

[下载APP](#)[开源软件](#)[问答](#)[动弹](#)[博客](#)[翻译](#)[资讯](#)[码云](#)[代码托管](#)[更多](#)[众包](#)[注册](#)[SibylY的个人空间](#) > [perl](#) > [正文](#)

## Perl常用模块 (Data::Dumper) 原



SibylY 发布于 2014/06/13 18:09 字数 2152 阅读 5458 收藏 6 点赞 1 评论 0

### 为什么使用引用?

在perl4中, hash表中的value字段只能是scalar, 而不能是list, 这对于有些情况是很不方便的, 比如有下面的数据:



大家都在搜...



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

注册

Perldumper, Germany

Berlin, Germany

Washington, USA

Helsinki, Finland

New York, USA

我们想要按国家将城市分类，每个国家后面对应城市列表，如果用perl4来做，必须将城市列表组合成字符串才行，如果用perl5就可以用引用有了引用，就可以构造复杂的hash结构，就可以用列表作为hash的值了。



## 如何定义引用?

1、定义变量的时候，在变量名前面加个\，就得到了这个变量的一个引用，比如

# 数组的引用

```
my@array= (1,2,3) ;
```

```
my$aref=\@array ;
```

#哈希的引用

```
my%hash= ("name"=>"zdd","age"=>30,"gender"=>"male") ;
```

```
my$href=\%hash ;
```

#标量的引用

```
my$scalar=1 ;
```

```
my$sref=\$scalar
```

## 2、匿名引用

方法一不是很常用，最常用的还是匿名引用，方法如下



大家都在搜....



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

注册

```
$aref= [ 1,"foo",undef,13 ];
```

匿名数组的元素仍然可以是匿名数组，所以我们可以用这种方法构造数组的数组，可以构造任意维度的数组。

```
my $aref = [  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9],  
]
```



匿名哈希引用-用{}定义

```
$href= { APR =>4, AUG =>8 };
```

## 使用引用

定义了引用之后，可以使用不同的方法来访问引用，这里主要有三种方法。记忆这三种方法有个诀窍，将他们与普通的变量访问作比较即可。

1、与普通变量的访问方法相比，假设原来的变量名是name，则此方法在所有name出现的地方用\$name代替，如下

```
my $scalar = 1 ;  
my @array = (1, 2, 3) ;  
my %hash = ('zdd' => 30, 'autumn' => 27) ;  
  
my $sref = \$scalar ;    # scalar reference  
my $aref = \@array ;     # array reference  
my $href = \%hash ;      # hash reference
```



大家都在搜...



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

注册

```
print @$aret, "\n" ;    # 用$aret代替aret
print %$href, "\n" ;    # 用$href代替href
print $$aref[2], "\n" ;
print $$href{'zdd'}, "\n" ;
```

2、与普通变量的访问方法相比，假设变量原来的名字是name，则现在用{\$name}来代替name。

@a	@{\$aref}	An array
reverse@a	reverse @{\$aref}	Reverse the array
\$a[3]	\${aref}[3]	An element of the array
\$a[3] =17;	\${aref}[3] =17	Assigning an element



同理，哈希引用的使用方法如下。

%h	%{\$href}	A hash
keys%h	keys%{\$href}	Get the keys from the hash
\$h{'red'}	\${href}{'red'}	An element of the hash
\$h{'red'} =17	\${href}{'red'} =17	Assigning an element

注意：当{}内部是\$var的形式时，{}是可以省略的，也就是说@{\$aref}等价于@\$aref，不过初学最好养成使用{}的习惯。

3、前两种方法比较繁琐，这种很简洁，就是使用箭头符号->

```
$aref->[]    #数组解引用

$href->{}     #哈希解引用

$href->()     #子过程解引用
```



大家都在搜...



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

注册

```
$sref=2 ;
```

## 解引用总结:

```
my $scalar = 1 ;
my @array = (1, 2, 3) ;
my %hash = ('zdd' => 30, 'autumn' => 27) ;
```

```
my $sref = \$scalar ;    # scalar reference
my $aref = \@array ;     # array reference
my $href = \%hash ;      # hash reference
```

### # 方法一

```
print $$sref, "\n" ;
print @$aref, "\n" ;
print %$href, "\n" ;
print $$aref[2], "\n" ;
print $$href{'zdd'}, "\n" ;
```

### # 方法二

```
print ${$sref}, "\n" ;
print @{$aref}, "\n" ;
print %{$href}, "\n" ;
print ${$aref}[2], "\n" ;
print ${$href}{'zdd'}, "\n" ;
```

### # 方法三, 不适用于标量

```
print $aref->[0], "\n" ;
print $href->{'zdd'}, "\n" ;
```





大家都在搜...



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

注册

```
@a = (  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
)
```

我们知道[1, 2, 3]定义了一个（1， 2， 3）的匿名引用，所以数组a实际上包含三个元素，每个元素是一个引用，该引用指向一个数组，所以我们可以用下面的方法来访问数组元素（注意，下标从0开始）

`$a[1][2]`表示第二行第三列元素6，也可以写成`$a[1]->[2]`，不过很少有人这么写。还可以写成`${$a[1]}[2]`，几乎没人这么写！

## 多维数组的引用

```
my $aref = [1, [2, 3], [4, 5, 6]] ;  
print $aref->[0] , "\n" ; #1  
print $aref->[1][1], "\n" ; #3  
print $aref->[2][0], "\n" ; #4
```

- 前者是真正的数组，所以定义变量是使用@，后者是指向匿名数组的引用，所以定义的时候使用\$
- 前者的数组元素是匿名数组，而外层数组则是实体数组，后者无论元素还是外层数组都是匿名数组
- 前者可以用`$a[x][y]`的形式访问，而后者只能用解引用的方式访问，即`$a->[x][y]`的形式。

## 哈希的哈希

也就是哈希表中的每个元素也是一个哈希表，比如一个学生集合组成的哈希，其key是学生名字（唯一），其值是每个学生的属性，比如年龄，身高及学号等。



大家都在搜....



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

注册

```
    'age' => 30,  
    'hight' => 170,  
    'id' => '001',  
},  
  
    'autumn' => {  
        'age' => 27,  
        'hight' => 165,  
        'id' => '002',  
    }  
};赋值
```



## 引用的赋值

`$aref2 = $aref1`; 将使得`$aref2`和`$aref1`指向同一个数组，如果想将`$aref1`指向的数组拷贝一份给`$aref2`的话，使用下面的方法，`[]`里面对数组进行解引用，而`[]`以解引用后的数组为内容生成了一个新的匿名数组，又赋值给`$aref2`。

```
$aref2 = [ @{$aref1} ];
```

注意：不能使用下面的形式，外层的`[]`是不可缺少的。由于`=`左边是标量，所以右边的数组会被解释为标量环境，得到的是数组元素个数，而不是元素本身。但是如果加上`[]`就可以了，这样perl知道这是一个匿名数组的赋值。

```
$aref2 = @{$aref1};
```

## 判断一个变量是否是引用



大家都在搜...



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

活动

```
my $aref1 = [1, 2, 0] ;

print ref $aref1, "\n" ; #输出 ARRAY

if (ref $aref1) {
    print "true\n" ; #输出 true
}
```

判断两个引用是否指向同一个目标

可以用eq，这将以字符串的形式判断，也可以使用==

```
my $aref1 = [1, 2, 0] ;
my $aref2 = $aref1 ;
print $aref1, "\n" ;
print $aref2, "\n" ;

if ($aref1 eq $aref2) {
    print "reference equal\n" ;
}
if($aref1 == $aref2) {
    print "reference equal\n" ;
}
```

## Data::Dumper

stringified perl data structures, suitable for both printing and eval





OSCHINA

大家都在搜...



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

活动

# simple procedural interface

print Dumper(\$foo, \$bar);

# extended usage with names

print Data::Dumper-&gt;Dump([\$foo, \$bar], [qw(foo \*ary)]);



Data::Dumper有面向对象和直接使用函数两种调用方法,

```
use Data::Dumper;
my @fruit = qw(apple banana orang);
my @vegetable = ("tomato","potato","cabbage",\@fruit);
my @meat = ("chicken","fish","beaf",\@vegetable);
my $ref = \@meat;
print "ref:@$ref","\n";
print "meat:@meat","\n";
print Dumper($ref);
print Dumper(@meat);
```

结果:

```
[mtk07256@mbjswgbm502 keyword_scan2]$perl testre.pl
```

```
ref:chicken fish beaf ARRAY(0xf211d8)
```

```
meat:chicken fish beaf ARRAY(0xf211d8)
```

```
$VAR1 = [
```



大家都在搜....

Q

下载APP

开源软件

问答

动弹

博客

翻译

资讯

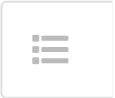
码云

代码托管

更多

众包

招聘



```
.....,
'beaf',
[
    'tomato',
    'potato',
    'cabbage',
    [
        'apple',
        'banana',
        'orang'
    ]
]
];
$VAR1 = 'chicken';
$VAR2 = 'fish';
$VAR3 = 'beaf';
$VAR4 = [
    'tomato',
    'potato',
    'cabbage',
    [
        'apple',
        'banana',
        'orang'
```



J'

大家都在搜....



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

注册

## Data::Dumper 和 Storage

结合Data::Dumper和Storable存储和重新获取数据。你可以用U盘将数据拷走

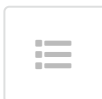
```
#!/usr/bin/perl -w
use Data::Dumper;
use Storable;
```

```
my $a = "good";
my @myarray = ("hello", "world", "123", 4.5);
my %myhash = ( "foo" => 35,
               "bar" => 12.4,
               "2.5"=> "hello",
               "wilma" => 1.72e30,
               "betty" => "bye/n");
```

```
print Dumper($a) . "\n"x2;
print Dumper(\@myarray) . "\n"x2;
print Dumper(\%myhash) . "\n"x2;
print Dumper((\%myhash, \@myarray)) . "\n"x2;
```

```
#!/usr/bin/perl -w
use Data::Dumper;
use Storable;
```

```
my $a = "good";
my @myarray = ("hello", "world", "123", 4.5);
my %myhash = ( "foo" => 35,
```





大家都在搜...



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

招聘

```
"wilma" => 1.72e30,  
"betty" => "bye/n");
```

```
print Dumper($a) ."\n"x2;
```

```
print Dumper(\@myarray) ."\n"x2;
```

```
print Dumper(\%myhash) ."\n"x2;
```

```
print Dumper((\%myhash, \@myarray)) ."\n"x2;
```

```
###use Storable
```

```
print "\nmethod 1,use Storable retrieve data:\n";
```

```
store \%myhash,'./file.txt'; #保存数据
```

```
my $hashref=retrieve('./file.txt'); #重新获取数据
```

```
print Dumper(\%$hashref);
```



由于在linux系统执行store所生成的文件，在Windows下可能无法识别，而Dumper生成的文本文件可以跨系统恢复。

如果你想保存你的数据结构以便以后用于其他程序，那么你有很多方法可以用。最简单的方法就是使用 Perl 的 Data::Dumper 模块，它把一个（可能是自参考的）数据结构变成一个字符串，你可以把这个字符串 保存在程序外部，以后用 eval 或者 do 重新组成：

```
use Data::Dumper;  
my %hash=('a'=>1,'b'=>2);  
open (FILE, "> hash_dumper") or die "can't open tvinfo: $!";  
print FILE D;  
close FILE or die "can't close tvinfo: $!";
```

其他的程序（或者同一个程序）可以稍后从文件里把它读回来：



大家都在搜...



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

注册

在使用 use strict 时失败。

```
open (FILE, "< hash_file") or die "can't open hash_file: $!";  
local $/=undef;          # 一次把整个文件读取进来  
my $file=<FILE>;  
my $hashref= eval $file; # 重新生成##eval{$file}; 是无效的####  
die "can't recreate tv data from tvinfo.perldata: $" if $@;  
close FILE               or die "can't close tvinfo: $!";  
print %$hashref;
```



还有许多其他的解决方法可以用，它们的存储格式的范围从打包的二进制（非常快）到 XML（互换性非常好）。检查一下靠近你的 CPAN 镜像！

© 著作权归作者所有

¥ 打赏

👍 点赞 (1)

☆ 收藏 (6)

➦ 分享

🚩 举报

◀ 上一篇: Perl常用模块 (Process)

下一篇: diff结果分析 ▶



SibylY

粉丝 30 博文 439 码字总数 360908 作品 0

📍 海淀 🏢 程序员



大家都在搜...



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

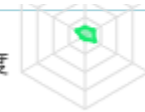
众包

注册

相关文章

最新文章

开源贡献度



技术贡献度

活动活跃性

## perl 的调试和性能测试

perl程序的调试，以前玩php都是print,var\_dump(), firebug之类来调试。在perl中，也有很多方法来调试的。在我看来调试分二种，一种是功能调试（排错之类保证功能完整）。一种是性能调试。...

AlphaJay 2012/04/05 0 0



## mysql二进制免编译安装

MySQL安装 MySQL常用的安装包：rpm、源码、二进制免编译 cd /usr/local/src //进入这个目录下 [root@localhost ~]# cd /usr/local/src wget http://mirrors.sohu.com/mysql/MySQL-5.6/mysql.....

zenw 2017/12/27 0 0

## LAMP架构介绍、MYSQL介绍、安装

LAMP 架构 > LAMP 指的Linux（操作系统）、Apache（httpd服务器）、MySQL（数据库软件）和PHP（有时也是指Perl或Python）的缩写，一般用来建立web服务器（三个角色可以在一台机器，也可以分开...

豆渣锅 2017/12/14 0 0

## 测试看你 Perl 水平处于那个阶段

本文转贴于：<http://www.php-oa.com/2010/09/02/perl-test.html> 我写 Perl 一些日子了，自己有幸和大师们学习和交流 Perl，非常感谢这些人的指点，特别要感谢对我 Perl 指点最多的大师。因为...

红薯 2010/09/08 2.2K 11

## 11.1 -11.5LAMP架构介绍, MySQL,MariaDB介绍, MySQL安装



大家都在搜...



下载APP

开源软件

问答

动弹

博客

翻译

资讯

码云

代码托管

更多

众包

注册

Linux\_老吴 2018/07/28 0 0

加载更多

## OSCHINA 社区

关于我们  
联系我们  
合作伙伴  
Open API

## 微信公众号



## 在线工具

码云 Gitee.com  
企业研发管理  
CopyCat-代码克隆检测  
实用在线工具



## OSCHINA APP

聚合全网技术文章，根据你的阅读喜好进行个性推荐

下载 APP