

TypeScript String

String 对象用于处理文本（字符串）。

语法

```
var txt = new String("string");  
或者更简单方式：  
var txt = "string";
```

String 对象属性

下表列出了 String 对象支持的属性：

序号	属性 & 描述	实例
1.	constructor 对创建该对象的函数的引用。	<pre>var str = new String("This is string"); console.log("str.constructor is:" + str.constructor)</pre> <p>输出结果：</p> <pre>str.constructor is:function String() { [native code] }</pre>
2.	length 返回字符串的长度。	<pre>var uname = new String("Hello World") console.log("Length "+uname.length) // 输出 11</pre>
3.	prototype 允许您向对象添加属性和方法。	<pre>function employee(id:number,name:string) { this.id = id this.name = name } var emp = new employee(123,"admin") employee.prototype.email="admin@runoob.com" // 添加属性 email console.log("员工号: "+emp.id) console.log("员工姓名: "+emp.name) console.log("员工邮箱: "+emp.email)</pre>

String 方法

下表列出了 String 对象支持的方法：

序号	方法 & 描述	实例
1.	<code>charAt()</code> 返回在指定位置的字符。	<pre>var str = new String("RUNOOB"); console.log("str.charAt(0) 为:" + str.charAt(0)); // R console.log("str.charAt(1) 为:" + str.charAt(1)); // U console.log("str.charAt(2) 为:" + str.charAt(2)); // N console.log("str.charAt(3) 为:" + str.charAt(3)); // O console.log("str.charAt(4) 为:" + str.charAt(4)); // O console.log("str.charAt(5) 为:" + str.charAt(5)); // B</pre>
2.	<code>charCodeAt()</code> 返回在指定的位置的字符的 Unicode 编码。	<pre>var str = new String("RUNOOB"); console.log("str.charCodeAt(0) 为:" + str.charCodeAt(0)); // 82 console.log("str.charCodeAt(1) 为:" + str.charCodeAt(1)); // 85 console.log("str.charCodeAt(2) 为:" + str.charCodeAt(2)); // 78 console.log("str.charCodeAt(3) 为:" + str.charCodeAt(3)); // 79 console.log("str.charCodeAt(4) 为:" + str.charCodeAt(4)); // 79 console.log("str.charCodeAt(5) 为:" + str.charCodeAt(5)); // 66</pre>
3.	<code>concat()</code> 连接两个或更多字符串，并返回新的字符串。	<pre>var str1 = new String("RUNOOB"); var str2 = new String("GOOGLE"); var str3 = str1.concat(str2); console.log("str1 + str2 : "+str3) // RUNOOBGGOOGLE</pre>
4.	<code>indexOf()</code> 回某个指定的字符串值在字符串中首次出现的位置。	<pre>var str1 = new String("RUNOOB"); var index = str1.indexOf("OO"); console.log("查找的字符串位置 : " + index); // 3</pre>

5.	lastIndexOf() 从后向前搜索字符串，并从起始位置（0）开始计算返回字符串最后出现的位置。	<pre>var str1 = new String("This is string one and again string"); var index = str1.lastIndexOf("string"); console.log("lastIndexOf 查找到的最后字符串位置 :" + index); // 29 index = str1.lastIndexOf("one"); console.log("lastIndexOf 查找到的最后字符串位置 :" + index); // 15</pre>
6.	localeCompare() 用本地特定的顺序来比较两个字符串。	<pre>var str1 = new String("This is beautiful string"); var index = str1.localeCompare("This is beautiful string"); console.log("localeCompare first :" + index); // 0</pre>
7.	match() 查找找到一个或多个正则表达式的匹配。	<pre>var str="The rain in SPAIN stays mainly in the plain"; var n=str.match(/ain/g); // ain,ain,ain</pre>
8.	replace() 替换与正则表达式匹配的子串	<pre>var re = /(\w+)\s(\w+)/; var str = "zara ali"; var newstr = str.replace(re, "\$2, \$1"); console.log(newstr); // ali, zara</pre>
9.	search() 检索与正则表达式相匹配的值	<pre>var re = /apples/gi; var str = "Apples are round, and apples are juicy."; if (str.search(re) == -1) { console.log("Does not contain Apples"); } else {</pre>

		<pre>console.log("Contains Apples"); }</pre>
10.	<code>slice()</code> 提取字符串的片断，并在新的字符串中返回被提取的部分。	
11.	<code>split()</code> 把字符串分割为子字符串数组。	<pre>var str = "Apples are round, and apples are juicy."; var splitted = str.split(" ", 3); console.log(splitted) // ['Apples', 'ar e', 'round,']</pre>
12.	<code>substr()</code> 从起始索引号提取字符串中指定数目的字符。	
13.	<code>substring()</code> 提取字符串中两个指定的索引号之间的字符。	<pre>var str = "RUNOOB GOOGLE TAOBAO FACEBOOK"; console.log("(1,2): " + str.substring(1, 2)); // U console.log("(0,10): " + str.substring(0, 10)); // RUNOOB GOO console.log("(5): " + str.substring(5)); // B GOOGLE TAOBAO FACEBOOK</pre>
14.	<code>toLocaleLowerCase()</code> 根据主机的语言环境把字符串转换为小写，只有几种语言（如土耳其语）具有地方特有的大小写映射。	<pre>var str = "Runoob Google"; console.log(str.toLocaleLowerCase()); // runoob google</pre>
15.	<code>toLocaleUpperCase()</code> 据主机的语言环境把字符串转换为大写，只有几种语言（如土耳其语）具有地方特有的大小写映射。	<pre>var str = "Runoob Google"; console.log(str.toLocaleUpperCase()); // RUNOOB GOOGLE</pre>
16.	<code>toLowerCase()</code> 把字符串转换为小写。	<pre>var str = "Runoob Google"; console.log(str.toLowerCase()); // runoob google</pre>

17.	toString() 返回字符串。	<pre>var str = "Runoob"; console.log(str.toString()); // Runoob</pre>
18.	toUpperCase() 把字符串转换为大写。	<pre>var str = "Runoob Google"; console.log(str.toUpperCase()); // RUNOOB GOOGLE</pre>
19.	valueOf() 返回指定字符串对象的原始值。	<pre>var str = new String("Runoob"); console.log(str.valueOf()); // Runoob</pre>

← TypeScript Number

TypeScript Array(数组) →

 点我分享笔记