

正则表达式的相关方法

正则表达式的相关方法

- ◆ 正则表达式可以“打点”调用哪些方法呢？

方法	简介
test()	测试某字符串是否匹配正则表达式，返回布尔值
exec()	根据正则表达式，在字符串中进行查找，返回结果数组或者null

test()方法

- ◆ 正则表达式的test()方法用来测试某字符串是否匹配此正则表达式，它返回true或false

```
/^[a-z]{3}[a-ln-z]$/ .test('abcd')
```

```
true
```

exec()方法

- ◆ exec()方法功能是：在一个指定字符串中执行一个搜索匹配查找，返回一个结果数组或null

```
var str = 'abc123def456ghi789';  
var regexp = /\d+/;  
var result1 = regexp.exec(str);
```

```
▶ ["123", index: 3, input: "abc123def456ghi789", groups: undefined]
```

exec()方法的逐条遍历

- ◆ exec()方法最有趣的是，有“g”修饰符的正则表达式将自动成为“有状态”的，这意味着可以对单个字符串中的多次匹配结果进行逐条的遍历

exec()方法的逐条遍历

```
var str = 'abc123def456ghi789';  
var regexp = /\d+/g; // 有g修饰符  
var result1 = regexp.exec(str);  
var result2 = regexp.exec(str);  
var result3 = regexp.exec(str);  
var result4 = regexp.exec(str);  
console.log(result1);  
console.log(result2);  
console.log(result3);  
console.log(result4);
```

会进行逐条遍历

```
▶ ["123", index: 3, input: "abc123def456ghi789", groups: undefined]  
▶ ["456", index: 9, input: "abc123def456ghi789", groups: undefined]  
▶ ["789", index: 15, input: "abc123def456ghi789", groups: undefined]  
null
```

exec()方法的逐条遍历

```
var str = 'abc123def456ghi789';  
var regexp = /\d+/g;    // 有g修饰符  
var result;  
while (result = regexp.exec(str)) {  
    console.log(result);  
}
```

会进行逐条遍历

```
▶ ["123", index: 3, input: "abc123def456ghi789", groups: undefined]  
▶ ["456", index: 9, input: "abc123def456ghi789", groups: undefined]  
▶ ["789", index: 15, input: "abc123def456ghi789", groups: undefined]  
null
```

字符串的相关方法

字符串的相关方法

◆ 字符串有哪些方法可以使用正则表达式呢？

方法	简介
search()	在字符串中根据正则表达式进行查找匹配，返回首次匹配到的位置索引，测试不到则返回-1
match()	在字符串中根据正则表达式进行查找匹配，返回一个数组，找不到则返回null
replace()	使用替换字符串替换掉匹配到的子字符串，可以使用正则表达式
split()	分隔字符串为数组，可以使用正则表达式

谢谢！