定义正则表达式有两种方法，第一种通过"/正则表达式/修饰符"这种形式直接写出来，第二种通过“new RegExp('正则表达式'，'修饰符)'”创建一个RegExp对象。其中修饰符为可选项，有三个取值g：全局匹配；i：不区分大小写；m：多行匹配

\ 做为转意，即通常在"\"后面的字符不按原来意义解释，如/b/匹配字符"b"，当b前面加了反斜杆后/\b/，转意为匹配一个单词的边界。   
-或-   
对正则表达式功能字符的还原，如"\*"匹配它前面元字符0次或多次，/a\*/将匹配a,aa,aaa，加了"\"后，/a\\*/将只匹配"a\*"。   
  
^ 匹配一个输入或一行的开头，/^a/匹配"an A"，而不匹配"An a"   
$ 匹配一个输入或一行的结尾，/a$/匹配"An a"，而不匹配"an A"   
\* 匹配前面元字符0次或多次，/ba\*/将匹配b,ba,baa,baaa   
+ 匹配前面元字符1次或多次，/ba\*/将匹配ba,baa,baaa   
? 匹配前面元字符0次或1次，/ba\*/将匹配b,ba   
(x) 匹配x保存x在名为$1...$9的变量中   
x|y 匹配x或y   
{n} 精确匹配n次   
{n,} 匹配n次以上   
{n,m} 匹配n-m次   
[xyz] 字符集(character set)，匹配这个集合中的任一一个字符(或元字符)   
[^xyz] 不匹配这个集合中的任何一个字符   
[\b] 匹配一个退格符   
\b 匹配一个单词的边界   
\B 匹配一个单词的非边界   
\cX 这儿，X是一个控制符，/\cM/匹配Ctrl-M   
\d 匹配一个字数字符，/\d/ = /[0-9]/   
\D 匹配一个非字数字符，/\D/ = /[^0-9]/   
\n 匹配一个换行符   
\r 匹配一个回车符   
\s 匹配一个空白字符，包括\n,\r,\f,\t,\v等   
\S 匹配一个非空白字符，等于/[^\n\f\r\t\v]/   
\t 匹配一个制表符   
\v 匹配一个重直制表符   
\w 匹配一个可以组成单词的字符(alphanumeric，这是我的意译，含数字)，包括下划线，如[\w]匹配"$5.98"中的5，等于[a-zA-Z0-9]   
\W 匹配一个不可以组成单词的字符，如[\W]匹配"$5.98"中的$，等于[^a-zA-Z0-9]。