`use strict` 的作用

作者：Vanessa
原文链接：https://ld246.com/article/1550891475480
来源网站：链滴

2019-02-23

回答

在你的 JavaScript 源文件开头包含 'use strict' 可以启用严格模式，该模式强制开启了更为严格的 JavaScript 代码解析和错误处理。他被认为是一种很好的实践，为开发者提供了以下便捷之处：

- 通过抛出错误来消除了一些原有的静默错误，让调试更加容易。
- 修复了一些导致 JavaScript 引擎难以执行优化的缺陷：有时候，相同的代码，严格模式可以比非严模式下运行得更快。
- 提高 JavaScript 的安全性，如私有变量的保护等。

```
"use strict"
function fun() {
  return arguments.callee; // Uncaught TypeError: 'caller', 'callee', and 'arguments' properties
ay not be accessed on strict mode functions or the arguments objects for calls to them
}
```

- 简化 eval() 和 arguments，如 arguments 不会随参数的变化而变化等。

```
"use strict"
function f(a){
  a = 42;
  return [a, arguments[0]];
}
var pair = f(17);
console.log(pair) // [42, 17]，非严格模式下输出为 [42, 42]
```

- 防止意外的全局变量。

```
"use strict"
mistypedVaraible = 17; // Uncaught ReferenceError: mistypedVaraible is not defined
```

- 禁止重新定义。

```
"use strict";
undefined = 5; // Uncaught TypeError: Cannot assign to read only property 'undefined' of obj
ct '#<Window>'
```

加分回答

- 使用 delete 时会抛出错误

```
"use strict";
delete mistypedVariable; // Uncaught SyntaxError: Delete of an unqualified identifier in strict
mode.
```

- 强制消除 this 的引用，默认为 undefined。

```
"use strict";
name = "atatus";
```

原文链接：`use strict` 的作用

```
function testFunction() {
    console.log(this.name);
}
// 非严格模式下输出 atatus
testFunction(); // Uncaught TypeError: Cannot read property 'name' of undefined
```

- 禁用了在 ECMAScript 未来版本中可能会定义的一些语法。

返回总目录

30 秒面试系列一