



链滴

js 字符串操作函数

作者: [luomuren](#)

原文链接: <https://ld246.com/article/1647911902820>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



一、字符串函数

1、字符串转换

字符串转换是最基础的要求和工作，你可以将任何类型的数据都转换为字符串，你可以用下面三种方的任何一种：

```
var num=24; var mystr=num.toString(); //"24"
```

你同样可以这么做：

```
var num=24; var mystr=String(num); //"24"
```

或者，在简单点儿：

```
var num=24; var mystr="" + num; //"24"
```

2、字符串分割

将字符串进行拆分返回一个新的数组，JavaScript就给我们提供了一个非常方便的函数：

```
var mystr="qingchenghuwoguoxiansheng,woaishenghuo,woaiziji";  
var arr1=mystr.split(","); //[ "qingchenghuwoguoxiansheng", "woaishenghuo", "woaiziji" ];  
var arr2=mystr.split(""); //[ "q", "i", "n", "g", "c", "h", "e", "n", "g", "h", "u", "w", "o", "g", "u", "o", "x", "i", "a",  
n", "s", "h", "e", "n", "g", "w", "o", "a", "i", "s", "h", "e", "n", "g", "h", "u", "o", "w", "o", "a", "i", "z", "i", "j", "i",  
];
```

split()的第二个参数，表示返回的字符串数组的最大长度

```
var mystr="qingchenghuwoguoxiansheng,woaishenghuo,woaiziji";
```

```
var arr1=mystr.split(","); //["qingchenghuoguooxiansheng","woaishenghuo"];
var arr2=mystr.split("",8);
//[ "q","i","n","g","c","h","e","n"];
```

3、字符串替换

仅仅查找到字符串并不会是题目的停止，一般题目还经常会要求你去进行替换操作，那就继续看以下码：

```
var mystr="wozaijinxingzifuchuantihuancaozuo,zifuchuantihuan";
var replaceStr=mystr.replace("zifuchuan"," ");
//wozaijinxing tihuancaozuo,zifuchuantihuan
var replaceStr=mystr.replace(/zifuchuan/," ");
//wozaijinxing tihuancaozuo,zifuchuantihuan
var replaceStr=mystr.replace(/zifuchuan/g," ");
//wozaijinxing tihuancaozuo, tihuan
```

默认只进行第一次匹配操作的替换，想要全局替换，需要置上正则全局标识g

4、获取字符串长度

获取字符串的长度经常会用到，方法很简单：

```
var mystr="qingchenghuoguooxiansheng,woaishenghuo,woaiziji";
var arrLength=mystr.length; //47
```

5、查询子字符串

判断字符串内是否包含子串，不少开发者会使用for循环来判断，而忘记了JavaScript提供子串函数：

```
// indexOf(), 该方法对大小写敏感。
// 返回字符串中一个子串第一处出现的索引（从左到右搜索）。
// 如果没有匹配项，返回 -1 。
var mystr="Hello world!";
var index=mystr.indexOf("llo"); //2
var index1=mystr.indexOf("l"); //2
var index2=mystr.indexOf("l",3); //3
//lastIndexOf(), 该方法对大小写敏感。返回字符串中一个子串最后一处出现的索引（从右到左搜索），如果没有匹配项，返回 -1 。
var mystr="Hello world!";
var index=mystr.lastIndexOf("llo"); //2
var index1=mystr.lastIndexOf("l"); //9
var index2=mystr.lastIndexOf("l",4); //3
```

6、返回指定位置的字符或其字符编码值

查找给定位置的字符，可以使用如下函数：

```
var mystr="Hello World!";
var index=mystr.charAt(7); //o
```

同样，它的一个兄弟函数就是查找对应位置的字符编码值，如：

```
var mystr="Hello World!"; var charCode=mystr.charCodeAt(7); //111
```

7、字符串匹配

可以直接通过字符串进行匹配，也可以通过正则进行匹配，可能需要你对正则表达式有一定的了解，来看看match()函数：

```
var mystr="hi,mynameisguoxiansheng6,33iswho?";
var matchStr=mystr.match("guo"); //guo
var matchStr1=mystr.match("Guo"); //null
var regexp1=/\d+/g;
var regexp2=/guo/g;
var regexp3=/guo/;
var matchStr2=mystr.match(regexp1); //["6","33"]
var matchStr3=mystr.match(regexp2); //["guo"]
var matchStr3=mystr.match(regexp3); //["guo",index:11,input:"hi,mynameisguoxiansheng6,33i
who?"]
matchStr3.index //11
matchStr3.input //hi,mynameisguoxiansheng6,33iswho?
```

注意：

- 1.此处使用字符串直接进行匹配，被匹配的字符串内包含要匹配的字符串时，返回所要匹配的字符串。
- 2.如果使用正则匹配字符串时，如果正则表达式没有 g (全局标识)标志，返回与正则匹配相同的结果而且返回的数组拥有一个额外的 input 属性，该属性包含原始字符串。另外，还拥有一个 index 属性该属性表示匹配结果在被字符串中的索引（以0开始）。如果正则表达式包含 g 标志，则该方法返回匹配字符串的数组。

再来看看使用exec()函数：

```
var mystr="hi,mynameisguoxiansheng6,33iswho?";
var regexp1=/guo/g;
var matchStr=regexp1.exec(mystr); //["guo"]
var regexp2=/guo/;
var matchStr1=regexp2.exec(mystr); //["guo",index:11,input:"hi,mynameisguoxiansheng6,33i
who?"]
matchStr1.index //11
matchStr1.input //hi,mynameisguoxiansheng6,33iswho?
```

简单吧，仅仅是把正则和字符串换了个位置，即exec()函数是在正则上调用，传递字符串的参数。对上面两个方法，匹配的结果都是返回第一个匹配成功的字符串，如果匹配失败则返回null。

再来看一个类似的函数search()：

```
var mystr = "hi,mynameisguoxiansheng6,33iswho?";
var regexp1 = /guo/;
var matchStr = mystr.search(regexp1); //11
```

进行正则匹配查找。如果查找成功，返回字符串中匹配的索引值。否则返回 -1

8、字符串连接

可以将两个或多个字符串进行加法操作，同时可以使用JavaScript提供的concat函数：

先看加法操作进行字符串连接：

```
var mystr1="Hello"; var mystr2="world!";  
var newStr=mystr1+" "+mystr2; //Hello world!
```

是不是很简单呀，那继续看看

concat函数吧：

```
var mystr1="Hello";  
var mystr2=" world,";  
var mystr3="Hello";  
var mystr4="guoxiansheng";  
var newStr=mystr1.concat(mystr2+mystr3+" "+mystr4); //Hello world,Hello guoxiansheng
```

concat()函数可以有多个参数，传递多个字符串，拼接多个字符串。

9、字符串切割和提取

有三种可以从字符串中抽取和切割的方法：

第一种， slice()函数：

```
var mystr="hello world!";  
var sliceStr1=mystr.slice(-3); //ld!  
var sliceStr2=mystr.slice(-3,-1); //ld  
var sliceStr3=mystr.slice(3); //lo world!  
var sliceStr4=mystr.slice(3,7); //lo w
```

第二种： substring()函数：

```
var mystr="hello world!";  
var sliceStr1=mystr.substring(3); //lo world!  
var sliceStr2=mystr.substring(3,7); //lo w
```

第三种： substr()函数：

```
var mystr="hello world!";  
var sliceStr1=mystr.substr(3); //lo world!  
var sliceStr2=mystr.substr(3,7); //lo wo
```

注：

1.slice() 可以为负数，如果起始位置为负数，则从字符串最后一位向前找对应位数并且向后取结束位，如果为正整数则从前往后取起始位置到结束位置。

2.substring()只能非负整数，截取起始结束位置同slice()函数一致。

3.substr()与第一、第二种函数不同，从起始位置开始截取，结束位置为第二个参数截取的字符串最长度。

以上三种函数未填第二参数时，自动截取起始位置到字符串末尾。

10、字符串大小写转换

```
var mystr="Hello World!";  
var lowCaseStr=mystr.toLowerCase(); //hello world!  
var upCaseStr=mystr.toUpperCase(); //HELLO WORLD!
```

11、字符串去空格

trim方法用来删除字符串前后的空格

```
var mystr="  hello world  ";  
var trimStr=mystr.trim(); //hello world
```

PS:若对字符串学习还是不够透彻，请查阅W3C :http://www.w3school.com.cn/jsref/jsref_obj_string.asp https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/String