# 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 avaScript中，正则表达式也是对象。

●通常用来查找、替换那些符合正则表达式的文本，许多语言都支持正则表达式。

## 使用方法

1、创建规则

使用方法：const 变量名 = /表达式|表达式/ // “|”有或的意思

2、test()方法 用来查看正则表达式与指定的字符串是否匹配

使用方法：规则变量.test(被检测的字符串变量) //返回真：true、假：false

3、exec()方法 在一个指定字符串中执行一个搜索匹配

使用方法：规则变量.exec(被检测的字符串变量) //返回真：一个数组、假：null

## 元字符（特殊字符）

●是一些具有特殊含义的字符，可以极大提高了灵活性和

强大的匹配功能。

●比如英文26个英文字母，我们使用元字符 [a-z] 简介和灵活。

1. 边界符 (表示位置，开头和结尾，必须用什么开头，用什么结尾)

^ 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)

注意：如果^和$在一起，表示必须是精确匹配

1. 量词 (表示重复次数)

\* 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

1. 字符类 (比如\d 表示 0~9)

. 任意一个字符 //不能识别换行符

[] 匹配字符集合// [abc]，匹配a或b或c，[a-z0-9A-Z]，匹配数字和大小字母其中一个

[^...] 取反字符集合// [^abc]，匹配除了a或b或c

1. 预定义字符类

\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]

## 修饰符

使用方法：const 变量名 = /表达式/修饰符

修饰符：

i：是单词 ignore 的缩写，正则匹配时字母不区分大小写 /^Wgs$/i.test(‘wgs’)

g：是单词 global的缩写，匹配所有满足正则表达式的结果 ，替换字符串时常用

## 替换字符串

replace 替换：字符串.replace(/正则表达式/，’替换的文本’); //加上修饰符g替换全部