1。^\d+$　　//匹配非负整数（正整数 + 0）   
2。^[0-9]\*[1-9][0-9]\*$　　//匹配正整数   
3。^((-\d+) ?(0+))$　　//匹配非正整数（负整数 + 0）   
4。^-[0-9]\*[1-9][0-9]\*$　　//匹配负整数   
5。^-?\d+$　　　　//匹配整数   
6。^\d+(\.\d+)?$　　//匹配非负浮点数（正浮点数 + 0）   
7。^(([0-9]+\.[0-9]\*[1-9][0-9]\*) ?([0-9]\*[1-9][0-9]\*\.[0-9]+) ?([0-9]\*[1-9][0-9]\*))$　　//匹配正浮点数   
8。^((-\d+(\.\d+)?) ?(0+(\.0+)?))$　　//匹配非正浮点数（负浮点数 + 0）   
9。^(-(([0-9]+\.[0-9]\*[1-9][0-9]\*) ?([0-9]\*[1-9][0-9]\*\.[0-9]+) ?([0-9]\*[1-9][0-9]\*)))$　　//匹配负浮点数   
10。^(-?\d+)(\.\d+)?$　　//匹配浮点数   
11。^[A-Za-z]+$　　//匹配由26个英文字母组成的字符串   
12。^[A-Z]+$　　//匹配由26个英文字母的大写组成的字符串   
13。^[a-z]+$　　//匹配由26个英文字母的小写组成的字符串   
14。^[A-Za-z0-9]+$　　//匹配由数字和26个英文字母组成的字符串   
15。^\w+$　　//匹配由数字、26个英文字母或者下划线组成的字符串   
16。^[\w-]+(\.[\w-]+)\*@[\w-]+(\.[\w-]+)+$　　　　//匹配email地址   
17。^[a-zA-z]+://匹配(\w+(-\w+)\*)(\.(\w+(-\w+)\*))\*(\?\S\*)?$　　//匹配url   
18。匹配中文字符的正则表达式： [\u4e00-\u9fa5]   
19。匹配双字节字符(包括汉字在内)：[^\x00-\xff]   
20。应用：计算字符串的长度（一个双字节字符长度计2，ASCII字符计1）   
String.prototype.len=function(){return this.replace([^\x00-\xff]/g,"aa").length;}   
21。匹配空行的正则表达式：\n[\s ? ]\*\r   
22。匹配HTML标记的正则表达式：/ <(.\*)>.\* <\/\1> ? <(.\*) \/>/   
23。匹配首尾空格的正则表达式：(^\s\*) ?(\s\*$)  
\* 正则表达式用例   
\* 1、^\S+[a-z A-Z]$ 不能为空 不能有空格 只能是英文字母   
\* 2、\S{6,}        不能为空 六位以上   
\* 3、^\d+$          不能有空格 不能非数字   
\* 4、(.\*)(\.jpg ?\.bmp)$ 只能是jpg和bmp格式   
\* 5、^\d{4}\-\d{1,2}-\d{1,2}$ 只能是2004-10-22格式   
\* 6、^0$            至少选一项   
\* 7、^0{2,}$        至少选两项   
\* 8、^[\s ?\S]{20,}$ 不能为空 二十字以上   
\* 9、^\+?[a-z0-9](([-+.] ?[\_]+)?[a-z0-9]+)\*@([a-z0-9]+(\. ?\-))+[a-z]{2,6}$邮件   
\* 10、\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*([,;]\s\*\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*)\* 输入多个地址用逗号或空格分隔邮件   
\* 11、^(\([0-9]+\))?[0-9]{7,8}$电话号码7位或8位或前面有区号例如（022）87341628   
\* 12、^[a-z A-Z 0-9 \_]+@[a-z A-Z 0-9 \_]+(\.[a-z A-Z 0-9 \_]+)+(\,[a-z A-Z 0-9 \_]+@[a-z A-Z 0-9 \_]+(\.[a-z A-Z 0-9 \_]+)+)\*$   
\*    只能是字母、数字、下划线；必须有@和.同时格式要规范 邮件   
\* 13 ^\w+@\w+(\.\w+)+(\,\w+@\w+(\.\w+)+)\*$上面表达式也可以写成这样子，更精练。   
    14 ^\w+((-\w+) ?(\.\w+))\*\@\w+((\. &brvbar-)\w+)\*\.\w+$  
匹配中文字符的正则表达式： [\u4e00-\u9fa5]   
评注：匹配中文还真是个头疼的事，有了这个表达式就好办了   
  
  
匹配双字节字符(包括汉字在内)：[^\x00-\xff]   
评注：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）   
  
  
匹配空白行的正则表达式：\n\s\*\r   
评注：可以用来删除空白行   
  
  
匹配HTML标记的正则表达式： <(\S\*?)[^>]\*>.\*? ? <.\*? />   
评注：网上流传的版本太糟糕，上面这个也仅仅能匹配部分，对于复杂的嵌套标记依旧无能为力   
  
  
匹配首尾空白字符的正则表达式：^\s\* ?\s\*$   
评注：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   
  
  
匹配Email地址的正则表达式：\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*   
评注：表单验证时很实用   
  
  
匹配网址URL的正则表达式：[a-zA-z]+://[^\s]\*   
评注：网上流传的版本功能很有限，上面这个基本可以满足需求   
  
  
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}$   
评注：表单验证时很实用   
  
  
匹配国内电话号码：\d{3}-\d{8} ?\d{4}-\d{7}   
评注：匹配形式如 0511-4405222 或 021-87888822   
  
  
匹配腾讯QQ号：[1-9][0-9]{4,}   
评注：腾讯QQ号从10000开始   
  
  
匹配中国邮政编码：[1-9]\d{5}(?!\d)   
评注：中国邮政编码为6位数字   
  
  
匹配身份证：\d{15} ?\d{18}   
评注：中国的身份证为15位或18位   
  
  
匹配ip地址：\d+\.\d+\.\d+\.\d+   
评注：提取ip地址时有用   
  
  
匹配特定数字：   
^[1-9]\d\*$　 　 //匹配正整数   
^-[1-9]\d\*$ 　 //匹配负整数   
^-?[1-9]\d\*$　　 //匹配整数   
^[1-9]\d\* &brvbar0$　 //匹配非负整数（正整数 + 0）   
^-[1-9]\d\* &brvbar0$　　 //匹配非正整数（负整数 + 0）   
^[1-9]\d\*\.\d\* &brvbar0\.\d\*[1-9]\d\*$　　 //匹配正浮点数   
^-([1-9]\d\*\.\d\* &brvbar0\.\d\*[1-9]\d\*)$　 //匹配负浮点数   
^-?([1-9]\d\*\.\d\* &brvbar0\.\d\*[1-9]\d\* &brvbar0?\.0+ &brvbar0)$　 //匹配浮点数   
^[1-9]\d\*\.\d\* &brvbar0\.\d\*[1-9]\d\* &brvbar0?\.0+ &brvbar0$　　 //匹配非负浮点数（正浮点数 + 0）   
^(-([1-9]\d\*\.\d\* &brvbar0\.\d\*[1-9]\d\*)) &brvbar0?\.0+ &brvbar0$　　//匹配非正浮点数（负浮点数 + 0）   
评注：处理大量数据时有用，具体应用时注意修正   
  
  
匹配特定字符串：   
^[A-Za-z]+$　　//匹配由26个英文字母组成的字符串   
^[A-Z]+$　　//匹配由26个英文字母的大写组成的字符串   
^[a-z]+$　　//匹配由26个英文字母的小写组成的字符串   
^[A-Za-z0-9]+$　　//匹配由数字和26个英文字母组成的字符串   
^\w+$　　//匹配由数字、26个英文字母或者下划线组成的字符串   
"^[\\w-]+(;　　　　//email地址      
"^[a-zA-z]+://(;　　//url  
匹配中文字符的正则表达式： [\u4e00-\u9fa5]   
评注：匹配中文还真是个头疼的事，有了这个表达式就好办了  
匹配双字节字符(包括汉字在内)：[^\x00-\xff]   
评注：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）  
匹配空白行的正则表达式：\n\s\*\r   
评注：可以用来删除空白行  
匹配HTML标记的正则表达式： <(\S\*?)[^>]\*>.\*? ? <.\*? />   
评注：网上流传的版本太糟糕，上面这个也仅仅能匹配部分，对于复杂的嵌套标记依旧无能为力  
匹配首尾空白字符的正则表达式：^\s\* ?\s\*$   
评注：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  
匹配Email地址的正则表达式：\w+([-+.]\w+)\*@\w+([-.]\w+)\*\.\w+([-.]\w+)\*   
评注：表单验证时很实用  
匹配网址URL的正则表达式：[a-zA-z]+://[^\s]\*   
评注：网上流传的版本功能很有限，上面这个基本可以满足需求  
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}$   
评注：表单验证时很实用  
匹配国内电话号码：\d{3}-\d{8} ?\d{4}-\d{7}   
评注：匹配形式如 0511-4405222 或 021-87888822  
匹配腾讯QQ号：[1-9][0-9]{4,}   
评注：腾讯QQ号从10000开始  
匹配中国邮政编码：[1-9]\d{5}(?!\d)   
评注：中国邮政编码为6位数字  
匹配身份证：\d{15} ?\d{18}   
评注：中国的身份证为15位或18位  
匹配ip地址：\d+\.\d+\.\d+\.\d+   
评注：提取ip地址时有用  
匹配特定数字：   
^[1-9]\d\*$　 　 //匹配正整数   
^-[1-9]\d\*$ 　 //匹配负整数   
^-?[1-9]\d\*$　　 //匹配整数   
^[1-9]\d\* &brvbar0$　 //匹配非负整数（正整数 + 0）   
^-[1-9]\d\* &brvbar0$　　 //匹配非正整数（负整数 + 0）   
^[1-9]\d\*\.\d\* &brvbar0\.\d\*[1-9]\d\*$　　 //匹配正浮点数   
^-([1-9]\d\*\.\d\* &brvbar0\.\d\*[1-9]\d\*)$　 //匹配负浮点数   
^-?([1-9]\d\*\.\d\* &brvbar0\.\d\*[1-9]\d\* &brvbar0?\.0+ &brvbar0)$　 //匹配浮点数   
^[1-9]\d\*\.\d\* &brvbar0\.\d\*[1-9]\d\* &brvbar0?\.0+ &brvbar0$　　 //匹配非负浮点数（正浮点数 + 0）   
^(-([1-9]\d\*\.\d\* &brvbar0\.\d\*[1-9]\d\*)) &brvbar0?\.0+ &brvbar0$　　//匹配非正浮点数（负浮点数 + 0）   
评注：处理大量数据时有用，具体应用时注意修正  
匹配特定字符串：   
^[A-Za-z]+$　　//匹配由26个英文字母组成的字符串   
^[A-Z]+$　　//匹配由26个英文字母的大写组成的字符串   
^[a-z]+$　　//匹配由26个英文字母的小写组成的字符串   
^[A-Za-z0-9]+$　　//匹配由数字和26个英文字母组成的字符串   
^\w+$　　//匹配由数字、26个英文字母或者下划线组成的字符串   
"^[\\w-]+(;　　　　//email地址   
"^[a-zA-z]+://(;　　//urlview plaincopy to clipboardprint?   
正则几个基本概念：  
正则几个基本概念：view plaincopy to clipboardprint?   
1.贪婪：+,\*,?,{m,n}等默认是贪婪匹配，即尽可能多匹配，也叫最大匹配   
如果后面加上?，就转化为非贪婪匹配，需要高版本支持  
1.贪婪：+,\*,?,{m,n}等默认是贪婪匹配，即尽可能多匹配，也叫最大匹配   
如果后面加上?，就转化为非贪婪匹配，需要高版本支持view plaincopy to clipboardprint?   
2.获取：默认用(x &brvbary)是获取匹配，很多时候只是测试，不一定要求得到所匹配的数据，尤其在嵌套匹配或大数据中就要用非获取匹配(?:x &brvbary)，这样提高了效率，优化了程序。  
2.获取：默认用(x &brvbary)是获取匹配，很多时候只是测试，不一定要求得到所匹配的数据，尤其在嵌套匹配或大数据中就要用非获取匹配(?:x &brvbary)，这样提高了效率，优化了程序。view plaincopy to clipboardprint?   
3.消耗：默认是消耗匹配，一般在预查中是非消耗匹配。   
举个例子，2003-2-8要变为2003-02-08   
如果用/-(\d)-/第二次匹配将从8开始，从而只替换第一个2，错误   
如果用/-(\d)(?=-)/则第二次匹配从第二个-开始，即不消耗字符-  
3.消耗：默认是消耗匹配，一般在预查中是非消耗匹配。   
举个例子，2003-2-8要变为2003-02-08   
如果用/-(\d)-/第二次匹配将从8开始，从而只替换第一个2，错误   
如果用/-(\d)(?=-)/则第二次匹配从第二个-开始，即不消耗字符-view plaincopy to clipboardprint?   
4.预查：js中分为正向预查和负向预查   
如上面的(?=pattern)是正向预查，在任何匹配 pattern 的字符串开始处匹配查找字符串。还有(?!pattern)是负向预查，在任何不匹配 pattern 的字符串开始处匹配查找字符串。负向预查有时会用在对[^]的扩充，[^]只是一些字符，而?!可以使整个字符串。  
4.预查：js中分为正向预查和负向预查   
如上面的(?=pattern)是正向预查，在任何匹配 pattern 的字符串开始处匹配查找字符串。还有(?!pattern)是负向预查，在任何不匹配 pattern 的字符串开始处匹配查找字符串。负向预查有时会用在对[^]的扩充，[^]只是一些字符，而?!可以使整个字符串。view plaincopy to clipboardprint?   
5.回调：一般用在替换上，即根据不用的匹配内容返回不用的替换值，从而简化了程序，需要高版本支持  
5.回调：一般用在替换上，即根据不用的匹配内容返回不用的替换值，从而简化了程序，需要高版本支持view plaincopy to clipboardprint?   
6.引用：\num 对所获取的第num个匹配的引用。   
例如，'(.)\1\1' 匹配AAA型。'(.)(.)\2\1' 匹配ABBA型。  
6.引用：\num 对所获取的第num个匹配的引用。   
例如，'(.)\1\1' 匹配AAA型。'(.)(.)\2\1' 匹配ABBA型。view plaincopy to clipboardprint?   
正则表达式保留字   
^ (carat)      
. (period)      
[ (left bracket}      
$ (dollar sign)      
( (left parenthesis)      
) (right parenthesis)      
? (pipe)      
\* (asterisk)      
+ (plus symbol)      
? (question mark)      
{ (left curly bracket, or left brace)      
\ backslash  
正则表达式保留字   
^ (carat)   
. (period)   
[ (left bracket}   
$ (dollar sign)   
( (left parenthesis)   
) (right parenthesis)   
? (pipe)   
\* (asterisk)   
+ (plus symbol)   
? (question mark)   
{ (left curly bracket, or left brace)   
\ backslash view plaincopy to clipboardprint?   
构造 匹配于  
构造 匹配于 view plaincopy to clipboardprint?   
字符      
x 字符 x      
\\ 反斜线字符      
\0n 八进制值的字符0n (0 <= n <= 7)      
\0nn 八进制值的字符 0nn (0 <= n <= 7)      
\0mnn 八进制值的字符0mnn 0mnn (0 <= m <= 3, 0 <= n <= 7)      
\xhh 十六进制值的字符0xhh      
\uhhhh 十六进制值的字符0xhhhh      
\t 制表符('\u0009')      
\n 换行符 ('\u000A')      
\r 回车符 ('\u000D')      
\f 换页符 ('\u000C')      
\a 响铃符 ('\u0007')      
\e 转义符 ('\u001B')      
\cx T对应于x的控制字符 x  
字符   
x 字符 x   
\\ 反斜线字符   
\0n 八进制值的字符0n (0 <= n <= 7)   
\0nn 八进制值的字符 0nn (0 <= n <= 7)   
\0mnn 八进制值的字符0mnn 0mnn (0 <= m <= 3, 0 <= n <= 7)   
\xhh 十六进制值的字符0xhh   
\uhhhh 十六进制值的字符0xhhhh   
\t 制表符('\u0009')   
\n 换行符 ('\u000A')   
\r 回车符 ('\u000D')   
\f 换页符 ('\u000C')   
\a 响铃符 ('\u0007')   
\e 转义符 ('\u001B')   
\cx T对应于x的控制字符 x view plaincopy to clipboardprint?   
字符类      
[abc] a, b, or c (简单类)      
[^abc] 除了a、b或c之外的任意 字符（求反）      
[a-zA-Z] a到z或A到Z ，包含（范围)      
[a-z-[bc]] a到z，除了b和c ： [ad-z]（减去）      
[a-z-[m-p]] a到z，除了m到 p： [a-lq-z]      
[a-z-[^def]] d, e, 或 f  
字符类   
[abc] a, b, or c (简单类)   
[^abc] 除了a、b或c之外的任意 字符（求反）   
[a-zA-Z] a到z或A到Z ，包含（范围)   
[a-z-[bc]] a到z，除了b和c ： [ad-z]（减去）   
[a-z-[m-p]] a到z，除了m到 p： [a-lq-z]   
[a-z-[^def]] d, e, 或 f view plaincopy to clipboardprint?   
预定义的字符类      
. 任意字符（也许能与行终止符匹配，也许不能）      
\d 数字: [0-9]      
\D 非数字: [^0-9]      
\s 空格符: [ \t\n\x0B\f\r]      
\S 非空格符: [^\s]      
\w 单词字符: [a-zA-Z\_0-9]      
\W 非单词字符: [^\w]