**正则表达式**

**一.补充**

(一)．转义字符“\”；

能在字符串里面加 ”号

var abc=“123\”123” 🡪 打印123”123

（二）．多行字符串；

（三）．字符串换行符\n；回车

（四）.回车全等于\r\n

**二.RegExp**

定义：类似与Date，

正则表达式的作用：匹配特殊字符或有特殊搭配原则的字符的最佳选择。

**两种创建方式**

1-直接量

var reg= /abc/igm 后面跟i忽视大小写/abc/i

后面跟g是全局匹配，找到所有符合的贪婪匹配原则

后面跟m是执行多行匹配，首先字符串多行，（插入:/^a/查的必须是每行开头的a）

2-new RegExp( );

var reg=new RegExp(“abc”，“i”)

(推荐直接量)

**reg.test(str);----返回true和false ，str里有没有reg规定的字符串返回布尔**

**str.match(reg);----返回能匹配的字符串的集合[ ]**

var reg =/[1234567890][1234567890][1234567890]/g;

每一个[ ]里设有一个范围，上述查询的三个字符取值的范围都是0-9，是可以按asc码排

var str="nkdcbsdb4563fcvfsh234";

str.match(reg)--🡪[“456”,”234”]

把^放在[ ]里面表示非

var reg=/[^1]/表示第一位不是1都可以

用（ | ）括起来表示或

var reg=/（1|2）/表示第一位是1或者2

**（）里面是子表达式，应用检验是否含有aaaa bbbb这种四个的连着的一样的字符串**

**var reg=/(\w)\1\1\1/===(\w)表示第一个子表达式，\1表示引用第一个子表达式里的东西**

**元字符**

\.===除了\n和\r

\w===[0-9A-z\_]

\W===除了\w

\d===[0-9]

\D===除了\d

\s===空白字符===[\t\n\r\v\f ]最后有一个空格

\S===除了\s

\b===单词边界，一个字符的头和尾，放前面就是前单词边界，放后面就是后单词边界

\B===除了\b

\0===查看NUL字符

\n===查找换行符

\f===查找换页符

\r===查找回车符

\t===查看制表符

\v===查看垂直指标符

\xxx===查看八进制xxx规定的字符

\xdd===查看十六进制数dd规定的字符

\uxxxx===查看以十六进制数xxxx规定的Unicode字符

\d\D===代表一切

**量词 （**量词变量后面加一个？就是非贪婪匹配，能取少不取多）

正则表达式原则是贪婪匹配从多到少

代表数量的词

n+===n可以出现1次到无数次

/1+/g

n\*===n可以出现0次到无数次

n?===0到1次

n{X}===X个

n{x,y}===x个到y个

n{x,}===x到正无穷个

n$===是否以n结尾的

写一个正则表达式检验一个字符首尾是否含有数字

var reg=/^\d|\d$/

首尾都含有

var reg=/^\d[\d\D]\*d$/

**正则表达式上的属性方法**

1.reg.ignoreCase ()是否写了i g m里的i返回boolean

2.reg.global() 是否写了i g m里的g返回boolean

3.reg.multiline() 是否写了i g m里的m返回boolean

4.reg.source ()返回正则表达式的内容

**5.reg.test()判断括号里的字符串是否含有正则表达式的形式返回true和false**

加了g!!!reg.exec()接着匹配，轮圈一个个转，游标会根据匹配一直在变

6.reg.lastIndex游标

不加g lastIndex一直不变一直是0

**Sring对象关于正则表达式的属性方法**

1.match检索与正则表达式相匹配的子字符串，返回第一个（无g）或所有（有g）形成数组

2.search 匹配到的位置，匹配不到返回-1

3.split 按正则表达式里的拆分

**4.replace(“a”,”b”)字符串里a换成b，不访问所有，只访问第一个**

用正则表达式就可以全换

var reg=/(\w)\1(\w)\1/g

var str=aabb

console.log=(str.replace(reg,“$2$2$1$1”))

aabb换成bbaa

其中$也是反向引用子表达式

把the-first-name转变成theFirstName

var reg=/-(\w)/g

var str=“the-first-name”

console.log(str.replace(reg,function($,$1){

return $1.toUpperCase();

}))

**正向预查（修饰）**

var str=”aaaabaa”查找b前面的a

var reg=/a(?=b)/g;

?!=非正向预查

**把10000000转化成10,000,000**

**var reg=/(?=(\B)(\d{3})+$)/g**

**var num=“10000000”**

**console.log(num.replace(reg,”,”));**

1.\d是数字

2.\d{3}三位的数字

3.（\d{3}）+三位数的数字组合有多个

4.（\d{3}）+$是以三位数字的组合结尾，就是表示从后往前数

5./(？=（\d{3}）+$)/g是以三位数字的组合结尾前面东西，没有写表示空,这样从后往前数每三个前面的空表示出来了

6./(？=（\B）（\d{3}）+$)/g前面单词边界拿掉

常用的正则表达式

**邮箱**

var mailReg = /^\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*$/;

**电话**

var phoneReg = /^1\d{10}$/;

**身份证**

var idCardReg = /(^\d{15}$)|(^\d{18}$)|(^\d{17}(\d|X|x)$)/;

**中国名字**

var ChineseNameReg = /^[\u4e00-\u9fa5]{2,4}$/;

**普通域名**

var urlReg = /^(http(s)?:\/\/)?(www\.)?[a-zA-Z0-9][-a-zA-Z0-9]{0,62}(\.[a-zA-Z0-9][-a-zA-Z0-9]{0,62})+(:\d+)\*(\/\w+\.\w+)\*([\?&]\w+=\w\*)\*$/; https或http协议的url

**用户名**

var userNameReg = /^[A-Za-z0-9-\_]\*$/; 用户名为数字英文下划线或短划线

**QQ号**

var qqReg = /^[1-9][0-9]{4,}$/;

**邮编**

var mailReg = /^[1-9][0-9]{5}$/;

**HTML标签**

var tagReg = /<[^>]+>/; 可以用来去掉html文本中的标签，得到纯文字

**日期**

var dateReg = /^\d{4}-(0?[1-9]|1[0-2])-(0?[1-9]|[1-2]\d|3[0-1])$/; YYYY-MM-dd格式，短线可以视情况替换

**时间**

var timeReg = /^([0-1]\d|2[0-3]):[0-5]\d:[0-5]\d$/; hh:mm:ss格式，冒号可以视情况替换

**数字**

var numberReg=/^[0-9]\*$/;

**n位的数字**

var numberReg=/^\d{n}$/;

**至少n位的数字**

var numberReg=/^\d{n,}$/;

**m-n位的数字**

var numberReg=/^\d{m,n}$/

**零和非零开头的数字**

var numberReg=/^(0|[1-9][0-9]\*)$/;

**非零开头的最多带两位小数的数字**

var numberReg=/^([1-9][0-9]\*)+(.[0-9]{1,2})?$/

**带1-2位小数的正数或负数**

var numberReg=/^(\-)?\d+(\.\d{1,2})?$/;

**正数、负数、和小数**

var numberReg=/^(\-|\+)?\d+(\.\d+)?$/;

**有两位小数的正实数**

var numberReg=/^[0-9]+(.[0-9]{2})?$/;

**有1~3位小数的正实数**

var numberReg=/^[0-9]+(.[0-9]{1,3})?$/;

**非零的正整数**

var numberReg=/^[1-9]\d\*$ 或 ^([1-9][0-9]\*){1,3}$ /;或/ ^\+?[1-9][0-9]\*$/;

**非零的负整数**

var numberReg=/^\-[1-9][]0-9"\*$ /;或/ ^-[1-9]\d\*$/;

**非负整数**

var numberReg=/^\d+$/; 或 /^[1-9]\d\*|0$/;

**非正整数**

var numberReg=/^-[1-9]\d\*|0$ /;或 /^((-\d+)|(0+))$/;

**非负浮点数**

var numberReg=/^\d+(\.\d+)?$/ ;或 /^[1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*|0?\.0+|0$/;

**非正浮点数**

var numberReg=/^((-\d+(\.\d+)?)|(0+(\.0+)?))$/;

var numberReg=/^(-([1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*))|0?\.0+|0$/;

**正浮点数**

var numberReg=/^[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**

var numberReg=/^-([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**

var numberReg=/^(-?\d+)(\.\d+)?$ /;或/ ^-?([1-9]\d\*\.\d\*|0\.\d\*[1-9]\d\*|0?\.0+|0)$/;

**汉字**

var Chinese=/^[\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]+

**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]\*$

**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]\*)$

**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]+)?$

**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})?$

**规定小数点后面必须有两位**

^[0-9]+(.[0-9]{1,2})?$

**允许用户只写一位小数.下面我们该考虑数字中的逗号了**

^[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))

**正的小数或整数**

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