# JavaScript String 对象详解

JavaScript中的String对象，是字符串值的对象表示，主要用于文本处理。

所有主流浏览器均支持String对象。更多细节信息请参考[JavaScript String对象的属性和方法](http://www.365mini.com/page/tag/javascript-string-object)。

### 语法

String对象(或称之为String()函数)，其语法有以下两种：

1、充当String对象的**构造函数**使用，用于结合new关键字构造一个新的String对象。

new String( [ str ] )

2、当作**普通函数**使用，将其它数据类型转换为String数据类型的原始字符串。简而言之，这是一个类型转换函数。

String( [ str ] )

### 参数

|  |  |
| --- | --- |
| **参数** | **描述** |
| str | 可选/任意类型表示字符串的任意值，默认为空字符串("")。 |

如果参数value不是字符串，则调用该参数的[toString()](http://www.365mini.com/page/javascript-tostring.htm)方法，将其转为字符串。

### 返回值

1、如果String()函数被当做String对象的构造函数来使用，则以new关键字构造一个新的String对象(String类型)。该String对象表示参数所指定的字符串值。

2、如果String()函数被当作普通函数使用，则返回转换后的String数据类型的数值。

注意：大多数时候，你无需显示地通过new关键字来构造一个String对象，因为在JavaScript中，String对象和String数据类型是相通的，你可以在String数据类型的变量上直接使用String对象的所有属性和方法(相关示例见下面)。

### 示例&说明

以下是String()函数的几种常见用法示例：

// 构造新的String对象  
var strObj1 = new String( ); // ""  
var strObj2 = new String( "Hello" ); // "Hello"  
var strObj3 = new String( 5.3 ); // "5.3"  
var strObj4 = new String( new Object() ); // "[object Object]"  
  
 // 将数据转换为String数据类型(是原始字符串值，不是String对象)  
var str1 = String( );; // ""  
var str2 = String( "Hello" ); // "Hello"  
var str3 = String( 5.3 ); // "5.3"  
var str4 = String( new Object() ); // "[object Object]"

值得注意的是，String对象的数据类型为Object，我们可以通过typeof运算符进行查看，其结果是"object"。

var strObj = new String( "CodePlayer" );  
var str1 = "CodePlayer";  
var str2 = String( "CodePlayer" );  
  
document.writeln( typeof strObj ); // object 【在js中，绝大多数类型的对象使用typeof运算符，其结果都是"object"】  
document.writeln( typeof str1 ); // string  
document.writeln( typeof str2 ); // string  
  
//String对象和String数据类型的值相等，但类型不相等  
document.writeln( str1 === str2 ); // true (全等)  
document.writeln( str1 == strObj ); // true (值相等)  
document.writeln( str1 === strObj ); // false (不全等：值相等，但类型不相等)  
  
//只有对String对象使用instanceof运算符，其结果才返回true  
document.writeln( strObj instanceof String ); // true  
document.writeln( str1 instanceof String ); // false

[运行代码](http://www.365mini.com/diy.php?f=javascript-new-string-demo)(其他代码请自行复制到演示页面运行)

在JS中，String数据类型和String对象是相通的，因此String数据类型可以直接使用String对象的属性和方法。因此，大多数情况下，你无需显示地通过new关键字来创建一个新的String对象。

var strObj = new String( "CodePlayer" );  
var str = "CodePlayer";  
  
//String数据类型也可以直接调用String对象的属性和方法  
document.writeln( strObj.constructor === String ); // true  
document.writeln( strObj.constructor === str.constructor ); // true  
document.writeln( strObj.substring( 0, 4 ) ); // Code  
document.writeln( str.substring( 0, 4 ) ); //  Code  
  
// 字面值形式的String数据类型也可以直接调用String对象的属性和方法  
document.writeln( "Hello World".substring( 6, 11 ) ); // World

虽然String数据类型可以使用String对象的所有属性和方法，但是你不能为String数据类型的值添加自定义的属性或方法，也不能修改其内置的属性和方法(如果你这样做了，虽然不会报错，但是添加和修改操作并不会生效)。

var strObj = new String( "CodePlayer" );  
var str = "CodePlayer";  
  
// 添加自定义属性  
strObj.newProperty = "String对象的新属性";  
str.newProperty = "String数据类型的新属性";  
  
// 修改内置的方法  
strObj.substring = function (start, end){  
    return "修改String对象的substring()方法";  
};  
  
str.substring = function (start, end){  
    return "修改String数据类型的substring()方法";  
};  
  
document.writeln( strObj.newProperty ); // String对象的新属性  
// 添加的属性未生效  
document.writeln( str.newProperty ); // undefined  
  
document.writeln( strObj.substring(0, 4) ); // 修改String对象的substring()方法  
// 修改的方法也未生效，使用其内置的方法  
document.writeln( str.substring(0, 4) ); // Code