

[Windows Server技术放大镜](#)[面向业务的运维 IT短剧解疑难](#)[辟虚拟化新天地 进思杰技术社区](#)[即刻参加微软Windows Phone培训](#)

开发频道

[首页](#) | [Java](#) | [.NET](#) | [Web](#) | [XML](#) | [语言工具](#) | [测试](#) | [游戏](#) | [移动](#) | [架构](#) | [项目管理](#) | [全部文章](#)

搜索

您所在的位置：[首页](#) > [开发](#) > [架构&设计](#) >

Perl grep函数用法详解

<http://developer.51cto.com> 2010-07-23 13:53 leitinghui csdn.net [我要评论\(0\)](#)

Perl grep函数有很多值得学习的地方，它用法你是否熟悉，本文向大家简单介绍一下，希望本文的介绍能让你有所收获。

本文和大家重点学习一下Perl grep函数的用法，Perl grep函数有两种表达方式，具体是哪两种请看本文的详细介绍吧。

Perl grep函数详解

(一)grep有2种表达方式：

◆grepBLOCKLIST

◆grepEXPR,LIST

BLOCK表示一个code块，通常用{}表示；EXPR表示一个表达式，通常是正则表达式。原文说EXPR可是任何东西，包括一个或多个变量，操作符，文字，函数，或子函数调用。

[关注](#) 手把手教你实现.NET程序打包

.NET程序进行打包，主要是为了方便用户进行安装。

LIST是要匹配列表。

Perl grep函数对列表里的每个元素进行BLOCK或EXPR匹配，它遍历列表，并临时设置元素为\$_。在列表上下文里，grep返回匹配命中的所有元素，结果也是个列表。在标量上下文里，grep返回匹配命中的元素个数。

(二)grepvs.loops

```
openFILE"printgrep/terrorism|nuclear/i,;;
```

这里打开一个文件myfile，然后查找包含terrorism或nuclear的行。;返回一个列表，它包含了文件的完整内容。可能你已发现，如果文件很大的话，这种方式很耗费内存，因为文件的所有内容都拷贝到内存里了。

代替的方式是使用loop（循环）来完成：

```
01. while($line=;){
02.   if($line=~ /terrorism|nuclear/i){print$line}
03. }
```

上述code显示，loop可以完成grep能做的任何事情。那为什么还要用grep呢？答案是grep更具Perl风格，而loop是C风格的。

更好的解释是：（1）grep让读者更显然的知道，你在从列表里选择某元素；（2）grep比loop简洁。

一点建议：如果你是Perl新手，那就规矩的使用loop比较好；等你熟悉Perl了，就可使用grep这个有力的工具。

(三)几个Perl grep函数的示例

1.统计匹配表达式的列表元素个数

```
$num_apple=grep/^apple$/i,@fruits;
```

TOP10

24小时	本周	本月
01	Eclipse插件大全 挑选最牛的TOP30	
02	15个国外最流行的CSS框架（附下载）	
03	年度最热门SNS开源系统出炉（附下载）	
04	3种类型的程序员：“虫族”，“人族”，“神族”	
05	Google首席Java架构师谈数学与程序员的关系	
06	程序员的十个层次 你属于哪一层？	
07	2010年12月编程语言排行榜：触摸经典语言化石	
08	10个国外开发的HTML 5精彩应用	
09	您的WebApp真的需要jQuery吗？	
10	JDK最新版本下载及JDK安装与配置	

TAG 热点标签

刀片服务器 云计算 ARP攻防 思科培训

 订阅51CTO 邮刊

点击这里查看样刊

立即订阅

全站热点

在标量上下文里，`grep`返回匹配中的元素个数；在列表上下文里，`grep`返回匹配中的元素的一个列表。

所以，上述code返回apple单词在@fruits数组中存在的个数。因为\$num_apple是个标量，它强迫grep结果位于标量上下文里。

2.从列表里抽取唯一元素

```
01. @unique=grep{++$count{$_}<2}
02. qw(abacddefghh);
03. print"@uniquen";
```

上述code运行后会返回：abcdefgh

即qw(abacddefghh)这个列表里的唯一元素被返回了。为什么会这样呀？让我们看看：

%count是个hash结构，它的key是遍历qw()列表时，逐个抽取的列表元素。++\$count{\$_}表示\$_对应的hash值自增。在这个比较上下文里，++\$count{\$_}与\$count{\$_}++的意义是不一样的哦，前者表示在比较之前，就将自身值自增1；后者表示在比较之后，才将自身值自增1。所以，++\$count{\$_}<2表示将\$count{\$_}加1，然后与2进行比较。\$count{\$_}值默认是undef或0。所以当某个元素a第一次被当作hash的关键字时，它自增后对应的hash值就是1，当它第二次当作hash关键字时，对应的hash值就变成2了。变成2后，就不满足比较条件了，所以a不会第2次出现。

所以上述code就能从列表里唯一1次的抽取元素了。

◆抽取列表里精确出现2次的元素

```
01. @crops=qw(wheatcornbarleyricecornsoybeanhay
02. alfalfaricehaybeetscornhay);
03. @duplicates=grep{$count{$_}==2}
```

- [选择网络操作系统应考虑的因素](#)
- [Chrome浏览器加速推进Linux和Mac版](#)
- [Linux下新手基本操作及技巧看图上路](#)
- [DB2基础:日期和时间的使用](#)
- [7月第一周安全回顾 新类型E-mail攻击出现 ..](#)
- [Oracle SOA Suite预览尝鲜（连载）](#)
- [PalmSource移软推出数据同步化解决方案](#)
- [W3C正起草新VoiceXML标准 解决安全问题](#)
- [Linux的优点](#)
- [使用Eclipse和RDT插件进行Ruby开发](#)

```
04. grep{++$count{$_}>1}@crops;
05. print"@duplicatesn";
```

运行结果是：rice

这里grep了2次哦，顺序是从右至左。首先grep{++\$count{\$_}>1}@crops;返回一个列表，列表的结果是@crops里出现次数大于1的元素。

然后再对产生的临时列表进行grep{\$count{\$_}==2}计算，这里的意思你也该明白了，就是临时列表里，元素出现次数等于2的被返回。

所以上述code就返回rice了，rice出现次数大于1，并且精确等于2，明白了吧?:-)

3.在当前目录里列出文本文件

```
01. @files=grep{-fand-T}glob'*. *';
02. print"@filesn";
```

这个就很容易理解哦。glob返回一个列表，它的内容是当前目录里的任何文件，除了以'.'开头的。{}是个code块，它包含了匹配它后面的列表的条件。这只是grep的另一种用法，其实与grepEXPR,LIST这种用法差不多了。-fand-T匹配列表里的元素，首先它必须是个普通文件，接着它必须是个文本文件。据说这样写效率高点哦，因为-T开销更大，所以在判断-T前，先判断-f了。

4.选择数组元素并消除重复

```
01. @array=qw(Tobeornottobethatisthequestion);
02. @found_words=
03. grep{$_ =~ /b|o|iand++$counts{$_}<2;}@array;
04. print"@found_wordsn";
```

运行结果是：Tobeornottoquestion

`{}`里的意思就是，对`@array`里的每个元素，先匹配它是否包含**b**或**o**字符（不分大小写），然后每个元素出现的次数，必须小于2（也就是1次啦）。

`grep`返回一个列表，包含了`@array`里满足上述2个条件的元素。

5.从二维数组里选择元素，并且x

```
01. #Anarrayofreferencestoanonymousarrays
02. @data_points=([5,12],[20,-3],
03. [2,2],[13,20]);
04. @y_gt_x=grep{$_->[0]<$_->[1]}@data_points;
05. foreach$xy(@y_gt_x){print"$xy->[0],$xy->[1]n"}
```

运行结果是：

5,12

13,20

这里，你应该理解匿名数组哦，`[]`是个匿名数组，它实际上是个数组的引用（类似于C里面的指针）。

`@data_points`的元素就是匿名数组。例如：

```
foreach(@data_points){
print$_->[0];}
```

这样访问到匿名数组里的第1个元素，把0替换成1就是第2个元素了。

所以`{$_->[0]<$_->[1]}`就很明白了哦，它表示每个匿名数组的第一个元素的值，小于第二个元素的值。而`grep{$_->[0]<$_->[1]}@data_points`就会返回满足上述条件的匿名数组列表。所以，就得到你要的结果啦！

6.简单数据库查询

Perl grep函数的{}复杂程度如何，取决于program可用虚拟内存的数量。如下是个复杂的{}示例，它模拟了一个数据库查询：

```
01.  #@databaseisarrayofreferencestoanonymoushashes
02.  @database=(
03.    {name=>"WildGinger",
04.    city=>"Seattle",
05.    cuisine=>"AsianThaiChineseKoreanJapanese",
06.    expense=>4,
07.    music=>"",
08.    meals=>"lunchdinner",
09.    view=>"",
10.    smoking=>"",
11.    parking=>"validated",
12.    rating=>4,
13.    payment=>"MCVISAAMEX",
14.  },
15.  #{...},etc.
16. );
17.
18. subfindRestaurants{
19.   my($database,$query)=@_;
20.   returngrep{
21.     $query->{city}?
22.     lc($query->{city})eqlc($_->{city}):1
23.     and$query->{cuisine}?
```

```

24. $_->{cuisine}=~/ $query->{cuisine}/i:1
25. and$query->{min_expense}?
26. $_->{expense}>=$query->{min_expense}:1
27. and$query->{max_expense}?
28. $_->{expense}<=$query->{max_expense}:1
29. and$query->{music}?$_->{music}:1
30. and$query->{music_type}?
31. $_->{music}=~/ $query->{music_type}/i:1
32. and$query->{meals}?
33. $_->{meals}=~/ $query->{meals}/i:1
34. and$query->{view}?$_->{view}:1
35. and$query->{smoking}?$_->{smoking}:1
36. and$query->{parking}?$_->{parking}:1
37. and$query->{min_rating}?
38. $_->{rating}>=$query->{min_rating}:1
39. and$query->{max_rating}?
40. $_->{rating}<=$query->{max_rating}:1
41. and$query->{payment}?
42. $_->{payment}=~/ $query->{payment}/i:1
43. }@$database;
44. }
45.
46. %query=(city=>'Seattle', cuisine=>'Asian|Thai');
47. @restaurants=findRestaurants(@database,%query);
48. print"$restaurants[0]->{name}n";
49.

```

运行结果是：WildGinger

【编辑推荐】

[Eclipse平台中Perl脚本开发](#)

[Perl学习笔记----Perl命令行](#)

[Perl数组和引用使用指导](#)

[Perl基础 解析Perl标量和数组概念](#)

[Perl模式匹配中的特殊字符用法指南](#)

【责任编辑：[程华权](#) TEL：（010）68476606】

原文：[Perl grep函数用法详解](#)

标 签：[Perl grep函数](#)

上一篇：[实例解析Perl语言中BEGIN和END用法](#) 下一篇：[学习笔记 Perl grep函数使用揭秘](#)

网友评论

[查看所有评论（0）](#)

发表评论

通行证： 密码： [注册通行证](#)

发表评论请注意语言文明

51CTO.com
技术成就梦想

验证码： 请点击后输入验证码

☐ 匿名发表

► 读 书

▪ [Web信息架构：设计大型网站（第3版）](#)

► 论 坛

▪ [数据库应用技术](#)

- [Google AdSense实战宝典：谷歌广告网..](#)
- [如何管理软件企业](#)
- [Java Web开发实战宝典](#)
- [操作系统教程（第3版）](#)

- [如何配置VMware虚拟机网络环境全程..](#)
- [sybase培训教材](#)
- [Sybase ASE快速参考手册](#)
- [sybase语法大全](#)

► 博 客

- [解决了类似分页的代码问题](#)
- [简单留言本\(数据库版\)](#)
- [病毒历险记](#)
- [一个学期一堆感想](#)
- [最近发现](#)

► 下 载

- [phpmailer 发送验证邮件](#)
- [music173动漫吧全站完整](#)
- [metatraffic统计计数汉化](#)
- [q-searchv1.0build20040315](#)
- [newxpsql版下载](#)

Copyright©2005-2010 [51CTO.COM](#) 版权所有 未经许可 请勿转载