**Chapter07 正则表达式&&表单辅助特效**

什么是正则表达式？

回答：

1. 正则表达式是由一个字符序列形成的搜索模式。
2. 当你在文本中搜索数据时，你可以用搜索模式来描述你要查询的内容。
3. 正则表达式可以是一个简单的字符，或一个更复杂的模式。
4. 正则表达式可用于所有文本搜索和文本替换的操作。

正则表达式，又称规则表达式。

在代码中常简写为regex、regexp。正则表达式通常缩写成“regex”，单数有regexp、regex，复数有regexps、regexes、regexen。

语法：/正则表达式主体/修饰符(可选)

一个正则表达式就是由普通字符（例如字符 a 到 z）以及特殊字符（称为元字符）组成的文字模式。该模式描述在查找文字主体时待匹配的一个或多个字符串。正则表达式作为一个模板，将某个字符模式与所搜索的字符串进行匹配。

创建正则表达式 :

var re = new RegExp();//RegExp是一个对象,和Aarray一样

//但这样没有任何效果,需要将正则表达式的内容作为字符串传递进去

re =new RegExp("a");//最简单的正则表达式,将匹配字母a

re=new RegExp("a","i");//第二个参数,表示匹配时不分大小写

RegExp构造函数第一个参数为正则表达式的文本内容,而第一个参数则为可选项标志.标志可以组合使用

•g （全文查找）

•i （忽略大小写）

•m （多行查找）

实例：

var patt = /runoob/i；

实例解析：

/runoob/i 是一个正则表达式。

runoob 是一个正则表达式主体 (用于检索)。

i 是一个修饰符 (搜索不区分大小写)。

**使用字符串方法：**

在 JavaScript 中，正则表达式通常用于两个字符串方法 : search() 和 replace()。

search() 方法 用于检索字符串中指定的子字符串，或检索与正则表达式相匹配的子字符串，并返回子串的起始位置。

replace() 方法 用于在字符串中用一些字符替换另一些字符，或替换一个与正则表达式匹配的子串。

课堂例题：

var val = "abc我就是我123bc";

document.write(val);

//1.判断上面字符串中是否包含字母b

var reg =/Bc/gi;

//reg.test(val) 如果val中包含b字母 那么这个方法返回true 否则返回false

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

console.log(reg.exec(val));

//2.查找上面字符串是不是以ab开始

reg=/^abc$/;

console.log("字符串是否以ab开始："+reg.test(val));

//3.判断上面的字符串是否都是数字组成

reg = /\d{12}/;

reg=/[0-9]{12}/;

console.log("字符串是否由数字组成："+reg.test(val));

//4.判断上面的字符串不能包含数字

reg=/\D{12}/;

reg=/[^0-9]{12}/;

reg=/^\{12}/;

console.log("不能由数字组成："+reg.test(val));

例：

一、校验数字的表达式：

1. 数字：^[0-9]\*$

2. n位的数字：^\d{n}$

3. 至少n位的数字：^\d{n,}$

4. m-n位的数字：^\d{m,n}$

5. 零和非零开头的数字：^(0|[1-9][0-9]\*)$

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

7. 带1-2位小数的正数或负数：^(\-)?\d+(\.\d{1,2})?$

8. 正数、负数、和小数：^(\-|\+)?\d+(\.\d+)?$

9. 有两位小数的正实数：^[0-9]+(.[0-9]{2})?$

10. 有1~3位小数的正实数：^[0-9]+(.[0-9]{1,3})?$

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

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

13. 非负整数：^\d+$ 或 ^[1-9]\d\*|0$

14. 非正整数：^-[1-9]\d\*|0$ 或 ^((-\d+)|(0+))$

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

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

17. 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]\*))$

18. 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]\*)))$

19. 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：

1. 汉字：^[\u4e00-\u9fa5]{0,}$

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

3. 长度为3-20的所有字符：^.{3,20}$

4. 由26个英文字母组成的字符串：^[A-Za-z]+$

5. 由26个大写英文字母组成的字符串：^[A-Z]+$

6. 由26个小写英文字母组成的字符串：^[a-z]+$

7. 由数字和26个英文字母组成的字符串：^[A-Za-z0-9]+$

8. 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}$

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

10. 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}$

11. 可以输入含有^%&',;=?$\"等字符：[^%&',;=?$\x22]+ 12 禁止输入含有~的字符：[^~\x22]+

三、特殊需求表达式：

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

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

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

4. 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}$

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

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

7. 身份证号(15位、18位数字)：^\d{15}|\d{18}$

8. 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}?$

9. 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}$

10. 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}$

11. 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}$

12. 日期格式：^\d{4}-\d{1,2}-\d{1,2}

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

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

15. 钱的输入格式：

16. 1.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]\*$

17. 2.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]\*)$

18. 3.一个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]\*)$

19. 4.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]+)?$

20. 5.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})?$

21. 6.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})?$

22. 7.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})?$

23 8.1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})?$

24. 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

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

26. 中文字符的正则表达式：[\u4e00-\u9fa5]

27. 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))

28. 空白行的正则表达式：\n\s\*\r (可以用来删除空白行)

29. 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)

30. 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)

31. 腾讯QQ号：[1-9][0-9]{4,} (腾讯QQ号从10000开始)

32. 中国邮政编码：[1-9]\d{5}(?!\d) (中国邮政编码为6位数字)

33. IP地址：\d+\.\d+\.\d+\.\d+ (提取IP地址时有用)

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