**正则表达式**

用于定义一些字符串的规则

计算机可以根据正则表达式，来检查一个字符串是否符合规则，

或者将字符串中符合规则的内容提取出来

**创建正则表达式对象**

**语法：**

var 变量 = new RegExp("正则表达式","匹配模式");

使用typeof检查正则对象会返回object

**eg.**

var reg = new RegExp("a" ) ;//这个正则表达式可以来检查一个字符串中是否含有a

在构造函数中可以传递一个匹配模式来作为第二个参数，

可以是 i 忽略大小写

g 全局匹配模式

m 换行

**eg.**

var reg =new RegExp("ab","i");

**量词**

通过量词设置一个内容出现的次数 量词只对它前边的一个内容起作用

**{n}正好出现n次**

var reg = /a{3}/;

**{m,n}出现m到n次**

reg= /ab{1,3}c/;

**{m,}出现m次以上**

reg= /ab{1,}c/;

**n+表示包含一个n字符串 相当于{1,}**

reg = /ab+c/;

**n\*表示包含0个也行 相当于{0,}**

reg = /ab\*c/;

**n? 表示包含0个或1个 相当于{0,1}**

reg = /ab?c/;

**检查一个字符串是否以a开头**

reg=/^a/;

**检查一个字符串是否以a结尾**

reg=/a$/;

**如果在正则表达式中同时使用^$则要求字符串必须完全符合正则表达式**

reg=/^a$/ ;

**以a开头或者以a结尾**

**eg:**

/\*

\* 创建一个手机号的正则表达式

\* 第一位是1

\* 第二位是3-9

\* 剩下9位任意

\*/

var phonestr = "13123456789";

var phoneReg = /^1[3-9][0-9]{9}$/;

console.log(phoneReg.test(phonestr));

**元字符**

任意字符串 一个.匹配一个任意的字符

检查一个字符串中是否含有.

因为\.表示任意字符

在正则表达式中使用\作为转义字符

\. 表示.

\\ 表示\ 在字符串中需要转义字符\\来表示斜杠

\* 注意,使用构造函数时，由于它的参数是一个字符串，而\是字符串中转义字符，

\* 如果要表示\则需要使用\\来表示

\*/

var reg = /\./ ;//等价于下面这个

reg =new RegExp("\\.");//字面量里的\.是匹配.

//用构造函数创建时想告诉构造函数这是\.的话 因为是字符串所以要输\\.

console.log(reg.test("abc"));

console.log(reg);

**\w**

-任意字母数字下划线 [A-Za-z0-9\_]

**\W**

-除了字母数字下划线 [^A-Za-z0-9\_]

**\d**

-任意数字

**\D**

-除了数字

**\s**

-空格

**\S**

-除了空格

**\b**

-单词边界 字符串两边都有空格才能匹配到 \b一次使用两个 一前一后

var str = "asabcasd";

var reg = /\babc\b/;

console.log(reg.test(str));

**\B**

-除了单词边界 字符串两边不能有空格才能匹配到 一个空格都不行 也是成对出现

/\*

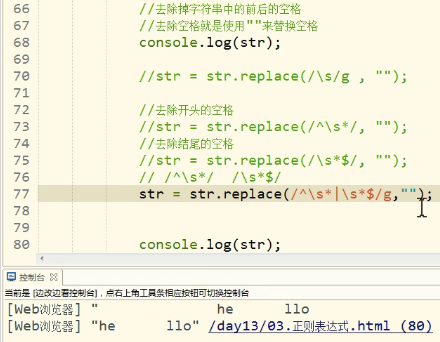
\*创建一个正则表达式检查一个字符串中是否含有单词child

\*/

reg =/\bchild\b/ ;

console.log(reg.test( "hello children"));

**用户名去除开头和结尾的空格**



**电子邮件的正则**



**正则表达式的方法:**

**test()**

——使用这个方法可以用来检查一个字符串是否符合正则表达式的规则，

如果符合则返回true，否则返回false

**exec()**

——条件满足返回数组，不满足返回null

**使用字面量来创建正则表达式**

**语法:**

var 变量 = /正则表达式/匹配模式

**eg.**

reg = /a/i;//等同于用构l造函数创建正则表达式

var reg =new RegExp("a" , ""i");

**区别：**

使用字面量的方式创建更加简单

使用构造函数创建更加灵活

创建一个正则表达式，检查一个字符串中是否有字母

——使用│表示或者的意思 reg = / a | b | c | d | e | f | g / ;

——[ ]里的内容也是或的关系

一个中括号表示一个占位字符

中括号内是取值范围

[ab] == a | b reg = /a[bde]c/

[a-z]任意小写字母 reg= /[A-z]/

[A-Z]任意大写字母

[A-z]任意字母

[^ab] 这个字符不能是a或b

[0-9] 任意数字

^[A-Za-z0-9\_]表示只能有数字字母下划线

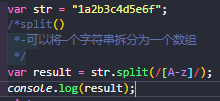
@1000phone.com$ 只能以这个字符串结尾

**支持正则表达式的String方法**

**split()**

——方法中可以传递一个正则表达式作为参数，这样方法将会根据正则表达式去拆分字符串

——即使不指定全局匹配也会全都拆开



**search()**

——可以搜索字符串中是否含有指定内容

——如果搜索到指定内容，则会返回第一次出现的索引，如果没有搜素到返回-1

——它可以接受一个正则表达式作为参数，然后会根据正则表达式去检索字符串

——即使设置了全局匹配，也只会查找第一个

截图.png

**match()**

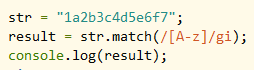
——可以根据正则表达式，从一个字符串中将符合亲件的内容提取出来

——默认情况下我们的match只会找到第一个符合要求的内容，找到以后就停止检索

可以设置正则表达式为全局匹配模式,这样就会匹配到所有的内容

可以为一个正则表达式设置多个匹配模式，且顺序无所谓

——matc()会将匹配到的内容封装到一个数组中返回，即使只查询到一个结果



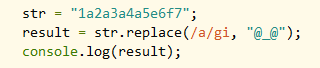
**replace()**

——可以将字符串的指定内容替换成新的内容

——参数：

1.被替换的内容，可以接受一个正则表达式作为参数

2.新的内容



只替换每一行开头的数字为#号



**一、校验数字的表达式**

数字：^[0-9]\*$

n位的数字：^\d{n}$

至少n位的数字：^\d{n,}$

m-n位的数字：^\d{m,n}$

零和非零开头的数字：^(0|[1-9][0-9]\*)$

非零开头的最多带两位小数的数字：^([1-9][0-9]\*)+(.[0-9]{1,2})?$

带1-2位小数的正数或负数：^(\-)?\d+(\.\d{1,2})?$

正数、负数、和小数：^(\-|\+)?\d+(\.\d+)?$

有两位小数的正实数：^[0-9]+(.[0-9]{2})?$

有1~3位小数的正实数：^[0-9]+(.[0-9]{1,3})?$

非零的正整数：^[1-9]\d\*$ 或 ^([1-9][0-9]\*){1,3}$ 或 ^\+?[1-9][0-9]\*$

非零的负整数：^\-[1-9][]0-9″\*$ 或 ^-[1-9]\d\*$

非负整数：^\d+$ 或 ^[1-9]\d\*|0$

非正整数：^-[1-9]\d\*|0$ 或 ^((-\d+)|(0+))$

非负浮点数：^\d+(\.\d+)?$ 或 ^[1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*|0?\.0+|0$

非正浮点数：^((-\d+(\.\d+)?)|(0+(\.0+)?))$ 或 ^(-([1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*))|0?\.0+|0$

正浮点数：^[1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*$ 或 ^(([0-9]+\.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*\.[0-9]+)|([0-9]\*[1-9][0-9]\*))$

负浮点数：^-([1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*)$ 或 ^(-(([0-9]+\.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*\.[0-9]+)|([0-9]\*[1-9][0-9]\*)))$

浮点数：^(-?\d+)(\.\d+)?$ 或 ^-?([1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*|0?\.0+|0)$

**二、校验字符的表达式**

汉字：^[\u4e00-\u9fa5]{0,}$

英文和数字：^[A-Za-z0-9]+$ 或 ^[A-Za-z0-9]{4,40}$

长度为3-20的所有字符：^.{3,20}$

由26个英文字母组成的字符串：^[A-Za-z]+$

由26个大写英文字母组成的字符串：^[A-Z]+$

由26个小写英文字母组成的字符串：^[a-z]+$

由数字和26个英文字母组成的字符串：^[A-Za-z0-9]+$

由数字、26个英文字母或者下划线组成的字符串：^\w+$ 或 ^\w{3,20}$

中文、英文、数字包括下划线：^[\u4E00-\u9FA5A-Za-z0-9\_]+$

中文、英文、数字但不包括下划线等符号：^[\u4E00-\u9FA5A-Za-z0-9]+$ 或 ^[\u4E00-\u9FA5A-Za-z0-9]{2,20}$

可以输入含有^%&',;=?$\”等字符：[^%&',;=?$\x22]+

禁止输入含有~的字符：[^~\x22]+

**三、特殊需求表达式**

Email地址：^\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*$

域名：[a-zA-Z0-9][-a-zA-Z0-9]{0,62}(/.[a-zA-Z0-9][-a-zA-Z0-9]{0,62})+/.?

InternetURL：[a-zA-z]+://[^\s]\* 或 ^http://([\w-]+\.)+[\w-]+(/[\w-./?%&=]\*)?$

手机号码：^(13[0-9]|14[5|7]|15[0|1|2|3|5|6|7|8|9]|18[0|1|2|3|5|6|7|8|9])\d{8}$

电话号码(“XXX-XXXXXXX”、”XXXX-XXXXXXXX”、”XXX-XXXXXXX”、”XXX-XXXXXXXX”、”XXXXXXX”和”XXXXXXXX)：^($$\d{3,4}-)|\d{3.4}-)?\d{7,8}$

国内电话号码(0511-4405222、021-87888822)：\d{3}-\d{8}|\d{4}-\d{7}

身份证号(15位、18位数字)：^\d{15}|\d{18}$

短身份证号码(数字、字母x结尾)：^([0-9]){7,18}(x|X)?$ 或 ^\d{8,18}|[0-9x]{8,18}|[0-9X]{8,18}?$

帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)：^[a-zA-Z][a-zA-Z0-9\_]{4,15}$

密码(以字母开头，长度在6~18之间，只能包含字母、数字和下划线)：^[a-zA-Z]\w{5,17}$

强密码(必须包含大小写字母和数字的组合，不能使用特殊字符，长度在8-10之间)：^(?=.\*\d)(?=.\*[a-z])(?=.\*[A-Z]).{8,10}$

日期格式：^\d{4}-\d{1,2}-\d{1,2}

一年的12个月(01～09和1～12)：^(0?[1-9]|1[0-2])$

一个月的31天(01～09和1～31)：^((0?[1-9])|((1|2)[0-9])|30|31)$

**钱的输入格式：**

有四种钱的表示形式我们可以接受:”10000.00″ 和 “10,000.00″, 和没有 “分” 的 “10000″ 和 “10,000″：^[1-9][0-9]\*$

这表示任意一个不以0开头的数字，但是，这也意味着一个字符”0″不通过，所以我们采用下面的形式：^(0|[1-9][0-9]\*)$

一个0或者一个不以0开头的数字.我们还可以允许开头有一个负号：^(0|-?[1-9][0-9]\*)$

这表示一个0或者一个可能为负的开头不为0的数字.让用户以0开头好了.把负号的也去掉，因为钱总不能是负的吧.下面我们要加的是说明可能的小数部分：^[0-9]+(.[0-9]+)?$

必须说明的是，小数点后面至少应该有1位数，所以”10.”是不通过的，但是 “10″ 和 “10.2″ 是通过的：^[0-9]+(.[0-9]{2})?$

这样我们规定小数点后面必须有两位，如果你认为太苛刻了，可以这样：^[0-9]+(.[0-9]{1,2})?$

这样就允许用户只写一位小数。下面我们该考虑数字中的逗号了，我们可以这样：^[0-9]{1,3}(,[0-9]{3})\*(.[0-9]{1,2})?$

1到3个数字，后面跟着任意个 逗号+3个数字，逗号成为可选，而不是必须：^([0-9]+|[0-9]{1,3}(,[0-9]{3})\*)(.[0-9]{1,2})?$

备注：这就是最终结果了，别忘了”+”可以用”\*”替代。如果你觉得空字符串也可以接受的话(奇怪，为什么?)最后，别忘了在用函数时去掉去掉那个反斜杠，一般的错误都在这里

xml文件：^([a-zA-Z]+-?)+[a-zA-Z0-9]+\\.[x|X][m|M][l|L]$

中文字符的正则表达式：[\u4e00-\u9fa5]

双字节字符：[^\x00-\xff] (包括汉字在内，可以用来计算字符串的长度(一个双字节字符长度计2，ASCII字符计1))

空白行的正则表达式：\n\s\*\r (可以用来删除空白行)

HTML标记的正则表达式：<(\S\*?)[^>]\*>.\*?</\1>|<.\*? /> (网上流传的版本太糟糕，上面这个也仅仅能部分，对于复杂的嵌套标记依旧无能为力)

首尾空白字符的正则表达式：^\s\*|\s\*$或(^\s\*)|(\s\*$) (可以用来删除行首行尾的空白字符(包括空格、制表符、换页符等等)，非常有用的表达式)

腾讯QQ号：[1-9][0-9]{4,} (腾讯QQ号从10000开始)

中国邮政编码：[1-9]\d{5}(?!\d) (中国邮政编码为6位数字)

IP地址：\d+\.\d+\.\d+\.\d+ (提取IP地址时有用)

IP地址：((?:(?:25[0-5]|2[0-4]\\d|[01]?\\d?\\d)\\.){3}(?:25[0-5]|2[0-4]\\d|[01]?\\d?\\d))

**四、字母,数字,下划线或者数字的正则表达式**

1.由数字、26个英文字母或者下划线组成的字符串:

^[0-9a-zA-Z\_]{1,}$

2.非负整数（正整数 + 0 ）:

^/d+$

3. 正整数:

^[0-9]\*[1-9][0-9]\*$

4.非正整数（负整数 + 0）：

^((-/d+)|(0+))$

5. 负整数 :

^-[0-9]\*[1-9][0-9]\*$

6.整数:

^-?/d+$

7.非负浮点数（正浮点数 + 0）:

^/d+(/./d+)?$

8.正浮点数 :

^(([0-9]+/.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*/.[0-9]+)|([0-9]\*[1-9][0-9]\*))$

9. 非正浮点数（负浮点数 + 0）:

^((-/d+(/./d+)?)|(0+(/.0+)?))$

10.负浮点数 :

^(-(([0-9]+/.[0-9]\*[1-9][0-9]\*)|([0-9]\*[1-9][0-9]\*/.[0-9]+)|([0-9]\*[1-9][0-9]\*)))$

11. 浮点数 :

^(-?/d+)(/./d+)?$

12.由26个英文字母组成的字符串 :

^[A-Za-z]+$

13. 由26个英文字母的大写组成的字符串 :

^[A-Z]+$

14.由26个英文字母的小写组成的字符串 :

^[a-z]+$

15. 由数字和26个英文字母组成的字符串 :

^[A-Za-z0-9]+$

16.由数字、26个英文字母或者下划线组成的字符串 :

^/w+$

17.email地址 :

^[/w-]+(/.[/w-]+)\*@[/w-]+(/.[/w-]+)+$

18.url:

^[a-zA-z]+://(/w+(-/w+)\*)(/.(/w+(-/w+)\*))\*(/?/S\*)?$

19. 年-月-日:

/^(d{2}|d{4})-((0([1-9]{1}))|(1[1|2]))-(([0-2]([1-9]{1}))|(3[0|1]))$/

20.月/日/年:

/^((0([1-9]{1}))|(1[1|2]))/(([0-2]([1-9]{1}))|(3[0|1]))/(d{2}|d{4})$/

21.Emil:

^([w-.]+)@(([[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.)|(([w-]+.)+))([a-zA-Z]{2,4}|[0-9]{1,3})(]?)$

22. 电话号码:

(d+-)?(d{4}-?d{7}|d{3}-?d{8}|^d{7,8})(-d+)?

23.IP地址:

^(d{1,2}|1dd|2[0-4]d|25[0-5]).(d{1,2}|1dd|2[0-4]d|25[0-5]).(d{1,2}|1dd|2[0-4]d|25[0-5]).(d{1,2}|1dd|2[0-4]d|25[0-5])$

24. 匹配中文字符的正则表达式：

[/u4e00-/u9fa5]

25.匹配双字节字符(包括汉字在内)：

[^/x00-/xff]

26. 匹配空行的正则表达式：

/n[/s| ]\*/r

27.匹配HTML标记的正则表达式：

/<(.\*)>.\*<///1>|<(.\*) //>/

28.匹配首尾空格的正则表达式：

(^/s\*)|(/s\*$)

29.匹配Email地址的正则表达式：

/w+([-+.]/w+)\*@/w+([-.]/w+)\*/./w+([-.]/w+)\*

30. 匹配网址URL的正则表达式：

^[a-zA-z]+://(//w+(-//w+)\*)(//.(//w+(-//w+)\*))\*(//?//S\*)?$

31. 匹配帐号是否合法(字母开头，允许5-16字节，允许字母数字下划线)：

^[a-zA-Z][a-zA-Z0-9\_]{4,15}$

32. 匹配国内电话号码：

(/d{3}-|/d{4}-)?(/d{8}|/d{7})?

33.匹配腾讯QQ号：

^[1-9]\*[1-9][0-9]\*$

34. 只能输入数字：

^[0-9]\*$

35.只能输入n位的数字：

^/d{n}$

36.只能输入至少n位的数字：

^/d{n,}$

37.只能输入m~n位的数字：

^/d{m,n}$

38.只能输入零和非零开头的数字：

^(0|[1-9][0-9]\*)$

39.只能输入有两位小数的正实数：

^[0-9]+(.[0-9]{2})?$

40. 只能输入有1~3位小数的正实数：

^[0-9]+(.[0-9]{1,3})?$

41.只能输入非零的正整数：

^/+?[1-9][0-9]\*$

42. 只能输入非零的负整数：

^/-[1-9][0-9]\*$

43.只能输入长度为3的字符：

^.{3}$

44. 只能输入由26个英文字母组成的字符串：

^[A-Za-z]+$

45.只能输入由26个大写英文字母组成的字符串：

^[A-Z]+$

46. 只能输入由26个小写英文字母组成的字符串：

^[a-z]+$

47.只能输入由数字和26个英文字母组成的字符串：

^[A-Za-z0-9]+$

48. 只能输入由数字和26个英文字母或者下划线组成的字符串：

^/w+$

49.验证用户密码(正确格式为： 以字母开头，长度在5~17 之间，只能包含字符、数字和下划线)

^[a-zA-Z]/w{5,17}$

50.验证是否包含有 ^%&',;=?$/"等字符：

[^%&',;=?$/x22]+

51.只能输入汉字：

^[\u4e00-\u9fa5]{0,}$

52、只含有汉字、数字、字母、下划线不能以下划线开头和结尾

^(?!\_)(?!.\*?\_$)[a-zA-Z0-9\_\u4e00-\u9fa5]+$

53、只含有汉字、数字、字母、下划线，下划线位置不限

^[a-zA-Z0-9\_\u4e00-\u9fa5]+$

54、2~4个汉字

@"^[\u4E00-\u9FA5]{2,4}$



