

Object.prototype.toString()

🌐 翻译正在进行中。

⚠️ 非标准

该特性是非标准的，请尽量不要在生产环境中使用它！

toString() 方法返回一个表示对象源代码的字符串。

语法

```
Object.toString(); obj.toString()
```

返回值

一个表示对象的源代码的字符串。

描述

toString() 方法返回以下值：

- 对于内置的 **Object** 对象，**toString** 返回了下面的字符串，表示源码没法获取：

```
1 | function Object() {  
2 |     [native code]  
3 | }
```

- 对于 **Object** 的实例，**toString()** 会返回该实例源代码的字符串表示。

在调试时,你可以通过 **toString()** 来查看一个对象的内容。










重写toString()方法

允许对象重写 **toString()** 方法。例如：

```
1 function Person(name) {
2     this.name = name;
3 }
4
5 Person.prototype.toString = function Person_toSource() {
6     return "new Person(" + uneval(this.name) + ")";
7 };
8
9 alert(new Person("Joe").toSource()); // ---> new Person("Joe")
```

内置的toSource方法

每个JavaScript核心类型都有它自己的 `toSource()` 方法.这些对象是:

- `Array.prototype.toSource()`  — `Array` object.
- `Boolean.prototype.toSource()`  — `Boolean` object.
- `Date.prototype.toSource()`  — `Date` object.
- `Function.prototype.toSource()`  — `Function` object.
- `Number.prototype.toSource()`  — `Number` object.
- `RegExp.prototype.toSource()`  — `RegExp` object.
- `SIMD.%type%.prototype.toSource()`  — `SIMD` objects.
- `String.prototype.toSource()`  — `String` object.
- `Symbol.prototype.toSource()`  — `Symbol` object.
- `Math.toSource()` — Returns the String "Math".

Limitations on cyclical objects

In the case of objects that contain references to themselves, e.g. a cyclically linked list or a tree that can be traversed both ways, `toSource()` will not recreate the self-reference, as of Firefox 24. For example:

```
1 var obj1 = {};
2 var obj2 = { a: obj1 };
3 obj1.b = obj2;
4
5 console.log('Cyclical: ' + (obj1.b.a == obj1));
6
7 var objSource = obj1.toSource(); // returns "({b:{a:{}}})"
8
```

```
9 | obj1 = eval(objSource);
10 |
11 | console.log('Cyclical: ' + (obj1.b.a == obj1));
```

If a cyclical structure is employed and `toString()` is needed, the object must provide an override to `toString()`, either using a reference to a constructor or providing an anonymous function.

示例

使用 `toString()`

下面的代码定义了一个 `Dog` 对象类型还创建了一个 `Dog` 类型的对象实例 `theDog` :

```
1 | function Dog(name, breed, color, sex) {
2 |     this.name = name;
3 |     this.breed = breed;
4 |     this.color = color;
5 |     this.sex = sex;
6 | }
7 |
8 | theDog = new Dog("Gabby", "Lab", "chocolate", "girl");
```

在 `theDog` 上调用 `toString` 方法会显示出能定义该对象的源码:

```
1 | theDog.toString();
2 | // returns ({name:"Gabby", breed:"Lab", color:"chocolate", sex:"female"})
```

特性

不属于任何标准的一部分。在JavaScript1.3中实现。

浏览器兼容

Desktop

Mobile

Feature	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari
---------	--------	------	---------	-------------------	-------	--------

Basic Support	No	No	(Yes)	No	No	No
---------------	----	----	-------	----	----	----

相关链接

- [Object.prototype.toString\(\)](#)