Perl 标量 Perl 哈希

Perl 数组

Perl 数组一个是存储标量值的列表变量，变量可以是不同类型。

数组变量以 @ 开头。访问数组元素使用 **$ + 变量名称 + [索引值]** 格式来读取，实例如下：

实例

```
#!/usr/bin/perl
@hits = (25, 30, 40);
@names = ("google", "runoob", "taobao");
print "\$hits[0] = $hits[0]\n";
print "\$hits[1] = $hits[1]\n";
print "\$hits[2] = $hits[2]\n";
print "\$names[0] = $names[0]\n";
print "\$names[1] = $names[1]\n";
print "\$names[2] = $names[2]\n";
```

程序中 $ 符号使用了 \ 来转义，让他原样输出。

执行以上程序，输出结果为：

```
$hits[0] = 25
$hits[1] = 30
$hits[2] = 40
$names[0] = google
$names[1] = runoob
$names[2] = taobao
```

创建数组

数组变量以 **@** 符号开始，元素放在括号内，也可以以 **qw** 开始定义数组。

```
@array = (1, 2, 'Hello');
@array = qw/这是 一个 数组/;
```

第二个数组使用 **qw//** 运算符，它返回字符串列表，数组元素以空格分隔。当然也可以使用多行来定义数组：

```
@days = qw/google
taobao
...
runoob/;
```

你也可以按索引来给数组赋值，如下所示：

```
$array[0] = 'Monday';
...
$array[6] = 'Sunday';
```

访问数组元素

访问数组元素使用 **$ + 变量名称 + [索引值]** 格式来读取，实例如下：

实例

```
@sites = qw/google taobao runoob/;
print "$sites[0]\n";
print "$sites[1]\n";
print "$sites[2]\n";
print "$sites[-1]\n"; # 负数，反向读取
```

执行以上程序，输出结果为：

```
google
taobao
runoob
runoob
```

数组索引值从 0 开始，即 0 为第一个元素，1 为第二个元素，以此类推。

负数从反向开始读取，-1 为第一个元素， -2 为第二个元素

数组序列号

Perl 提供了可以按序列输出的数组形式，格式为 **起始值 + .. + 结束值**，实例如下：

实例

```
#!/usr/bin/perl
@var_10 = (1..10);
@var_20 = (10..20);
@var_abc = (a..z);
print "@var_10\n"; # 输出 1 到 10
print "@var_20\n"; # 输出 10 到 20
print "@var_abc\n"; # 输出 a 到 z
```

执行以上程序，输出结果为：

```
1 2 3 4 5 6 7 8 9 10
10 11 12 13 14 15 16 17 18 19 20
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

数组大小

数组大小由数组中的标量上下文决定。：

```
@array = (1,2,3);
print "数组大小: ",标量 @array,"\n";
```

数组长度返回的是数组物理大小，而不是元素的个数，我们可以看以下实例：

实例

```
#!/uer/bin/perl
@array = (1,2,3);
$array[50] = 4;
$size = @array;
$max_index = $#array;
print "数组大小: $size\n";
print "最大索引: $max_index\n";
```

执行以上程序，输出结果为：

```
数组大小:  51
最大索引: 50
```

从输出的结果可以看出，数组元素只有四个，但是数组大小为 51。

添加和删除数组元素

Perl 提供了一些有用的函数来添加和删除数组元素。

如果你之前没有编程经验，可能会问什么是函数，其实我们之前使用的 **print** 即是一个输出函数。

下表列出了数组中常用的操作函数：

序号	类型和描述
1	**push @ARRAY, LIST** 将列表的值放到数组的末尾
2	**pop @ARRAY** 弹出数组最后一个值，并返回它
3	**shift @ARRAY** 弹出数组第一个值，并返回它。数组的索引值也依次减一。
4	**unshift @ARRAY, LIST** 将列表放在数组前面，并返回新数组的元素个数。

实例

```
#!/usr/bin/perl
# 创建一个简单是数组
```

```
@sites = ("google","runoob","taobao");
print "1. \@sites = @sites\n";
# 在数组结尾添加一个元素
push(@sites, "baidu");
print "2. \@sites = @sites\n";
# 在数组开头添加一个元素
unshift(@sites, "weibo");
print "3. \@sites = @sites\n";
# 删除数组末尾的元素
pop(@sites);
print "4. \@sites = @sites\n";
# 移除数组开头的元素
shift(@sites);
print "5. \@sites = @sites\n";
```

执行以上程序，输出结果为：

```
1. @sites  = google runoob taobao
2. @sites  = google runoob taobao baidu
3. @sites  = weibo google runoob taobao baidu
4. @sites  = weibo google runoob taobao
5. @sites  = google runoob taobao
```

切割数组

我们可以切割一个数组，并返回切割后的新数组：

实例

```
#!/usr/bin/perl
@sites = qw/google taobao runoob weibo qq facebook 网易/;
@sites2 = @sites[3,4,5];
print "@sites2\n";
```

执行以上程序，输出结果为：

```
weibo qq facebook
```

数组索引需要指定有效的索引值，可以是正数后负数，每个索引值使用逗号隔开。

如果是连续的索引，可以使用 .. 来表示指定范围：

实例

```
#!/usr/bin/perl
@sites = qw/google taobao runoob weibo qq facebook 网易/;
@sites2 = @sites[3..5];
print "@sites2\n";
```

执行以上程序，输出结果为：

```
weibo qq facebook
```

替换数组元素

Perl 中数组元素替换使用 splice() 函数，语法格式如下：

```
splice @ARRAY, OFFSET [ , LENGTH [ , LIST ] ]
```

参数说明：

- @ARRAY：要替换的数组。
- OFFSET：起始位置。
- LENGTH：替换的元素个数。
- LIST：替换元素列表。

以下实例从第6个元素开始替换数组中的5个元素:

实例

```
#!/usr/bin/perl
@nums = (1..20);
print "替换前 - @nums\n";
splice(@nums, 5, 5, 21..25);
print "替换后 - @nums\n";
```

执行以上程序，输出结果为：

```
替换前 - 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
替换后 - 1 2 3 4 5 21 22 23 24 25 11 12 13 14 15 16 17 18 19 20
```

将字符串转换为数组

Perl 中将字符串转换为数组使用 split() 函数，语法格式如下：

```
split [ PATTERN [ , EXPR [ , LIMIT ] ] ]
```

参数说明：

- PATTERN：分隔符，默认为空格。
- EXPR：指定字符串数。
- LIMIT：如果指定该参数，则返回该数组的元素个数。

实例

```
#!/usr/bin/perl
# 定义字符串
$var_test = "runoob";
```

```
$var_string = "www-runoob-com";
$var_names = "google,taobao,runoob,weibo";
# 字符串转为数组
@test = split('', $var_test);
@string = split('-', $var_string);
@names = split(',', $var_names);
print "$test[3]\n"; # 输出 o
print "$string[2]\n"; # 输出 com
print "$names[3]\n"; # 输出 weibo
```

执行以上程序，输出结果为：

```
o
com
weibo
```

将数组转换为字符串

Perl 中将数组转换为字符串使用 join() 函数，语法格式如下：

```
join EXPR, LIST
```

参数说明：

- EXPR：连接符。
- LIST：列表或数组。

实例

```
#!/usr/bin/perl
# 定义字符串
$var_string = "www-runoob-com";
$var_names = "google,taobao,runoob,weibo";
# 字符串转为数组
@string = split('-', $var_string);
@names = split(',', $var_names);
# 数组转为字符串
$string1 = join( '-', @string );
$string2 = join( ',', @names );
print "$string1\n";
print "$string2\n";
```

执行以上程序，输出结果为：

```
www-runoob-com
google,taobao,runoob,weibo
```

数组排序

Perl 中数组排序使用 sort() 函数，语法格式如下：

```
sort [ SUBROUTINE ] LIST
```

参数说明：

- SUBROUTINE：指定规则。
- LIST：列表或数组。

实例

```
#!/usr/bin/perl
# 定义数组
@sites = qw(google taobao runoob facebook);
print "排序前: @sites\n";
# 对数组进行排序
@sites = sort(@sites);
print "排序后: @sites\n";
```

执行以上程序，输出结果为：

```
排序前: google taobao runoob facebook
排序后: facebook google runoob taobao
```

注意：数组排序是根据 ASCII 数字值来排序。所以我们在对数组进行排序时最好先将每个元素转换为小写后再排序。

特殊变量： $[

特殊变量 **$[** 表示数组的第一索引值，一般都为 0 ，如果我们将 **$[** 设置为 1，则数组的第一个索引值即为 1，第二个为 2，以此类推。实例如下：

实例

```
#!/usr/bin/perl
# 定义数组
@sites = qw(google taobao runoob facebook);
print "网站: @sites\n";
# 设置数组的第一个索引为 1
$[ = 1;
print "\@sites[1]: $sites[1]\n";
print "\@sites[2]: $sites[2]\n";
```

执行以上程序，输出结果为：

```
网站: google taobao runoob facebook
@sites[1]: google
@sites[2]: taobao
```

*一般情况我们不建议使用特殊变量 **$[**，在新版 Perl 中，该变量已废弃。*

合并数组

数组的元素是以逗号来分割，我们也可以使用逗号来合并数组，如下所示：

实例

```
#!/usr/bin/perl
@numbers = (1,3,(4,5,6));
print "numbers = @numbers\n";
```

执行以上程序，输出结果为：

```
numbers = 1 3 4 5 6
```

也可以在数组中嵌入多个数组，并合并到主数组中：

实例

```
#!/usr/bin/perl
@odd = (1,3,5);
@even = (2, 4, 6);
@numbers = (@odd, @even);
print "numbers = @numbers\n";
```

执行以上程序，输出结果为：

```
numbers = 1 3 5 2 4 6
```

从列表中选择元素

一个列表可以当作一个数组使用，在列表后指定索引值可以读取指定的元素，如下所示：

实例

```
#!/usr/bin/perl
$var = (5,4,3,2,1)[4];
print "var 的值为 = $var\n"
```

执行以上程序，输出结果为：

```
var 的值为 = 1
```

同样我们可以在数组中使用 .. 来读取指定范围的元素：

实例

```
#!/usr/bin/perl
@list = (5,4,3,2,1)[1..3];
print "list 的值 = @list\n";
```

执行以上程序，输出结果为：

```
list 的值 = 4 3 2
```

← Perl 标量 Perl 哈希 →

2 篇笔记 写笔记

splice()函数小结

1. 替换数组元素。

```
splice(@ARRAY,OFFSET,LENGTH,LIST)
```

- @ARRAY：要替换的数组。
- OFFSET：起始位置。
- LENGTH：替换的元素个数。
- LIST：替换元素列表。

示例：

```
@a = (1..5);
@b = (a..h);
print "原始  @a\n";
splice(@a , 2 , 2 , @b);
print "插入  @a\n";
```

结果：

```
原始  1 2 3 4 5
插入  1 2 a b c d e f g h 5
```

2. 删除。

```
splice(@ARRAY,OFFSET,LENGTH)
```

- @ARRAY：要删除的数组。
- OFFSET：起始位置。
- LENGTH：删除的元素个数。

示例：

```
@a = (1..20);
print "原始  @a\n";
splice(@a , 2 , 6);
print "删除  @a\n";
```

结果：

```
原始  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
删除  1 2 9 10 11 12 13 14 15 16 17 18 19 20
```

3. 删除到末尾。

```
splice(@ARRAY,OFFSET)
```

- @ARRAY：要删除到结尾的数组。
- OFFSET：起始位置。

示例：

```
@a = (1..20);
print "原始  @a\n";
splice(@a , 2);
print "删除  @a\n";
```

结果：

```
原始  1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
删除  1 2
```

zhi_007 1年前 (2017-10-15)

关于数组排序的总结

sort 函数遍历原始数组的每两个元素，把左边的值放入变量 **$a**，右边的值放入变量 **$b**。 然后调用比较函数。如果 **$a** 的内容应该在左边的话， 比较函数会返回 **1**；如果 **$b** 应该在左边的话，返回 **-1**，两者一样的话，返回 **0**。

依据数值大小排序:

```
my @line=qw /1 2 34 9 12 4 3 8 67 53 1 42 1 17 3 0/;
print "sorted numbers: ";

foreach my $item(sort {$a <=> $b} @line){
    print "$item ";
}
```

代码中使用 `<=>` 比较函数，表示依据数值字面大小排序，此为升序，若要降序排序，仅需调换标量 **$a** 和 **$b** 位置即可。

依据第一个字符 ASCII 值排序: 将上述代码中的比较函数改为 **cmp** 函数即可，这也是 perl 默认的比较函数。

tangxuan 1年前 (2018-03-16)