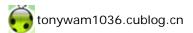
How are you doing?

老王的IT博客

不管现在如何差,总要奋力向前进!



收藏夹 发表文章 管理博客 首页 文章 相册 音乐 博客圈

## perl范围声明our,my,local

老实说,这个东西有点折腾人,主要是书上的表述很晦涩,其实搞懂不难。

our,"把名字限于某个范围",其实就是明确声明一个"全局变量",虽然是在某个模块或者函数里面定义的,外面的也可以访问,如果已经声明

```
过了,再次用"our",表示此处用的是全局的那个,不是同名的私有或者局部变量
our $PROGRAM_NAME = "waiter";
{
my $PROGRAM_NAME = "something";
our $PROGRAM_NAME = "server"; #这里的our和外面的相同,和前句不同。
# 这里调用的代码看到的是"server"
# 这里执行的代码看到的仍然是"server".
my,"把名字和值都限于限于某个范围",简单说,就是只能本层模块或者函数可以看到这个变量,高一层的或者低一层的都看不到的。
sub greeting1{
  my ($hello) = "How are you do?";
  greeting2();
}
sub greeting2{
  print "$hello\n";
$hello = "How are you doing?";
greeting2();
greeting1();
greeting2();
运行结果:
```

```
How are you doing?
How are you doing?
一个 How are you do? 都没有,在greeting1中call greeting2时, greeting2看不到greeting1的私有 $hello变量,只能看到外面的全局
变量$hello
local,"把值局限于某个范围",也有叫"动态词法范围",有点不好懂。我的理解,就是本层和本层下层的函数可以看到本层的变量,但是本层上
一层的不可以。到底范围是多少,不仅取决于本层的函数,还要看下一层的程序长度和深度,所以叫"动态范围"。
sub greeting1{
  local ($hello) = "How are you do?";
  greeting2();
}
sub greeting2{
  print "$hello\n";
$hello = "How are you doing?";
greeting2();
greeting1();
greeting2();
运行结果:
How are you doing?
How are you do?
How are you doing?
跟用 my 时不一样了吧? 此时在greeting1调用greeting2时, greeting2可以看到greeting1的局部变量$hello,外部的全局变量当然就隐藏
了。
```

发表于: 2007-12-13,修改于: 2007-12-14 09:34,已浏览3072次,有评论1条 <u>推荐</u> 投诉

## 网友评论

内容: GOO, THX

本站网友评论于:2008-05-28 00:01:52 (129.174.114.★)

发表评论

用户名: 密码: 免费注册

验证码: **69NH** 匿名