：o表示八进制数，否则报错

非严格模式下：

```
1 var num1=070;           //八进制的56
2 var num2=079;           //无效的八进制数值，解析为79
3 var num3=08;            //无效的八进制数值，解析为8
```

另外，ES6已经支持新的语法标准，八进制以0o来表示，这样一来就与16进制的0x形成统一的语法格式：

```
1 'use strict';
2 var a = 017; //Uncaught SyntaxError: ...
3 var b = 0o17; //ES6 Octal syntax: 8 + 7 = 15
```

12. 禁止对象属性重名

非严格模式重复的属性，后定义的会覆盖先定义的。严格模式下会报错

13. 函数必须声明在整个脚本或函数层面

推荐阅读

Vue干货小技巧——学会再也不做加班狗

阅读 1,314

超级基础却又超级容易出错的前端面试题（1）

阅读 89

学会这6个强大的CSS选择器，将真正帮你写出干净的CSS代码！

阅读 258

面试了几个前端，给爷整哭了！

阅读 29,572

vue优化页面

阅读 818

```
3 function doSomething() {
4   console.log('1');
5 }
6 } else {
7   function doSomething() {
8     console.log('2');
9   }
10 }
11
12 for (var i = 0; i < 5; i++) {
13   function doSomething() {
14     console.log('3');
15   }
16 }
17
18 doSomething(); //Uncaught ReferenceError: doSomething is not defined
```

14. 不能删除变量delete prop，会报错，只能删除属性delete global[prop]

这点视乎非严格模式也是这样

参考文献：

[ES5规范之严格模式](#)

[严格模式（Strict mode）--w3cschool](#)

[附录C: ECMAScript 的严格模式--w3cschool](#)

[ES5 规范之严格模式详解](#)

[Javascript 严格模式use strict详解](#)

[ES5严格模式](#)

[ES5严格模式--腾讯课堂渡一教育](#)

[JavaScript中with语句](#)

[callee和caller](#)

[在JavaScript中定义八进制数和十六进制数以及toString方法的一些用法](#)



3人点赞 >



笔记 (1)



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



jCodeLife 书山有路勤为径，学海无涯苦作舟

总资产14 (约0.99元) 共写了12.3W字 获得231个赞 共23个粉丝



写下你的评论...

全部评论 0

只看作者

关闭评论

按时间倒序 按时间正序

推荐阅读

Vue干货小技巧——学会再也不做加班狗

阅读 1,314

超级基础却又超级容易出错的前端面试题（1）

阅读 89

学会这6个强大的CSS选择器，将真正帮你写出干净的CSS代码！

阅读 258

面试了几个前端，给爷整哭了！

阅读 29,572

vue优化页面

阅读 818

写下你的评论...

评论 0

赞 3

...



推荐阅读

Vue干货小技巧——学会再也不做加班狗

阅读 1,314

超级基础却又超级容易出错的前端面试题（1）

阅读 89

学会这6个强大的CSS选择器，将真正帮你写出干净的CSS代码！

阅读 258

面试了几个前端，给爷整哭了！

阅读 29,572

vue优化页面

阅读 818



写下你的评论...

评论0

赞3

