

严格模式

2020年03月18日, 星期三 14:53

1.严格模式

严格模式是采用具有限制性 JavaScript 变体的一种方式

严格模式可以应用到整个脚本或个别函数中

2.开启严格模式

整个脚本文件开启严格模式，需要在所有语句之前放一个特定语句 "use strict";

某个函数开启严格模式，得把 "use strict"; 声明放在函数体所有语句之前

3.严格模式

严格模式对 JavaScript 的语法和行为，都做了一些改变：

- 普通模式中，如果一个变量没有声明就赋值，默认是全局变量

严格模式下，禁止这种用法，变量必须先声明再使用

```
> function fun1() {  
    // "use strict";  
    a = 2;  
}  
fun1();  
↵ undefined  
> a  
↵ 2
```

```
> function fun1() {  
    "use strict";  
    a = 2;  
}  
fun1();  
✖ ▶ Uncaught ReferenceError: a is not defined  
   at fun1 (<anonymous>:3:11)  
   at <anonymous>:5:7
```

- 普通模式中，在全局作用域函数中的 this 指向 window 对象

严格模式下，全局作用域中函数中的 this 是 undefined

```
> function fun2() {  
    // "use strict";  
    console.log(this);  
}  
fun2();  
▶ Window {parent: Window, opener: null, top: Window, length: 0, frames: Window, ...}  
↵ undefined  
> function fun2() {  
    "use strict";  
    console.log(this);  
}  
fun2();  
undefined  
↵ undefined
```

- 普通模式中，构造函数不加 new 也可以调用，this 指向全局对象

严格模式下，构造函数不加 new 调用，this 报错

```
> function Fun3(name) { // 构造函数定义时，首字母大写  
    "use strict";  
    this.name = name;  
}
```

```
> function Fun3(name) { //构造函数定义时, 首字母大写
    "use strict";
    this.name = name;
}
Fun3();
```

```
✖ ▶ Uncaught TypeError: Cannot set property 'name' of undefined
    at Fun3 (<anonymous>:3:23)
    at <anonymous>:5:9
```

- 普通模式中, 函数允许参数重名
严格模式下, 函数参数不允许重名

```
▼ Arguments(2) ⓘ
  0: 1
  1: 5
  length: 2
  ▶ callee: f fun4(a, a)
  ▶ Symbol(Symbol.iterator): f values()
  ▶ __proto__: Object
```

```
< undefined
```

```
> function fun4(a, a) {
    "use strict";
    a = 5;
    console.log(a);
    console.dir(arguments);
}
fun4(1, 2);
```

```
✖ Uncaught SyntaxError: Duplicate parameter name not allowed in this context
```