正则表达式:规则表达式;

就是跟字符串打交道;描述了一套规则;

regular Expression:

1322142324@qq.com √

zhao@126.com √

@asd.com ×

asdsad@.com ×

邮箱的组成：

用户名[数字,字母,中横线,字母不区分大小写,不能包含汉字]@邮箱类型[不能包含汉字].后缀名[.com,.cc,.org,.net,.me]

字符串：

var str = '';

str.indexOf();

str.lastIndexOf();

str.split();

str.charAt();

str.substring();

str.toLowerCase();

str.toUpperCase();

正则:

性能极高;

方便;

正则的写法:

var reg = new RegExp('要查找的字符串');找不到返回-1;从左往右找一次;区分大小写;

str.search(reg);

new RegExp('要查找的字符串','g');

i:ignore:忽略;

g:global:全局,全部;

match();匹配;

正则的简写：

var reg = //;

\:转义;

\n:换行;

\d: d: digital:数字;

量词:{}

{n}:正好n个;/\d{3}/:正好3个数字;

{n,m}最少n个,最多到m个;

{1,}最少1个,最多不限;---> +;

{0,}可以没有,有的话也不限制个数;----> \*;

/ab+/--->abbbbbb;

/(ab)+/--->abababab;

+:n多个;

||:或者;

|:正则里面的或者;

str.replace(reg,'\*\*\*');

str.replace(reg,function(s){ });

^:行首;放到[]里面就代表排除;

$:行尾;

i:ignore:忽略

g:global:全局

m:multi-line:多行

关于转义:

\d:digital:数字

\n:换行;

\s:space:空格；

\w:word:单词;

\D:非数字;

\W:非单词;

\S:非空格;

. :任意东西;

关于[]:

[abc]-->a|b|c;

[1-39]-->1到3,或者9

[a-z]--->a-z 26个小写字母;

[A-Z]--->A-Z 26个大写字母;

[0-9]--->0-9的数字--->\d;

[^0-9]--->非数字;--->\D

[^a-z]--->非小写字母;

[^A-Z]---->非大写字母;