**第4章 字符串 正则**

**1. 创建一个应用程序：智能表单邮件**



**顾客意见反馈表单，尝试加入智能处理功能，如，在顾客的输入信息中查找一些关键词和短语，然后将邮件发送到合适的雇员的邮箱。**

**例如， 邮件中包含了单词“advertising”，就发送到市场部。如果邮件是来自Bob的vip客户，就发送到Bob的邮箱。**

**feedback.html——表单脚本**

**————————————————————————————————**

**<html>**

**<head><title>Bob's Auto Parts - Customer Feedback</title></head>**

**<body>**

**<h1>Customer Feedback</h1>**

**<p>Please tell us what you think.</p>**

**<form action="processfeedback.php" method="post">**

**<p>Your name:<br/>**

**<input type="text" name="name" size="40" /></p>**

**<p>Your email address:<br/>**

**<input type="text" name="email" size="40" /></p>**

**<p>Your feedback:<br/>**

**<textarea name="feedback" rows="8" cols="40" wrap="virtual" /></textarea></p>**

**<p><input type="submit" value="Send feedback" /></p>**

**</form>**

**</body>**

**</html>**

**————————————————————————————————**

**processfeedback.php——处理邮件表单内容的基本脚本**

**——————————————————————————**

**<?php**

**//create short variable names**

**$name=$\_POST['name'];**

**$email=$\_POST['email'];**

**$feedback=$\_POST['feedback'];**

**//set up some static information**

**$toaddress = "feedback@example.com";**

**$subject = "Feedback from web site";**

**$mailcontent = "Customer name: ".$name."\n".**

**"Customer email: ".$email."\n".**

**"Customer comments:\n".$feedback."\n";**

**$fromaddress = "From: webserver@example.com";**

**//invoke mail() function to send mail**

**mail($toaddress, $subject, $mailcontent, $fromaddress);**

**?>**

**<html>**

**<head>**

**<title>Bob's Auto Parts - Feedback Submitted</title>**

**</head>**

**<body>**

**<h1>Feedback submitted</h1>**

**<p>Your feedback has been sent.</p>**

**</body>**

**</html>**

**————————————————————————————————**

**mail(to, subject, message)函数是用来发送电子邮件的，请查看帮助文档。**

**2. 字符串的格式化**

**A. 字符串的整理：chop() ltrim() trim()**

**清理多余的空格。**

**$name=$\_POST['name'];**

**$email=$\_POST['email'];**

**$feedback=$\_POST['feedback'];**

**改为：**

$name=trim($\_POST['name']);

$email=trim($\_POST['email']);

$feedback=trim($\_POST['feedback']);

**trim() 去除字符串开始和结束位置的空格，去除换行符\n 回车符\r、制表符\t \x0B、结束符\0、空格。**

**ltrim() rtrim() 分别从字符串的左边、右边 除去空格。**

**B. 格式化字符串以便显示**

**1，使用HTML格式化： nl2br()**

**用<br/>代替换行符。**

**2. printf() 将一个格式化了的字符串输出到浏览器中。**

**sprintf() 返回一个格式化了的字符串。**

**3. 改变字符串中的字母的大小写**

**strtoupper() strtolower() ucfirst() ucwords()**

**C. 格式化字符串以便存储：addslashes() stripslashes()**

**' " \ NULL，对字符串来说，有效。但是插入到数据库中会有问题，需要转义，即，前面加\ 。**

$feedback=**addslashes(**trim($\_POST['feedback'])**)**;



**而stripslashes() 函数会移除这些反斜杠。**

**也可以在服务器上执行这些函数，php.ini中的magic\_quotes\_gpc魔术引号配置。gpc，代表GET/POST/cookie. 意味着，来自这些方式、方法的变量，将被自动包括在引号内。**

**3. 用字符串函数连接和分割字符串**

**A. 使用函数explode() implode() join()**

**$email\_array = explode('@', $email);**

**将顾客电子邮件地址分割成两部分：用户名和域名，分别保存在$email\_array[0], $email\_array[1]中。**

**if( $email\_array[1] == "bigcustomer.com" ){**

**$toaddress = "bob@example.com";**

**}else {**

**$toaddress = "feedback@example.com";**

**}**

**解决大小写的问题：**

**if(strtolower( $email\_array[1]) == "bigcustomer.com" ){**

**$toaddress = "bob@example.com";**

**}else {**

**$toaddress = "feedback@example.com";**

**}**

**implode() join()与explode()功能相反。**

**$new\_email = implode('@',$email\_array);**

**B. 使用strtok()函数**

**一次只从字符串中取出一些片段(令牌)，从字符串中一次取出一个单词的处理，比explode()函数效果更好。**

**分隔符，可以是一个字符，也可以是一个字符串，但是会根据分隔符字符串中的每个字符来进行分割，而不是根据整个分割字符串来分割。**

**$token = strtok($feedback,' " ');**

**echo $token . '<br/>';**

**while($token != "") {**

**$token = strtok(" ");**

**echo $token . '<br/>';**

**}**



**将反馈中的每个令牌打印在每一行上，循环直到不再有令牌。空字符串被自动跳过。**

**通常要先使用empty()函数检查表单信息。**

**C. 使用substr()函数 返回给定起点和终点的子字符串。**

**如：**

**$test = "Your customer service is excellent";**

**echo substr($test,1);**

**echo '<br/>';**

**echo substr($test,-9);**

**echo '<br/>';**

**echo substr($test,0,4);**

**echo '<br/>';**

**echo substr($test,5,-13);**



**4. 字符串比较**

**A. 字符串排序：strcmp() strcasecmp() strnatcmp()**

**B. strlen()函数测试字符串的长度**

**echo strlen("hello");**

**电子邮件地址最小长度是6个字符：如a@a.to**

**if(strlen($email)<6){**

**echo "That email address is not valid";**

**exit;**

**}**

**5. 使用字符串函数匹配和替换子字符串**

**A. 在字符串中查找子字符串，strstr() strchr() strrchr() stristr()**

$toaddress = 'feedback@example.com';

//change the $toaddress if the criteria are met.

if(strstr($feedback,'shop'){

$toaddress = 'retail@example.com';

}else if (strstr($feedback,'delivery'){

$toaddress = 'fulfilment@example.com';

}else if (strstr($feedback,'bill'){

$toaddress = 'account@example.com';

}

**变体：strrchr() 从最后出现目标关键字的位置的前面返回被搜索字符串。 stristr()不区分大小写**

**B. 查找子字符串的位置：strpos() strrpos()返回目标关键字子字符串在被搜索字符串中的位置。找不到，则返回false。**

$test = "Hello World ";

echo strpos($test, "o"); //4

echo strpos($test, "o",5); //7

$test = "Hello World ";

**$result = strpos($test,"H") ;**

**if($result ===false){**

**echo "Not found ";**

**}else {**

**echo "Found at position " . $result;**

**}**

**C. 替换子字符串：str\_replace() substr\_replace()**

**$feedback = str\_replace($offcolor,'%!@\*',$feedback);**

**$test = substr\_replace($test,'X',-1);**

**用X 替换$test中的最后一个字符。**

**6. 正则表达式**

**.at 可以匹配 cat sat mat #at**

**[a-z]at 则限定匹配a到z之间的字符。[ ]中的内容，都是一个字符类——一个被匹配字符所属的字符集合。只匹配一个字符。**

**[aeiou] 可以表示元音字母。**

**[a-zA-Z] 表示一个范围。**

**[^a-z] ^表示否，即，匹配任何不在a~z之间的字符。**





**3. 重复**

**\* 表示重复0次或更多次。**

**+ 表示重复1次或更多次。**

**4. 子表达式 将一个表达式分隔为几个子表达式，表示：至少这些字符串中的一个需要精确匹配。**

**如：**

**(very)\*large**

**可以匹配 large、very large、very very large等。**

**5. 子表达式计数**

**{}中的数字来指定内容允许重复的次数。{3} 表示重复3次。{2,4}表示重复2~4次。{2，} 表示至少重复2次。**

**如：**

**（very）{1,3}**

**表示匹配 very、 very very、 very very very。**

**6. 定位到字符串的开始或末尾**

**^ 用在开始，表示子字符串必须出现在被搜索字符串的开始处。**

**$用在末尾，表示子字符串必须出现在字符串的末尾。**

**^bob 匹配字符串开始处bob。**

**com$ 匹配字符串末尾处com。**

**^[a-z]$ 匹配a~z之间的一个字符。**

**7. 分支**

**com|edu|net 匹配其中的一个。**

**8. 匹配特殊字符**

**9. 特殊字符一览表**

**POSIX正则表达式，用于方括号外面特殊字符摘要**



**POSIX正则表达式，用于方括号里面特殊字符摘要**



**10. 在智能表单中的应用**

**A。 在顾客的反馈中查找特定的名词。如，如果希望匹配 shop、customer service 或 retail，需要做3次不同的搜索。而使用正则表达式，就可以同时匹配这3个： shop|customer service | retail**

**B。 验证电子邮箱地址。**



**表示 至少有一个字母、数字、下划线、连字符、点号 或者这些字符组合为开始的字符串。注意，当在一个字符类的开始或末尾处使用点号时，点号将失去其特殊通配符的意义，只能成为一个点号字符。**

**@ 匹配 @**

 **与包含文字、数字字符和连字符的主机名匹配。**

**\. 匹配 . 在字符类外部使用点号，必须对其转义。**

 **匹配域名的剩下部分，它包含字母、数字和连字符。**

**要找到所有无效email地址，几乎是不可能的。注意：排斥1%的有效数据的校验函数，比允许出现10%的无效数据的校验函数，还要麻烦！**

**7. 用PHP函数，用正则表达式查找字符串**

**ereg() eregi()函数**

***bool ereg ( string pattern, string string [, array regs] )***

**以区分大小写的方式在 string 中寻找与给定的正则表达式 pattern 所匹配的子串。**

**如果找到与 pattern 中圆括号内的子模式相匹配的子串并且函数调用给出了第三个参数 regs，则匹配项将被存入 regs 数组中。$regs[1] 包含第一个左圆括号开始的子串，$regs[2] 包含第二个子串，以此类推。$regs[0] 包含整个匹配的字符串。**

**if(!eregi('^[a-zA-Z0-9\_\-\.]+@[a-zA-Z0-9\-]+\.[a-zA-Z0-9\-\.]+$' ,$email)){**

**echo "<p>that is not a valid email address.</p>".**

**exit;**

**}**

**$toaddress = "feedback@example.com";**

**if(eregi("shop|customer service|retail"),$feedback))**

**$toaddress ="retail@example.com";**

**} else if(eregi("deliver|fulfill",$feedback)){**

**$toaddress = "fulfillment@example.com";**

**} else if(eregi("bill|account",$feedback)){**

**$toaddress = "accounts@example.com";**

**}**

**if(eregi("bigcustomer\.com',$email)){**

**$toaddress = "bob@example.com";**

**}**

**8. 使用正则表达式替换子字符串**

**与 str\_replace()函数意义，可以使用两个函数：ereg\_replace() eregi\_replace()**

**9. 使用正则表达式分割字符串**

**split()函数**

***array split ( string pattern, string string [, int limit] )***

**提示: preg\_split() 函数使用了 Perl 兼容正则表达式语法，通常是比 split() 更快的替代方案。如果不需要正则表达式的威力，则使用 explode() 更快，这样就不会招致正则表达式引擎的浪费。**

**本函数返回一个字符串数组，每个单元为 string 经区分大小写的正则表达式 pattern 作为边界分割出的子串。如果设定了 limit，则返回的数组最多包含 limit 个单元，而其中最后一个单元包含了 string 中剩余的所有部分。如果出错，则 split() 返回 FALSE。**

**该函数对分割Email地址、域名、日期，是非常有用的。**

**$address = "username@example.com";**

**$arr = split("\.|@", $address);**

**while(list($key,$value) = each($arr)){**

**echo "<br/>".$value;**

**}**

**以上代码，将主机名分割为3个部分，并分别输出到一行**

