

# Object.prototype.toString()

**toString()** 方法返回一个表示该对象的字符串。

## 语法

```
object.toString()
```

## 返回值

表示该对象的字符串。

## 描述

每个对象都有一个 **toString()** 方法，当该对象被表示为一个文本值时，或者一个对象以预期的字符串方式引用时自动调用。默认情况下，**toString()** 方法被每个 **Object** 对象继承。如果此方法在自定义对象中未被覆盖，**toString()** 返回 "[object type]"，其中 **type** 是对象的类型。以下代码说明了这一点：

```
var o = new Object();  
o.toString(); // returns [object Object]
```

❏ **注意：**从JavaScript1.8.5开始 **toString()** 调用 **null** 返回 **[object Null]**，**undefined** 返回 **[object Undefined]**，如第5版的ECMAScript和随后的Errata。请参阅[使用 toString\(\) 检测对象类型](#)。

## 示例

### 覆盖默认的 toString 方法

可以自定义一个方法来取代默认的 **toString()** 方法。该 **toString()** 方法不能传入参数并且必须返回一个字符串。自定义的 **toString()** 方法可以是任何我们需要的值，但如果它附带有关对象的信息，它将变的非常有用。

以下代码定义了 **Dog** 对象类型，并创建了一个 **Dog** 类型的 **theDog** 对象：

```
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 var theDog = new Dog("Gabby","Lab","chocolate","female");
```

如果当前的对象调用了 `toString()` 方法，它将会返回从 `Object` 继承下来的 `toString()` 方法的返回默认值：

```
theDog.toString(); // returns [object Object]
```

下面的代码中定义了一个叫做 `dogToString()` 的方法来覆盖默认的 `toString()` 方法。这个方法生成一个 `"property = value;"` 形式的字符串，该字符串包含了当前对象的 `name`，`breed`，`color` 和 `sex` 的值。

```
1 Dog.prototype.toString = function dogToString() {
2     var ret = "Dog " + this.name + " is a " + this.sex + " " + this.color + " " + this.breed;
3     return ret;
4 }
```

使用上述代码，任何时候在字符串上下文中使用 `theDog`，JavaScript 都会自动调用 `dogToString()` 方法，并且返回以下字符串：

```
"Dog Gabby is a female chocolate Lab"
```

## 使用 `toString()` 检测对象类型

可以通过 `toString()` 来获取每个对象的类型。为了每个对象都能通过 `Object.prototype.toString()` 来检测，需要以 `Function.prototype.call()` 或者 `Function.prototype.apply()` 的形式来调用，传递要检查的对象作为第一个参数，称为 `thisArg`。

```
1 var toString = Object.prototype.toString;
2
```

```
3 toString.call(new Date); // [object Date]
4 toString.call(new String); // [object String]
5 toString.call(Math); // [object Math]
6
7 //Since JavaScript 1.8.5
8 toString.call(undefined); // [object Undefined]
9 toString.call(null); // [object Null]
```

## 规范

Specification	Status	Comment
<a href="#">ECMAScript 1st Edition (ECMA-262)</a>	<b>ST</b> Standard	Initial definition. Implemented in JavaScript 1.0.
<a href="#">ECMAScript 5.1 (ECMA-262)</a> Object.prototype.toString	<b>ST</b> Standard	Call on <code>null</code> returns <code>[object Null]</code> , and <code>undefined</code> returns <code>[object Undefined]</code>
<a href="#">ECMAScript 2015 (6th Edition, ECMA-262)</a> Object.prototype.toString	<b>ST</b> Standard	
<a href="#">ECMAScript Latest Draft (ECMA-262)</a> Object.prototype.toString	<b>LS</b> Living Standard	

## 浏览器兼容

	Desktop	Mobile				
Feature	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari
Basic Support	(Yes)	(Yes)	(Yes)	(Yes)	(Yes)	(Yes)

## 相关链接

- Object.prototype.toSource()
- Object.prototype.valueOf()
- Number.prototype.toString()

