后盾网 人人做后盾

www.houdunwang.com

正则表达式

后盾网 2011-2016 v5.0

- 正则表达式被用来检索或替换那些符合某个模式的文本内容。
- 许多程序设计语言都支持利用正则表达式进行字符串操作。例如：JavaScript、PHP、ASP、JAVA、Perl、C#、.NET、ColdFusion、Python、Visual Basic、MYSQL、LINUX、VI编辑器等等语言都支持正则表达式。

简单来说正则表达式就是完成字符串的增、删、改、查

正则表达式是什么？

定界符

- 正则表达式语句需要由分隔符（定界符）闭合包裹，分隔符可以使任意非字母数字, 非反斜线, 非空白字符
- 经常使用的分隔符是正斜线(/), hash符号(#) 以及取反符号(~)。
- 建议使用//做为定界符,因为与js一致

示例：

- $preg= "/houdunwang/";
- $c = preg_match($preg,'后盾网址houdunwang.com');

元字符（原子）

元字符是正则表达式中的最小元素，只代表单一（一个）字符

系统提供的元字符

\d	匹配任意一个数字	[0-9]
\D	与除了数字以外的任何一个字符匹配	[^0-9]
\w	与任意一个英文字母,数字或下划线匹配	[0-9a-zA-Z_]
\W	除了字母,数字或下划线外与任何字符匹配	[^0-9a-zA-Z_]
\s	与任意一个空白字符匹配	[\n\f\r\t\v]
\n	换行字符	
\t	制表符	
\S	与除了空白符外任意一个字符匹配	[^\n\f\r\t\v]

- 假如有这样的场景，如果我们想通过正则查找*符号，但是*在正则中有特殊的意义
- 如果写成/*/这会找到任意除换行符外的字符，所以我们要写成^*/这样是正确的，也就是把*号进行转义处理，恢复他的本来意思

字符转义

在一组字符中匹配某个元字符，在正则表达式中通过元字符表来完成，就是放到[..](方括号)中

[]	只匹配其中的一个原子
[^]	只匹配"除了"其中字符的任意一个原子
[0-9]	匹配0-9任何一个数字
[a-z]	匹配小写a-z任何一个字母
[A-Z]	匹配大写A-Z任何一个字母
.	点在正则中表示除换行符外的任意字符

示例：

- $str ="1.jpg@2.jpg@3.jpg#4.jpg";
- $preg="/[@#]/";
- $arr = preg_split($preg,$str);
- print_r($arr);

元字符表（原子表）

- 如果一次要匹配多个元子，可以通过元子组完成
- 原子组与原子表的差别在于原子组一次匹配多个元子，而原子表则是匹配成功表中的一个元字符就可以
- 元字符组用（ ）表示

示例:

```
1. $str = "后盾官网www.houdunwang.com后盾论坛http://
   bbs.houdunwang.com我在后盾的网名叫houdun";
2. $preg = "/(houdun)wang/is";
3. $newStr= preg_replace($preg,'<span style="color:#f00">\1</
   span>wang',$str);
4. echo $newStr;
```

以上示例是将houdunwang字符串中的houdun描红

元字符组（原子组）

选择修释符 |

| 这个符号带表选择修释符，也就是 | 左右两侧有一个匹配到就可以

示例：

- $str = "http://www.baidu.com与新浪网http://www.sina.com";
- $preg = "/\.(baidu|sina)\./is”;
- $new_str = preg_replace($preg,'.houdunwang.',$str);

重复匹配

如果要重复匹配一些内容时我们要使用重复匹配修饰符，包括以下几种

- *　　重复零次或更多次
- +　　重复一次或更多次
- ?　　重复零次或一次
- {n}　　重复n次
- {n,}　　重复n次或更多次
- {n,m}　　重复n到m次

因为正则最小单位是元字符，而我们很少只匹配一个元字符如a、b所以基本上重复匹配在每条正则语句中都是必用到的内容

正则表达式在进行重复匹配时，默认是贪婪匹配模式，也就是说会尽量匹配更多内容，但是有的时候我们并不希望他匹配更多内容，这时可以通过?进行修饰来禁止重复匹配

- *?　　重复任意次，但尽可能少重复
- +? 重复1次或更多次，但尽可能少重复
- ??　　重复0次或1次，但尽可能少重复
- {n,m}?　　重复n到m次，但尽可能少重复
- {n,}?　　重复n次以上，但尽可能少重复

禁止重复匹配

如果想匹配字符的边界，边界包括空格、标点符号、换行等，可以使用正则表达式的匹配字符边界修饰符如下

匹配字符串边界

- ^　匹配字符串的开始
  $　匹配字符串的结束，忽略换行符

匹配字符边界

正则表达式在执行时会按他们的默认执行方式进行，但有时候默认的处理方式总不能满足我们的需求，所以可以使用模式修正符更改默认方式。

- i　　不区分大小写字母的匹配
- s　　将字符串视为单行，换行符做普通字符看待，使“.”匹 配任何字符
- e　　将替换的字符串作为表达使用

模式修正符

preg_match

搜索subject与pattern给定的正则表达式的一个匹配

语法：

int preg_match (string $pattern , string $subject [, array &$matches [, int $flags = 0 [, int $offset = 0

常用正则操作函数

preg_match_all

搜索subject中所有匹配pattern给定正则表达式 的匹配结果并且将它们以flag指定顺序输出到matches中

语法：

int preg_match_all (string $pattern , string $subject , array & $matches [, int $flags = PREG_PATTERN_ORDER [, int $offset = 0

常用正则操作函数

preg_split

通过一个正则表达式分隔给定字符串

语法：

array preg_split (string $pattern , string $subject [, int $limit = -1 [, int $flags = 0]])

常用正则操作函数

preg_replace

执行一个正则表达式的搜索和替换

语法:

mixed preg_replace (mixed $pattern , mixed $replacement , mixed $subject [, int $limit = -1 [, int &$count]])

搜索subject中匹配pattern的部分, 以replacement进行替换

常用正则操作函数

preg_replace_callback

执行一个正则表达式搜索并且使用一个回调进行替换

语法:

mixed preg_replace_callback (mixed $pattern , callback $callback , mixed $subject [, int $limit = -1 [, int &$count]])

常用正则操作函数

1. 验证用户提交过来的邮箱与网址是否正确
2. 将html文档中的所有链接替换为http://www.houdunwang.com

作业