

脚本之家

软件下载

源码下载

在线工具

网页教程基础

服务器常用软件

手机版

关注微信

投稿中心

NEW

脚本之家
www.jb51.net

GAINET 景安 快云服务器 全线5折

新型高防轮循防御技术
环球高防路由调控源头处理

网页制作 网络编程 脚本专栏 脚本下载 数据库 CMS教程 电子书籍 平面设计 媒体动画 操作系统 网站运营 网络安全 在线手册

JavaScript ASP.NET PHP编程 AJAX相关 正则表达式 ASP编程 JSP编程 编程10000问 CSS/HTML Flex 脚本加解密 web2.0 XML/RSS 网页编辑器 相关技巧 黑客相关 网页播放器

您的位置: 首页 → 网络编程 → 正则表达式 → 正文内容 正则表达式语法

请输入关键词

搜索

低价出售流量10000IP只需8元

装软件赚钱, 每台电脑秒赚6.3元

众生网络 基于云计算的互联网基础服务运

枫信科技-江苏双线10M保证-399/元

鼎点网络百兆独享服务器仅需999元

DDoS高防IP, 300G无限防

美国服务器, 368元/月起, 高速, 稳定

服务器租用/托管-域名空间/认准腾佑科技

免备案国内虚拟主机-163cdn

[香港双高防]无视CC★DDoS/堪比广东!

16核独服350/固态盘独服288/创梦网络

微子网络 湛江双线高防 抗15G 700元起

BGP四线 亿恩1U服务器托管3999元/年

免备vps20/百独799/双线350/45互联

畅游网络 百独服务器 包跑满 998元

美国云20元/香港云29元/韩国云55元

脚本之家 百兆服务器租用、托管

九九数据 一 工信部认可正规资质IDC接入

韩国香港美国站群服务器 巨牛网络

服务器租用 199元起

云主机
仅需45元上云市场
千款软件任您选

免费试用

高防服务器
CN2直连 | 无视DDoS/CC | 24x7运维

免备案 香港空间

高性能云服务器就选阿里云

20000随机读写IOPS
256MB/S吞吐量

75折限时抢

阿里云
230万用户信赖

高性能云服务器限时抢购

首台5折

全网极速BGP访问

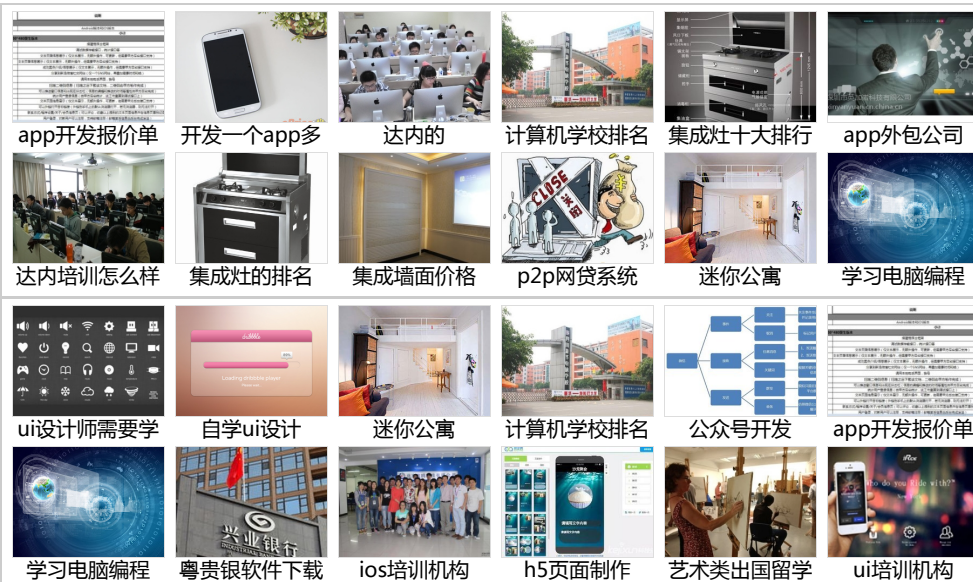


立即购买

正则表达式基本语法详解

作者: OnlyLiu 字体: [增加 减小] 类型: 转载 时间: 2015-10-14 我要评论

本文给大家介绍正则表达式的基本语法, 需要的朋友可以参考下



正则表达式是一种文本模式，包括普通字符（例如，a 到 z 之间的字母）和特殊字符（称为“元字符”）。模式描述在搜索文本时要匹配的一个或多个字符串。

1. 正则表达式基本语法

两个特殊的符号'^'和'\$'。他们的作用是分别指出一个字符串的开始和结束。例子如下：

"^The": 表示所有以"The"开始的字符串（"There", "The cat"等）；

"of despair\$": 表示所以以"of despair"结尾的字符串；

TRADING 212

从交易中获利



大家感兴趣的内容

- 1 Java 正则表达式详解
- 2 javascript 手机号码正则表达式
- 3 正则表达式匹配任意字符（包括换
- 4 js正则函数match、exec、test、s
- 5 grep用法详解 grep与正则表达式
- 6 比较正宗的验证邮箱的正则表达式
- 7 正则中需要转义的特殊字符小结
- 8 js 常用正则表达式表单验证代码
- 9 最常用的PHP正则表达式收集整理
- 10 JS利用正则配合replace替换指定字



始和结尾都是"abc"的字符串——呵呵，只有"abc"自己了；

可包含"notice"的字符串。

家取后加个"了"，如果你不使用两个特殊字符，你就在表示要查找的串在被查找串的任意部分——你并不把它定位在某一个顶端。

其它还有'*'，'+'和'?'这三个符号，表示一个或一序列字符重复出现的次数。它们分别表示"没有或更多"，"一次或更多"还有"没有或一次"。下面是几个例子：

"ab*": 表示一个字符串有一个a后面跟着零个或多个b。（"a", "ab", "abbb",.....）；

"ab+": 表示一个字符串有一个a后面跟着至少一个b或者更多；

"ab?": 表示一个字符串有一个a后面跟着零个或者一个b；

"a?b+\$": 表示在字符串的末尾有零个或一个a跟着一个或几个b。

你也可以使用范围，用大括号括起，用以表示重复次数的范围。

"ab{2}": 表示一个字符串有一个a跟着2个b（"abb"）；

"ab{2,}": 表示一个字符串有一个a跟着至少2个b；

"ab{3,5}": 表示一个字符串有一个a跟着3到5个b。

请注意，你必须指定范围的下限（如："{0,2}"而不是"{,2}"）。还有，你可能注意到了，'*'，'+'和'?'相当于"{0,}"，"{1,}"和"{0,1}"。

还有一个'|'，表示"或"操作：

"hi|hello": 表示一个字符串里有"hi"或者"hello"；

"(b|cd)ef": 表示"bef"或"cdef"；

"(a|b)*c": 表示一串"a""b"混合的字符串后面跟一个"c"；

'.'可以替代任何字符：

"a.[0-9]": 表示一个字符串有一个"a"后面跟着一个任意字符和一个数字；

"^.{3}\$": 表示有任意三个字符的字符串（长度为3个字符）；

方括号表示某些字符允许在一个字符串中的某一特定位置出现：

"[ab]": 表示一个字符串有一个"a"或"b"（相当于"a|b"）；

"[a-d]": 表示一个字符串包含小写的'a'到'd'中的一个（相当于"a|b|c|d"或者"[abcd]"）；

"^[a-zA-Z]": 表示一个以字母开头的字符串；

"[0-9]%": 表示一个百分号前有一位的数字；

"[,a-zA-Z0-9]\$": 表示一个字符串以一个逗号后面跟着一个字母或数字结束。

你也可以在方括号里用'^'表示不希望出现的字符，'^'应在方括号里的第一位。（如："%[^a-zA-Z]%"表示两个百分号中不应该出现字母）。

为了逐字表达，你必须在"^.\$()|*+?{"这些字符前加上转移字符\"。

请注意在方括号中，不需要转义字符。

2. 正则表达式验证控制文本框的输入字符类型

1. 只能输入数字和英文的：

```
<input onkeyup="value=value.replace(/[W]/g, '')" onbeforepaste="clipboardData.setData('text', clipboardData.getData('text').replace(/^[^d]/g, ''))" ID="Text1" NAME="Text1">
```

2. 只能输入数字的：

```
<input onkeyup="value=value.replace(/^[^d]/g, '')" onbeforepaste="clipboardData.setData('text', clipboardData.getData('text').replace(/^[^d]/g, ''))" ID="Text2" NAME="Text2">
```

3. 只能输入全角的：

```
<input onkeyup="value=value.replace(/^[^uFF00-\uFFFF]/g, '')" onbeforepaste="clipboardData.setData('text', clipboardData.getData('text').replace(/^[^uFF00-\uFFFF]/g, ''))" ID="Text3" NAME="Text3">
```

4. 只能输入汉字的：



最近更新的内容

- JavaScript replace new RegExp使用介绍
- [asp]中的正则表达式运用代码
- asp下替换非数字为空的正则
- PHP 正则表达式 推荐
- Javascript 字符串字节长度计算函数代码与
- iOS中使用正则表达式NSRegularExpression
- JScript 和 VBScript 正则表达式
- [php]正则表达式的五个成功习惯
- Oracle正则表达式使用介绍
- javascript正则表达式基础篇



常用在线小工具

- 在线生成二维码工具(加强版)
- 在线JSON代码检验/检验/美化/格式化
- 歇后语在线查询
- JavaScript压缩/格式化/加密工具
- JavaScript代码在线加密工具
- 在线XML格式化/压缩工具
- sql代码在线格式化美化工具
- 在线图片格式转换(jpg/bmp/gif/png)工具
- JavaScript代码格式化工具
- 在线XML/JSON互相转换工具

```
<input onkeyup="value=value.replace(/[\u4E00-\u9FA5]/g, '')" onbeforepaste="clipboardData.setData('text', clipboardData.getData('text').replace(/[\u4E00-\u9FA5]/g, ''))" ID="Text4" NAME="Text4">
```

3. 正则表达式的应用实例通俗说明

//校验是否全由数字组成

/^[0-9]{1,20}\$/

^ 表示打头的字符要匹配紧跟^后面的规则

\$ 表示打头的字符要匹配紧靠\$前面的规则

[] 中的内容是可选字符集

[0-9] 表示要求字符范围在0-9之间

{1,20}表示数字字符串长度合法为1到20，即为[0-9]中的字符出现次数的范围是1到20次。

/^ 和 \$/成对使用应该是表示要求整个字符串完全匹配定义的规则，而不是只匹配字符串中的一个子串。

//校验登录名：只能输入5-20个以字母开头、可带数字、“_”、“.”的字符串

/^[a-zA-Z]{1}([a-zA-Z0-9]|[_.])(4,19)\$/

^[a-zA-Z]{1} 表示第一个字符要求是字母。

([a-zA-Z0-9]|[_.])(4,19) 表示从第二位开始（因为它紧跟在上个表达式后面）的一个长度为4到9位的字符串，它要求是由大小写字母、数字或者特殊字符集[_.]组成。

//校验用户名：只能输入1-30个以字母开头的字符串

/^[a-zA-Z]{1,30}\$/

//校验密码：只能输入6-20个字母、数字、下划线

/^(\w){6,20}\$/

\w: 用于匹配字母，数字或下划线字符

//校验普通电话、传真号码：可以“+”或数字开头，可含有“-”和“ ”

/^[+]{0,1}(\d){1,3}[]?([-]?(\d)[]){1,12})+\$/

\d: 用于匹配从0到9的数字；

“?”元字符规定其前导对象必须在目标对象中连续出现零次或一次

可以匹配的字符串如：+123 -999 999； +123-999 999； 123 999 999； +123 999999等

//校验URL

/^http[s]{0,1}:\\V.+\$/ 或 /^http[s]{0,1}:\\V.{1,n}\$/ (表示url串的长度为length("https://") + n)

\\: 表示字符“/”。

. 表示所有字符的集

+ 等同于{1,}，就是1到正无穷吧。

```
1 //校验纯中文字符
2 /^[u4E00-u9FA5]+$/
```



4.正則表達式應用

5.javascript正则表达式检验

第 4 页 (共 9 页)

```
13 return true
14 }
15 //校验用户名: 只能输入1-30个以字母开头的字符串
16 function isTrueName(s)
17 {
18     var patrn=/^[a-zA-Z]{1,30}$/;
19     if (!patrn.exec(s)) return false
20     return true
21 }
22 //校验密码: 只能输入6-20个字母、数字、下划线
23 function isPasswd(s)
24 {
25     var patrn=/^(\w){6,20}$/;
26     if (!patrn.exec(s)) return false
27     return true
28 }
29 //校验普通电话、传真号码: 可以“+”开头, 除数字外, 可含有“-”
30 function isTel(s)
31 {
32     //var patrn=/^[+]{0,1}(\d){1,3}[ ]?([-]?(\d){1,12})+$/;
33     var patrn=/^[+]{0,1}(\d){1,3}[ ]?([-]?((\d)|[ ]){1,12})+$/;
34     if (!patrn.exec(s)) return false
35     return true
36 }
37 //校验手机号码: 必须以数字开头, 除数字外, 可含有“-”
38 function isMobil(s)
39 {
40     var patrn=/^[+]{0,1}(\d){1,3}[ ]?([-]?((\d)|[ ]){1,12})+$/;
41     if (!patrn.exec(s)) return false
42     return true
43 }
44 //校验邮政编码
45 function isPostalCode(s)
46 {
47     //var patrn=/^[a-zA-Z0-9]{3,12}$/;
48     var patrn=/^[a-zA-Z0-9 ]{3,12}$/;
49     if (!patrn.exec(s)) return false
50     return true
51 }
52 //校验搜索关键字
53 function isSearch(s)
54 {
55     var patrn=/^[^~!@#%$^&*()+=|\\[\]\{\}\:;\'\",.<>/?]{1}[^~!@#%$^&*()+=|\\[\]\{\}\:;\'\",.<>/?]{1}$/;
56     if (!patrn.exec(s)) return false
57     return true
58 }
59 function isIP(s) //by zergling
60 {
61     var patrn=/^[0-9.]{1,20}$/;
62     if (!patrn.exec(s)) return false
63     return true
64 }
65 /*****
66 * FUNCTION: isBetween
67 * PARAMETERS: val AS any value
68 * lo AS Lower limit to check
69 * hi AS Higher limit to check
70 * CALLS: NOTHING
71 * RETURNS: TRUE if val is between lo and hi both inclusive, otherwise false.
72 *****/
73 function isBetween (val, lo, hi) {
74     if ((val < lo) || (val > hi)) { return(false); }
75     else { return(true); }
76 }
77 /*****
78 * FUNCTION: isDate checks a valid date
79 * PARAMETERS: theStr AS String
80 * CALLS: isBetween, isInt
81 * RETURNS: TRUE if theStr is a valid date otherwise false.
82 *****/
83 function isDate (theStr) {
84     var the1st = theStr.indexOf('-');
85     var the2nd = theStr.lastIndexOf('-');
86     if (the1st == the2nd) { return(false); }
87     else {
88         var y = theStr.substring(0,the1st);
89         var m = theStr.substring(the1st+1,the2nd);
90         var d = theStr.substring(the2nd+1,theStr.length);
91         var maxDays = 31;
92         if (isInt(m)==false || isInt(d)==false || isInt(y)==false) {
93             return(false); }
94         else if (y.length < 4) { return(false); }
95         else if (!isBetween (m, 1, 12)) { return(false); }
96         else if (m==4 || m==6 || m==9 || m==11) maxDays = 30;
97         else if (m==2) {
98             if (y % 4 > 0) maxDays = 28;
99             else if (y % 100 == 0 && y % 400 > 0) maxDays = 28;
100            else maxDays = 29;

```

```

101 }
102 if (isBetween(d, 1, maxDays) == false) { return(false); }
103 else { return(true); }
104 }
105 }
106 /*****
107 * FUNCTION: isEuDate checks a valid date in British format
108 * PARAMETERS: theStr AS String
109 * CALLS: isBetween, isInt
110 * RETURNS: TRUE if theStr is a valid date otherwise false.
111 *****/
112 function isEuDate (theStr) {
113   if (isBetween(theStr.length, 8, 10) == false) { return(false); }
114   else {
115     var the1st = theStr.indexOf('/');
116     var the2nd = theStr.lastIndexOf('/');
117     if (the1st == the2nd) { return(false); }
118     else {
119       var m = theStr.substring(the1st+1,the2nd);
120       var d = theStr.substring(0,the1st);
121       var y = theStr.substring(the2nd+1,theStr.length);
122       var maxDays = 31;
123       if (isInt(m)==false || isInt(d)==false || isInt(y)==false) {
124         return(false); }
125       else if (y.length < 4) { return(false); }
126       else if (isBetween(m, 1, 12) == false) { return(false); }
127       else if (m==4 || m==6 || m==9 || m==11) maxDays = 30;
128       else if (m==2) {
129         if (y % 4 > 0) maxDays = 28;
130         else if (y % 100 == 0 && y % 400 > 0) maxDays = 28;
131         else maxDays = 29;
132       }
133       if (isBetween(d, 1, maxDays) == false) { return(false); }
134       else { return(true); }
135     }
136   }
137 }
138 /*****
139 * FUNCTION: Compare Date! Which is the latest!
140 * PARAMETERS: lessDate,moreDate AS String
141 * CALLS: isDate,isBetween
142 * RETURNS: TRUE if lessDate<moreDate
143 *****/
144 function isComdate (lessDate , moreDate)
145 {
146   if (!isDate(lessDate)) { return(false);}
147   if (!isDate(moreDate)) { return(false);}
148   var less1st = lessDate.indexOf('-');
149   var less2nd = lessDate.lastIndexOf('-');
150   var more1st = moreDate.indexOf('-');
151   var more2nd = moreDate.lastIndexOf('-');
152   var lessy = lessDate.substring(0,less1st);
153   var lessm = lessDate.substring(less1st+1,less2nd);
154   var lessd = lessDate.substring(less2nd+1,lessDate.length);
155   var morey = moreDate.substring(0,more1st);
156   var morem = moreDate.substring(more1st+1,more2nd);
157   var mored = moreDate.substring(more2nd+1,moreDate.length);
158   var Date1 = new Date(lessy,lessm,lessd);
159   var Date2 = new Date(morey,morem,mored);
160   if (Date1>Date2) { return(false);}
161   return(true);
162 }
163 /*****
164 * FUNCTION isEmpty checks if the parameter is empty or null
165 * PARAMETER str AS String
166 *****/
167 function isEmpty (str) {
168   if ((str==null)|| (str.length==0)) return true;
169   else return(false);
170 }
171 /*****
172 * FUNCTION: isInt
173 * PARAMETER: theStr AS String
174 * RETURNS: TRUE if the passed parameter is an integer, otherwise FALSE
175 * CALLS: isDigit
176 *****/
177 function isInt (theStr) {
178   var flag = true;
179   if (isEmpty(theStr)) { flag=false; }
180   else
181   { for (var i=0; i<theStr.length; i++) {
182     if (isDigit(theStr.substring(i,i+1)) == false) {
183       flag = false; break;
184     }
185   }
186   }
187   return(flag);
188 }

```

```

189  /*****
190  * FUNCTION: isReal
191  * PARAMETER: heStr AS String
192  decLen AS Integer (how many digits after period)
193  * RETURNS: TRUE if theStr is a float, otherwise FALSE
194  * CALLS: isInt
195  *****/
196  function isReal (theStr, decLen) {
197  var dot1st = theStr.indexOf('.');
198  var dot2nd = theStr.lastIndexOf('.');
199  var OK = true;
200  if (isEmpty(theStr)) return false;
201  if (dot1st == -1) {
202  if (!isInt(theStr)) return(false);
203  else return(true);
204  }
205  else if (dot1st != dot2nd) return (false);
206  else if (dot1st==0) return (false);
207  else {
208  var intPart = theStr.substring(0, dot1st);
209  var decPart = theStr.substring(dot2nd+1);
210  if (decPart.length > decLen) return(false);
211  else if (!isInt(intPart) || !isInt(decPart)) return (false);
212  else if (isEmpty(decPart)) return (false);
213  else return(true);
214  }
215  }
216  /*****
217  * FUNCTION: isEmail
218  * PARAMETER: String (Email Address)
219  * RETURNS: TRUE if the String is a valid Email address
220  * FALSE if the passed string is not a valid Email Address
221  * EMAIL FORMAT: AnyName@EmailServer e.g; webmaster@hotmail.com
222  * @ sign can appear only once in the email address.
223  *****/
224  function isEmail (theStr) {
225  var atIndex = theStr.indexOf('@');
226  var dotIndex = theStr.indexOf('.', atIndex);
227  var flag = true;
228  theSub = theStr.substring(0, dotIndex+1)
229  if ((atIndex < 1) || (atIndex != theStr.lastIndexOf('@')) || (dotIndex < atIndex +
230  { return(false); }
231  else { return(true); }
232  }
233  /*****
234  * FUNCTION: newWindow
235  * PARAMETERS: doc -> Document to open in the new window
236  hite -> Height of the new window
237  wide -> Width of the new window
238  bars -> 1-Scroll bars = YES 0-Scroll Bars = NO
239  resize -> 1-Resizable = YES 0-Resizable = NO
240  * CALLS: NONE
241  * RETURNS: New window instance
242  *****/
243  function newWindow (doc, hite, wide, bars, resize) {
244  var winNew="_blank";
245  var opt="toolbar=0,location=0,directories=0,status=0,menubar=0,";
246  opt+="scrollbars="+bars+",";
247  opt+="resizable="+resize+",";
248  opt+="width="+wide+",";
249  opt+="height="+hite);
250  winHandle=window.open(doc,winNew,opt);
251  return;
252  }
253  /*****
254  * FUNCTION: DecimalFormat
255  * PARAMETERS: paramValue -> Field value
256  * CALLS: NONE
257  * RETURNS: Formated string
258  *****/
259  function DecimalFormat (paramValue) {
260  var intPart = parseInt(paramValue);
261  var decPart =parseFloat(paramValue) - intPart;
262  str = "";
263  if ((decPart == 0) || (decPart == null)) str += (intPart + ".00");
264  else str += (intPart + decPart);
265  return (str);
266  }

```

```

"^\\d+$"    //非负整数（正整数 + 0）
"^[0-9]*[1-9][0-9]*$"    //正整数
"^((-\\d+)|(0+))$"    //非正整数（负整数 + 0）
"^-[0-9]*[1-9][0-9]*$"    //负整数
"^-?\\d+$"    //整数

```

```

"^\\d+(\\.\\d+)?$"    //非负浮点数（正浮点数 + 0）
"^((([0-9]+\\.\\d+)|([0-9]*[1-9][0-9]*\\.\\d+)|([0-9]*[1-9][0-9]*))$"    //正浮点数
"^((-\\d+(\\.\\d+)?)|([0+](\\.0+)?))$"    //非正浮点数（负浮点数 + 0）
"^(-|([0-9]+\\.\\d+)|([0-9]*[1-9][0-9]*\\.\\d+)|([0-9]*[1-9][0-9]*))$"    //负浮点数
"^(-?\\d+)(\\.\\d+)?$"    //浮点数
"^[A-Za-z]+$"    //由26个英文字母组成的字符串
"^[A-Z]+$"    //由26个英文字母的大写组成的字符串
"^[a-z]+$"    //由26个英文字母的小写组成的字符串
"^[A-Za-z0-9]+$"    //由数字和26个英文字母组成的字符串
"^\\w+$"    //由数字、26个英文字母或者下划线组成的字符串
"^([\\w-]+(\\.([\\w-]+)+)*@([\\w-]+(\\.([\\w-]+)+)*\\.([\\w-]+)+))\\.([\\w-]+)+$"    //email地址
"^[a-zA-Z]+:(\\w+(-\\w+)*)(\\.([\\w+(-\\w+)*))*((\\?|\\$)?)$"    //url

```

以上是本文的全部内容，希望大家喜欢。

您可能感兴趣的文章:

[正则表达式的语法](#)

[正则表达式语法](#)

[VBS教程：正则表达式简介 -正则表达式语法](#)

[JScript 8.0 正则表达式语法](#)

[javascript中基于replace函数的正则表达式语法](#)

[正则表达式语法规则及在Javascript和C#中的使用方法](#)

[php正则表达式的基本语法总结](#)

[C# 正则表达式分解和转换IP地址实例\(C# 正则表达式大全 c# 正则表达式语法\)](#)

[js 玩转正则表达式之语法高亮](#)

[PowerShell常用正则表达式和语法参考](#)



html5游戏



集成灶十大排行榜



艺术类出国留学



ui设计师需要学什



u盘制作

[粤贵银软件下载](#) [ui培训机构](#) [迷你公寓](#)

[上海ui培训](#) [集成吊顶](#) [学习电脑编程](#)

[海南三亚房价](#) [自学ui设计](#) [集成墙面价格](#)

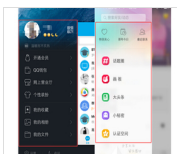
[环保集成灶](#) [快速注销公司](#) [集成吊顶十大](#)

[集成灶的排名](#) [软件工程师薪水](#) [集成墙面..](#)

[墙面集成](#) [app制作平台](#) [集成墙面怎么样](#)

[ui设计教程](#) [h5页面制作](#) [程序员](#)

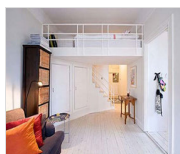
[ios培训班](#) [公司logo设计](#) [快速开发平台](#)



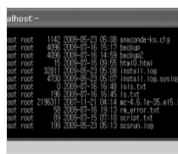
app制作



ios培训班



迷你公寓



ios编程入门



程序员

[app开发报价单](#) [手机app开发](#) [学习电脑编程](#)

[嵌入式就业前景](#) [编程学习网站](#) [农业物联网](#)

[软件工程师薪水](#) [web前端工程师](#)

[ui设计教程](#) [嘉兴二手房](#) [座椅电梯](#)

[快速注销公司](#) [艺术类出国留学](#) [什么是ui..](#)

[u盘价格](#) [看盘软件](#) [通达信软件](#)

[ui设计是什么](#) [u盘制作](#) [编程自学网](#)

[u盘定制](#) [手机app制作](#) [加密锁](#) [ui设计师](#)

Tags: 正则表达式 语法

相关文章

[正则表达式的字符串替换方法](#)

2016-01-01

[截字符串 去除HTML标记](#)

2006-06-06

Javascript中使用exec进行正则表达式全局匹配时的注意事项

正则入门连载！（献给不及格的程序员们）

javascript常用正则表达式合集

用正则按最后一个 "_" 分割"字符

asp.net 正则表达式的应用

在Asp中用"正则表达式对象"来校验数据的合法性

精通 JavaScript中的正则表达式手机整理 推荐

一个匹配形如"#100"的正则表达式

2007-02-02

2010-08-08

2007-05-05

2010-03-03

2007-03-03

2009-10-10

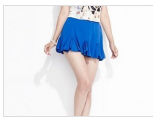
2007-08-08



深圳贷款



黄金分析软件下



货到付款的购物



u盘价格



包装盒设计



数据库



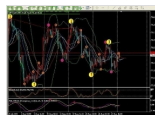
域名申请



u盘制作



棋牌游戏下载



现货软件



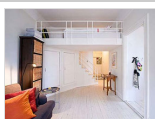
粤贵银软件下载



嘉兴二手房



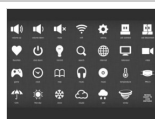
html5游戏



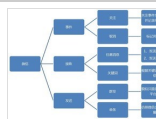
迷你公寓



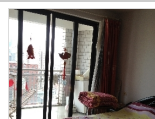
app制作平台



ui设计师需要学



公众号开发



黄浦区二手房



app设计



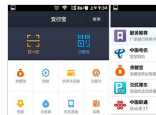
苹果笔记本



移动广告车



上海市房价



小微贷



苹果换屏

最新评论

评论(0人参与, 0条评论)



来说两句吧...

微博登录

QQ登录

手机登录

还没有评论, 快来抢沙发吧!

Powered by 畅言

关于我们 - 广告合作 - 联系我们 - 免责声明 - 网站地图 - 投诉建议 - 在线投稿

©Copyright 2006-2016 JB51.Net Inc All Rights Reserved. 脚本之家 版权所有