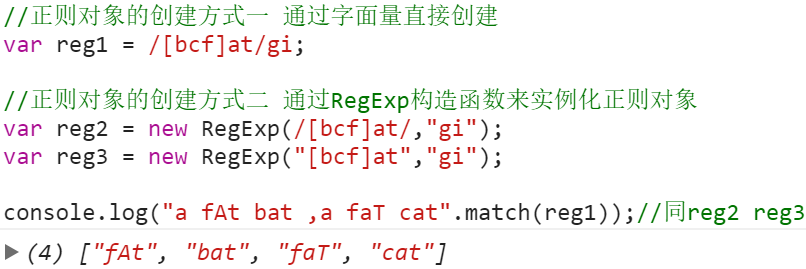
**第二十章 RegExp正则表达式**

**什么是正则表达式**

- 正则表达式是对字符串和特殊字符操作的一种逻辑公式，是对字符串执行模式匹配的工具  
- 正则表达式通常被用来检索、替换那些符合某个模式(规则)的文本  
- JS中正则表达式是一个描述字符模式的对象，可以通过字面量、RegExp构造器来生成

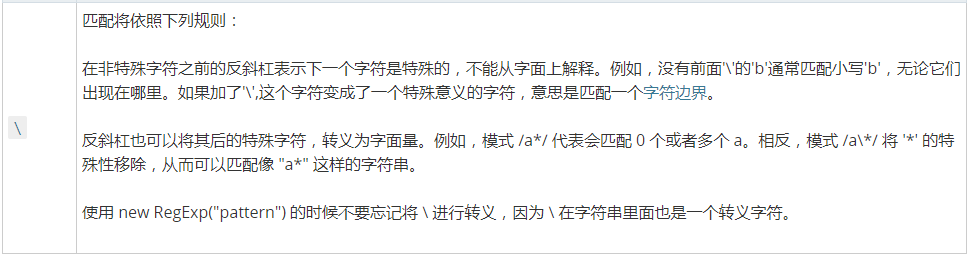


**正则表达式的元字符 及\相关字符**

元字符：1 3 5 a b c 等

转义字符：\t 、\v、\n、\r、\0、\f、\cX

与\相关的预定义特殊字符：\d、\D、\w、\W、\s、\S、\b 、\B（注意大小写的含义）



**正则表达式特殊字符 一 （字符类）**  
[ ]代表字符类，如[abc]代表abc中的任意一个字符，可以配合范围符号-如[a-c3-9]  
[^ ]代表字符类取反，如[^abc]代表非abc中的任意一个字符

一个 - 代表范围，如[a-z] 、[a-z0-9A-Z]  
一个 . 代表一个除了回车和换行符之外的所有字符 等效于[^\r\n]，（注意与\*的区别和含义）

**正则表达式特殊字符 二 （边界相关）**   
边界字符 ^ $ \b \B （注意^代表的意义与在[ ]中代表的意义不同）

**正则表达式特殊字符 三 （量词）**   
？出现0次或1次（最多1次） +出现1次或多次（至少1次） \*出现0次或多次（任意次）  
{n} 出现n次 {n,m} 出现n到m次 {n,}出现至少n次

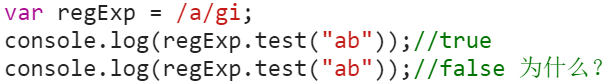
**正则表达式贪婪模式与非贪婪模式**思考：“12345678”.replace(/\d{3,6}/,'X');返回多少？  
默认为贪婪模式（即尽可能多的匹配），在量词后加？可设置为非贪婪模式

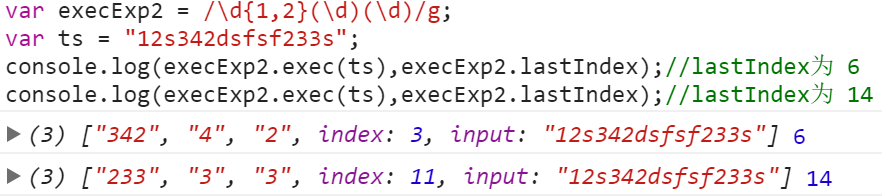
**正则表达式的分组**思考：匹配Name连续出现3次的正则，/Name{3}/，这样可以么？  
使用小括号来进行分组 ，如：/(Name){3}/g  
或 |、分组中的或 |

**正则表达式对象的属性（源自RegExp.prototype）**

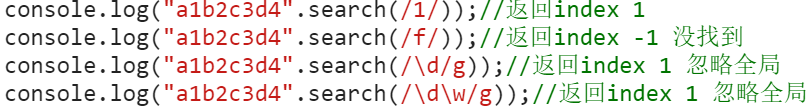
- global 默认为false  
- ignore case 默认为false  
- multiline 默认为false  
- lastIndex 表示当前匹配内容的最后一个字符的下一个位置  
- sourse 正则表达式文本字符串

**正则表达式RegExp原型方法（test）**

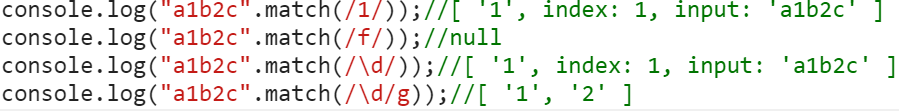


**正则表达式RegExp原型方法（exec），可获得详细信息**

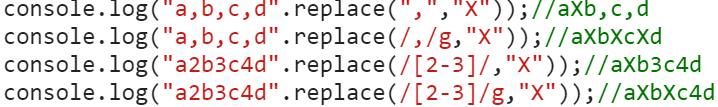
**字符串与正则相关的原型方法（String.prototype.search）**



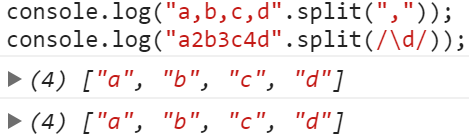
**字符串与正则相关的原型方法（String.prototype.match）**



**字符串与正则相关的原型方法（String.prototype.replace）**



**字符串与正则相关的原型方法（String.prototype.split）**



**常用正则表达式**/\w+((-\w+)|(\.\w+))\*\@[A-Za-z0-9]+((\.|-)[A-Za-z0-9]+)\*\.[A-Za-z0-9]+/ （邮箱）  
/^[A-Za-z0-9\_-]+$/ （密码）  
/((?:(?:25[0-5]|2[0-4]\d|[01]?\d?\d)\.){3}(?:25[0-5]|2[0-4]\d|[01]?\d?\d))/ （IP地址）  
/(.\*)\.(rar|zip|7zip|tgz)$/ （压缩格式）  
/(.\*)\.(jpg|bmp|gif|ico|pcx|jpeg|tif|png|raw|tga)$/ （图片判断）  
/^#[a-fA-F0-9]{6}$/ （颜色值）  
/^[A-Za-z0-9\_\-\u4e00-\u9fa5]+$/ （用户名）  
/0?(13|14|15|18)[0-9]{9}/ （手机号）  
/^[A-Za-z0-9\_()（）\-\u4e00-\u9fa5]+$/ （公司名称）

集成开发环境中的应用