

# 字符串 String

## String 类型

### 创建String对象

```
var stringObject = new String("hello");
```

### 属性

length 字符串包含的字符个数。

### 连接方法

concat(stringX, stringX, ..., stringX) 连接字符串。调用后原字符串不发生改变。

### 提取子串方法

slice(start, end(可选)) 提取字符串从 start 到 end （不包括该字符）的字符，返回一个新字符串。若参数是负值，则加上字符串长度。

substr(start, length(可选)) 在字符串中提取从 start 开始的长度为 length 的字符，返回一个新字符串。若第一个参数是负值，则加上字符串长度，若第二个参数是负值，则转换成0。

substring(start, end(可选)) 提取字符串中从 start 到 end （不包括该字符）的字符，返回一个新字符串。若参数是负值，则转换成0。

调用后原字符串不发生改变。

### 替换/删除子串方法

replace(要替换/删除的字符串/正则表达式, 新字符串) 替换/删除子串。

- replace(要被替换的字符串/正则表达式, 新字符串) 用一些字符替换另一些指定字符，或与正则表达式匹配的字符，返回一个新字符串。
- replace(要删除的字符串/正则表达式, "") 删除指定字符或与正则表达式匹配的字符，返回一个新字符串。

若第一个参数是字符串，则只替换或删除第一个匹配项。若需要替换或删除所有匹配项，用正则表达式的全局匹配 (g) 模式。

调用后原字符串不发生改变。

### 索引查找方法

charAt(index) 返回指定位置的字符。

方括号加索引访问法

### 指定字符查找方法

indexOf(要查找的子串, 查找的起点位置(可选)) 从字符串开头向后查找，返回第一个匹配项的索引。如果没有找到，则返回 -1。

lastIndexOf(要查找的子串, 查找的起点位置(可选)) 从字符串末尾向前查找，返回第一个匹配项的索引。如果没有找到，则返回 -1。

### 模式匹配查找方法

match(字符串/正则表达式) 在字符串内检索指定的值或正则表达式匹配，把找到的结果放入一个数组。本质上与调用 Regexp 的 exec() 相同。

search(正则表达式/Regexp对象) 在字符串内检索指定的值或正则表达式匹配，返回第一个匹配项的索引。

### 切割方法/转换数组方法

[split\(字符串/正则表达式, 返回的数组的最大长度\(可选\)\)](#) 把一个字符串分割成若干个字符串, 放入一个数组中。调用后原字符串不发生改变。

参数:

字符串/正则表达式 以该参数为分割标志分割数组。如果第一个参数是空字符串(""), 那么字符串中每个字符会被分割。

### 大小写转换方法

[toLowerCase\(\)](#) 把字符串转换为小写。

[toUpperCase\(\)](#) 把字符串转换为大写。

[toLocaleLowerCase\(\)](#) 按照本地方式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。

[toLocaleUpperCase\(\)](#) 按照本地方式把字符串转换为大写。

调用后原字符串发生改变。

### 去除空格方法

[trim\(\)](#) ECMAScript5 创建一个字符串副本, 删除前置及后缀的所有空格。调用后原字符串不发生改变。

### 比较方法

[localeCompare\(target\)](#) 比较两个字符串。如果字符串在字母表中排在 target 之前, 则返回一个负数(一般是-1)。如果字符串在字母表中排在 target 之后, 则该方法返回一个正数(一般是1)。如果两个字符串相等, 则返回 0。