



黑马程序员™
www.itheima.com

传智播客旗下
高端IT教育品牌

正则表达式



目录 Contents

- ◆ 正则表达式概述
- ◆ 正则表达式在 JavaScript 中的使用
- ◆ 正则表达式中的特殊字符
- ◆ 正则表达式中的替换

■ 1. 正则表达式概述

1.1 什么是正则表达式

正则表达式 (Regular Expression) 是用于匹配字符串中字符组合的模式。在 JavaScript 中，正则表达式也是对象。

正则表通常被用来检索、替换那些符合某个模式（规则）的文本，例如验证表单：用户名表单只能输入英文字母、数字或者下划线，昵称输入框中可以输入中文(匹配)。此外，正则表达式还常用于过滤掉页面内容中的一些敏感词(替换)，或从字符串中获取我们想要的特定部分(提取)等。

其他语言也会使用正则表达式，本阶段我们主要是利用 JavaScript 正则表达式完成表单验证。

■ 1. 正则表达式概述

1.2 正则表达式的特点

1. 灵活性、逻辑性和功能性非常的强。
2. 可以迅速地用极简单的方式达到字符串的复杂控制。
3. 对于刚接触的人来说，比较晦涩难懂。比如：`^\w+([-+.] \w+)*@\w+([-.] \w+)*\.\w+([-.] \w+)*$`
4. 实际开发,一般都是直接复制写好的正则表达式. 但是要求会使用正则表达式并且根据实际情况修改正则表达式. 比如用户名: `/^[a-z0-9_-]{3,16}$/`

目录 Contents

- ◆ 正则表达式概述
- ◆ 正则表达式在 JavaScript 中的使用
- ◆ 正则表达式中的特殊字符
- ◆ 正则表达式中的替换

2. 正则表达式在 JavaScript 中的使用

2.1 创建正则表达式

在 JavaScript 中，可以通过两种方式创建一个正则表达式。

1. 通过调用 RegExp 对象的构造函数创建

```
var 变量名 = new RegExp(/表达式/);
```

2. 通过字面量创建

```
var 变量名 = /表达式/;
```

// 注释中间放表达式就是正则字面量

2. 正则表达式在 JavaScript 中的使用

2.2 测试正则表达式 test

test() 正则对象方法，用于检测字符串是否符合该规则，该对象会返回 true 或 false，其参数是测试字符串。

```
regexObj.test(str)
```

1. regexObj 是写的正则表达式
2. str 我们要测试的文本
3. 就是检测str文本是否符合我们写的正则表达式规范.

目录 Contents

- ◆ 正则表达式概述
- ◆ 正则表达式在 JavaScript 中的使用
- ◆ 正则表达式中的特殊字符
- ◆ 正则表达式中的替换



3. 正则表达式中的特殊字符

3.1 正则表达式的组成

一个正则表达式可以由简单的字符构成，比如 `/abc/`，也可以是简单和特殊字符的组合，比如 `/ab*c/`。其中特殊字符也被称为元字符，在正则表达式中是具有特殊意义的专用符号，如 `^`、`$`、`+` 等。

特殊字符非常多，可以参考：

- MDN : https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Guide/Regular_Expressions
- jQuery 手册：正则表达式部分
- 正则测试工具: <http://tool.oschina.net/regex>

这里我们把元字符划分几类学习。

3. 正则表达式中的特殊字符

3.2 边界符

正则表达式中的边界符（位置符）用来提示字符所处的位置，主要有两个字符。

边界符	说明
^	表示匹配行首的文本(以谁开始)
\$	表示匹配行尾的文本(以谁结束)

如果 ^ 和 \$ 在一起，表示必须是精确匹配。



3. 正则表达式中的特殊字符

3.3 字符类

字符类表示有一系列字符可供选择，只要匹配其中一个就可以了。所有可供选择的字符都放在方括号内。

1. [] 方括号

```
/[abc]/.test('andy')    // true
```

后面的字符串只要包含 abc 中任意一个字符，都返回 true。

3. 正则表达式中的特殊字符

3.3 字符类

字符类表示有一系列字符可供选择，只要匹配其中一个就可以了。所有可供选择的字符都放在方括号内。

2. [-] 方括号内部 范围符-

```
/^[a-z]$/.test('c')    // true
```

方括号内部加上 - 表示范围，这里表示 a 到 z 26个英文字母都可以。



3. 正则表达式中的特殊字符

3.3 字符类

字符类表示有一系列字符可供选择，只要匹配其中一个就可以了。所有可供选择的字符都放在方括号内。

3. [^] 方括号内部 取反符^

```
[^abc]/.test('andy')    // false
```

方括号内部加上 ^ 表示取反，只要包含方括号内的字符，都返回 false。

注意和边界符 ^ 区别，边界符写到方括号外面。



3. 正则表达式中的特殊字符

3.3 字符类

字符类表示有一系列字符可供选择，只要匹配其中一个就可以了。所有可供选择的字符都放在方括号内。

4. 字符组合

```
[a-z1-9]/.test('andy') // true
```

方括号内部可以使用字符组合，这里表示包含 a 到 z 的26个英文字母和 1 到 9 的数字都可以。

3. 正则表达式中的特殊字符

3.4 量词符

量词符用来设定某个模式出现的次数。

量词	说明
*	重复零次或更多次
+	重复一次或更多次
?	重复零次或一次
{n}	重复n次
{n,}	重复n次或更多次
{n,m}	重复n到m次

3. 正则表达式中的特殊字符



案例：用户名验证

功能需求:

1. 如果用户名输入合法, 则后面提示信息为: 用户名合法, 并且颜色为绿色
2. 如果用户名输入不合法, 则后面提示信息为: 用户名不符合规范, 并且颜色为绿色

3. 正则表达式中的特殊字符



案例：用户名验证分析

分析:

1. 用户名只能为英文字母,数字,下划线或者短横线组成, 并且用户名长度为 6~16位.
2. 首先准备好这种正则表达式模式 `/^[a-zA-Z0-9-]{6,16}$/`
3. 当表单失去焦点就开始验证.
4. 如果符合正则规范, 则让后面的span标签添加 right 类.
5. 如果不符合正则规范, 则让后面的span标签添加 wrong 类.

■ 3. 正则表达式中的特殊字符

3.5 括号总结

1. 大括号 量词符. 里面表示重复次数
2. 中括号 字符集合。匹配方括号中的任意字符.
3. 小括号 表示优先级

可以在线测试: <https://c.runoob.com/>



3. 正则表达式中的特殊字符

3.6 预定义类

预定义类指的是某些常见模式的简写方式。

预定义类	说明
\d	匹配0-9之间的任一数字，相当于[0-9]
\D	匹配所有0-9以外的字符，相当于 [^0-9]
\w	匹配任意的字母、数字和下划线，相当于[A-Za-z0-9_]
\W	除所有字母、数字和下划线以外的字符，相当于 [^A-Za-z0-9_]
\s	匹配空格（包括换行符、制表符、空格符等），相等于[\t\r\n\v\f]
\S	匹配非空格的字符，相当于 [^\t\r\n\v\f]

3. 正则表达式中的特殊字符



案例：表单验证

分析:

1. 手机号码: `/^1[3|4|5|7|8][0-9]{9}$/`
2. QQ: `[1-9][0-9]{4,}` (腾讯QQ号从10000开始)
3. 昵称是中文: `^[\u4e00-\u9fa5]{2,8}$`

目录 Contents

- ◆ 正则表达式概述
- ◆ 正则表达式在 JavaScript 中的使用
- ◆ 正则表达式中的特殊字符
- ◆ 正则表达式中的替换



4. 正则表达式中的替换

4.1 replace 替换

replace() 方法可以实现替换字符串操作，用来替换的参数可以是一个字符串或是一个正则表达式。

```
stringObject.replace(regexp/substr,replacement)
```

1. 第一个参数: 被替换的字符串 或者 正则表达式
2. 第二个参数: 替换为的字符串
3. 返回值是一个替换完毕的新字符串



4. 正则表达式中的替换

4.2 正则表达式参数

/表达式/[switch]

switch(也称为修饰符) 按照什么样的模式来匹配. 有三种值:

- g : 全局匹配
- i : 忽略大小写
- gi : 全局匹配 + 忽略大小写

4. 正则表达式中的替换



案例：敏感词过滤





黑马程序员

www.itheima.com

传智播客旗下高端IT教育品牌