

RegExp.prototype.exec()



[查看所有贡献者](#)

`exec()` 方法在一个指定字符串中执行一个搜索匹配。返回一个结果数组或 `null`。

如果你正在执行一个匹配只是为了找到 `true` 或 `false`，可以使用 `RegExp.test()` 方法，或者 `String.search()` 方法。

语法

```
regexObj.exec(str)
```

参数

str

要匹配正则表达式的字符串。

返回值

如果成功匹配，`exec()` 方法返回一个数组并更新正则表达式对象的属性。返回的数组具有匹配的文本作为第一项，然后匹配的每个捕获括号的一个项包含捕获的文本。

如果匹配失败，`exec()` 方法返回 `null`。

描述

思考以下示例：

```
/*
    Match "quick brown" followed by "jumps",
    ignoring characters in between
    Remember "brown" and "jumps"
    Ignore case
*/
let regex = /quick\s(brown).+?(jumps)/ig;
// let regex = new RegExp(/quick\s(brown).+?(jumps)/,"ig");
// let regex = new RegExp("quick\s(brown).+?(jumps)","ig");
let result = regex.exec('The Quick Brown Fox Jumps Over The Lazy Dog');
```

下面的表格展示这个脚本的返回值：

对象	属性/索引	描述	例子
result	[0]	匹配的全部字符串	Quick Brown Fox Jumps
	[1], ...[n]]	括号中的分组捕获	[1] = Brown [2] = Jumps
	index	匹配到的字符位于原始字符串的基于0的索引值	4
	input	原始字符串	The Quick Brown Fox Jumps Over The Lazy Dog
re	lastIndex	下一次匹配开始的位置	25
	ignoreCase	是否使用了'i'标记使正则匹配忽略大小写	true
	global	是否使用了'g'标记来进行全局的匹配.	true
	multiline	是否使用了'm'标记使正则工作在多行模式（也就是，^和\$可以匹配字符串中每一行的开始和结束（行是由\n或\r分割的），而不只是整个输入字符串的最开始和最末尾处。）	false
	source	正则模式的字符串	quick\s(brown).+?(jumps)

示例

例子1：查找所有匹配

当正则表达式使用 "g" 标志时，可以多次执行 `exec` 方法来查找同一个字符串中的成功匹配。当你这样做时，查找将从正则表达式的 `lastIndex` 属性指定的位置开始。（`test()` 也会更新 `lastIndex` 属性）。例如，你使用下面的脚本：

```
var myRe = /ab*/g;
var str = 'abbccdefabh';
var myArray;
while ((myArray = myRe.exec(str)) !== null) {
    var msg = 'Found ' + myArray[0] + '. ';
    msg += 'Next match starts at ' + myRe.lastIndex;
    console.log(msg);
}
```

脚本运行结果如下：

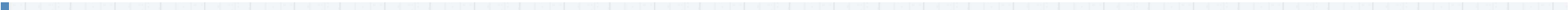
```
1 | Found abb. Next match starts at 3
2 | Found ab. Next match starts at 9
```

注意：不要把正则表达式字面量（或者正则表达式构造器）放在 `while` 条件表达式里。由于每次迭代时 `lastIndex` 的属性都被重置，如果匹配，将会造成一个死循环。

例子2：结合 RegExp 字面量使用 `exec()`

你也可以直接使用 `exec()` 而不是创建一个 `RegExp` 对象：

```
var matches = /(hello \S+)/.exec('This is a hello world!');
console.log(matches[1]);
```



运行上面的代码，会弹出一个内容为 "hello world!" 的对话框。

规范

Specification	Status	Comment
ECMAScript 3rd Edition. Implemented in JavaScript 1.2	Standard	Initial definition. Implemented in JavaScript 1.2.
↗ ECMAScript 5.1 (ECMA-262) RegExp.exec	<div>ST</div> Standard	
↗ ECMAScript 2015 (6th Edition, ECMA-262) RegExp.exec	<div>ST</div> Standard	

浏览器兼容性

	Desktop	Mobile				
Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari	
Basic support	(Yes)	(Yes)	(Yes)	(Yes)	(Yes)	

相关链接

- [Regular Expressions](#) chapter in the [JavaScript Guide](#)
- [RegExp](#)