

# Perl 引用

引用就是指针，Perl 引用是一个标量类型可以指向变量、数组、哈希表（也叫关联数组）甚至子程序，可以应用在程序的任何地方。

## 创建引用

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：

```
$scalarref = \ $foo;      # 标量变量引用
$arrayref  = \@ARGV;     # 列表的引用
$hashref   = \%ENV;      # 哈希的引用
$coderef   = \&handler;  # 子过程引用
$globref   = \*foo;      # GLOB句柄引用
```

在数组中我们可以用匿名数组引用，使用 [] 定义：

```
$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],
]
```

在哈希中我们可以用匿名哈希引用，使用 {} 定义：

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

我们也可以创建一个没有子程序名的匿名子程序引用：

```
$coderef = sub { print "Runoob!\n" };
```

## 取消引用

取消引用可以根据不同的类型使用 \$, @ 或 % 来取消，实例如下：

### 实例

```
#!/usr/bin/perl
$var = 10;
# $r 引用 $var 标量
$r = \ $var;
# 输出本地存储的 $r 的变量值
print "$var 为 : ", $$r, "\n";
@var = (1, 2, 3);
# $r 引用 @var 数组
$r = \@var;
# 输出本地存储的 $r 的变量值
print "@var 为: ", @$r, "\n";
%var = ('key1' => 10, 'key2' => 20);
# $r 引用 %var 数组
$r = \%var;
# 输出本地存储的 $r 的变量值
print "%var 为 : ", %$r, "\n";
```

执行以上实例执行结果为：

```
10 为 : 10
1 2 3 为: 123
%var 为 : key110key220
```

如果你不能确定变量类型，你可以使用 **ref** 来判断，返回值列表如下，如果没有以下的值返回 false：

```
SCALAR
ARRAY
HASH
CODE
GLOB
REF
```

实例如下：

### 实例

```
#!/usr/bin/perl
$var = 10;
$r = \ $var;
print "r 的引用类型 : ", ref($r), "\n";
@var = (1, 2, 3);
$r = \@var;
print "r 的引用类型 : ", ref($r), "\n";
%var = ('key1' => 10, 'key2' => 20);
$r = \%var;
print "r 的引用类型 : ", ref($r), "\n";
```

执行以上实例执行结果为：

r 的引用类型 : SCALAR

r 的引用类型 : ARRAY

r 的引用类型 : HASH

## 循环引用

循环引用在两个引用相互包含时出现。你需要小心使用，不然会导致内存泄露，如下实例：

### 实例

```
#!/usr/bin/perl
my $foo = 100;
$foo = \"$foo;
print "Value of foo is : ", $$foo, "\n";
```

执行以上实例执行结果为：

```
Value of foo is : REF(0x9aae38)
```

## 引用函数

函数引用格式: \&

调用引用函数格式: & + 创建的引用名。

实例如下：

### 实例

```
#!/usr/bin/perl
# 函数定义
sub PrintHash{
my (%hash) = @_;
foreach $item (%hash){
print "元素 : $item\n";
}
}

%hash = ('name' => 'runoob', 'age' => 3);
# 创建函数的引用
$cref = \"&PrintHash;
# 使用引用调用函数
&$cref(%hash);
```

执行以上实例执行结果为：

```
元素 : age
元素 : 3
元素 : name
元素 : runoob
```

[← Perl 子程序\(函数\)](#)[Perl 格式化输出 →](#)[✎ 点我分享笔记](#)