正则表达式

是有字母数字和一些特殊符号组成的：：：用来描述字符模式的对象

功能：

1. 模式匹配
2. 文本检索
3. 替换

要使用正则表达式，第一步，创建属于你们的正则表达式

创建方式有两种

1. 通过构造函数来创建（显式创建）

Var reg = new RegExp(“表达式”，[修饰符])；

1. 通过字面量（直接量）方式来创建（隐式创建） ----推荐，常用

Var reg = /表达式/[修饰符]；

两种当时含义有所区别：1是创建的是实例对象，2创建的是单个对象；

正则表达式的使用方法：

1. reg.test(想要检测的变量)；
   1. 功能：用于检测是否与正则匹配；
   2. 返回：布尔值，如果匹配，则返回true，不匹配则返回false;
2. reg.exec(想要检测的变量)；
   1. 功能：用于检测是否与正则匹配；
   2. 返回：如果匹配，则返回数组，不匹配则返回null；
      1. 说明：

支持两个属性：index:索引；input内容

正则表达式：

方括号：

[abc] 查找方括号之间的任何字符

[^abc] 查找任何不在方括号内的字符

[0-9]查找0-9之间的数字

[a-z]查找从a到z之间的小写字母

[A-Z] 查找A到Z之间的大写字母

元字符：（字符类）

. 查找单个字符，除了换号和行结束符

\w 查找字母数字下划线

\W 查找非字母数字下划线

\d 所有数字

\D 所有非数字

\s 查找空白字符

\S 查找非空白字符

\n 查找换行符

\r 查找回车符

\t 查找制表符

限定符：

^xxx 以xxx开始

xx$ 以xx结束

重复：设置匹配次数

{n} 匹配前一项n次

{n,m} 匹配前一项n到m次

{n,} 匹配前一项n到多次

\* 匹配前一项0到多次

? 匹配前一项0或者1次

+ 匹配前一项1到多次

联系，写一个手机号码的正则表达式

转义字符：将特殊符号恢复成它原本的字面意思

通过：\这个符号来进行转义

需要转义的符号： [ ] { } ( ) . ? \* + ^ $ \ /