

正则表达式的创建

正则表达式的创建

- ◆ 使用 `/内容/` 的语法形式，可以快速创建正则表达式
- ◆ 也可以使用 `new RegExp('内容')` 的形式，创建正则表达式
- ◆ 使用 `typeof` 运算符检查正则表达式的类型，结果是 `object`

元字符

元字符

◆ “元字符”是指一位指定类型的字符

元字符	功能
\d	匹配一个数字
\D	匹配一个非数字字符
\w	匹配一个单字字符（字母、数字或者下划线）
\W	匹配一个非单字字符
\s	匹配一个空白字符，包括空格、制表符和换行符
.	任意字符

元字符

◆ 开头和结尾

元字符	功能
^	匹配开头
\$	匹配结尾

元字符注意事项

◆ 如果使用new RegExp()写法，反斜杠需要多写一个

◆ 比如/^\\d\$/和new RegExp('^\\\\d\$')是相同的意思

元字符使用举例

- ◆ 某快递公司运单号形式是这样的：123-4567-890，请使用正则表达式检查某字符串是否符合此格式
- ◆ 某产品的验证秘钥形式是这样的：□□□-□□□□-□□□，其中□表示字母数字或者下划线，请使用正则表达式检查某字符串是否符合此格式

字符的转义

- ◆ 在特殊字符之前的反斜杠\表示下一个字符不是特殊字符，应该按照字面理解

`/^.$/`



检查字符串是不是任意字符

`/^\..$/`



检查字符串是不是一个点

`/^\\.$/`



检查字符串是不是一个反斜杠

字符的转义

- ◆ 不管一个符号有没有特殊意义，都可以在其之前加上一个\`\`以确保它表达的是这个符号本身
- ◆ 举例：某产品批号形式为：`123.45^67#89`，请使用正则表达式检查某字符串是否符合此格式

方括号表示法

方括号表示法

- ◆ 使用方括号，比如[xyz]，可以创建一个字符集合，表示匹配方括号中的任意字符
- ◆ 比如某学校的学号规定：第1位是一个字母，b表示本科生，y表示研究生，后面是7位数字，用正则表示为：

`/^[by]\d{7}$/`

方括号表示法

- ◆ 可以使用短横-来指定一个字符范围，^表示否定

元字符	等价的方括号表示法
<code>\d</code>	<code>[0-9]</code>
<code>\D</code>	<code>[^0-9]</code>
<code>\w</code>	<code>[A-Za-z0-9_]</code>
<code>\W</code>	<code>[^A-Za-z0-9_]</code>

方括号表示法小题目

- ◆ 题目1：请验证某字符串是否是5位字母，大小写均可
- ◆ 题目2：请验证某字符串是否是5位，且仅有小写字母、点构成
- ◆ 题目3：请验证某字符串是否是4位小写字母，且最后一位不能是m字母

量词

量词

量词	意义
*	匹配前一个表达式0次或多次。等价于{0,}
+	匹配前面一个表达式1次或者多次。等价于{1,}
?	匹配前面一个表达式0次或者1次。等价于{0,1}
{n}	n是一个正整数，匹配了前面一个字符刚好出现了n次
{n,}	n是一个正整数，匹配前一个字符至少出现了n次
{n,m}	n和m都是整数。匹配前面的字符至少n次，最多m次

量词小题目

- ◆ 题目1：请验证字符串是否符合手机号码的规则：11位数字，并且肯定以1开头
- ◆ 题目2：请验证某字符串是否是这样的：以字母开头，中间是任意位数字（最少1位）构成，并以字母结尾
- ◆ 题目3：请验证某字符串是否符合网址规则：以www.开头，中间是任意位的字符（字母数字下划线），最后以.com结尾，也可以以.com.cn结尾

修饰符

修饰符

- ◆ 修饰符也叫作标志（ flags ），用于使用正则表达式实现高级搜索

修饰符	意义
i	不区分大小写搜索
g	全局搜索

修饰符

◆ 修饰符的使用：

```
var re = /m/gi;
```

在正则表达式的第二个/后面写修饰符，可以多个连写

```
var re = new RegExp('m', 'gi');
```

new RegExp()形式的正则表达式，以字符串形式写在第二个参数

谢谢！