Access Modifiers in Perl

Spoken Tutorial Project
http://spoken-tutorial.org
National Mission on Education through ICT
http://sakshat.ac.in

Nirmala Venkat

27 April 2015

Learning Objective

We will learn about

Learning Objective

We will learn about

- **Scope of variables**

Learning Objective

We will learn about

- **Scope of variables**
- **Private variables**

Learning Objective

We will learn about

- **Scope of variables**
- **Private variables**
- **Dynamically scoped variables**

Learning Objective

We will learn about

- **Scope of variables**
- **Private variables**
- **Dynamically scoped variables**
- **Global Variables**

System Requirements

System Requirements

- **Ubuntu Linux 12.04 OS**

System Requirements

- **Ubuntu Linux 12.04 OS**
- **Perl 5.14.2**

System Requirements

- **Ubuntu Linux 12.04 OS**
- **Perl 5.14.2**
- **gedit Text Editor**

Pre-requisites

- **Basic knowledge of Perl Programming**

Pre-requisites

- **Basic knowledge of Perl Programming**
- **For relevant Perl tutorials, visit http://spoken-tutorial.org**

Scope of variables

Scope of variables

The scope of a variable

Scope of variables

The scope of a variable

- **Is the region of code within which a variable can be accessed**

Scope of variables

The scope of a variable

- **Is the region of code within which a variable can be accessed**
- **Refers to the visibility of variables**

Access Modifiers in Perl

Access Modifiers in Perl

- **my - Private variables**

Access Modifiers in Perl

- **my - Private variables**
- **local - Dynamically scoped variables**

Access Modifiers in Perl

- **my** - Private variables
- **local** - Dynamically scoped variables
- **our** - Global variables

Private variable : my

Private variable : my

- **my variable will lose scope outside the block in which they are declared**

Declaration: my

- **Declare a variable without giving a value**

```
my $fvalue;
```

Declaration: my

- **Declare a variable without giving a value**

```
my $fvalue;
```

- **Declare a variable by assigning a value**

```
my $fvalue = 1;
my $fname = "Rahul";
```

Declaration: my

- Declare a variable without giving a value

```
my $fvalue;
```

- Declare a variable by assigning a value

```
my $fvalue = 1;
my $fname = "Rahul";
```

- Declare several variables

```
my ($fname, $lname, $age );
```

Dynamically scoped variable: local

Dynamically scoped variable: local

- **local keyword gives a temporary scope to a global variable**

Dynamically scoped variable: local

- **local keyword gives a temporary scope to a global variable**
- **The variable is visible to any function called from the original block**

Dynamic Scope: local

- **Declaration:**

```
local $fvalue = 100;
local $fname = "Rakesh";
```

Global variable: our

- **Global variables can be accessed anywhere in the program**

Global variable: our

- **Global variables can be accessed anywhere in the program**
- **Declaration:**

```
our $fvalue = 100;
our $fname = "Priya";
```

Summary

In this tutorial we learnt,

- **Scope of variables**
- **Declaration of private variables**
- **Dynamically scoped variables and**
- **Global variables**

Summary

In this tutorial we learnt,

- Scope of variables
- Declaration of private variables
- Dynamically scoped variables and
- Global variables

Note: It is preferred to use my than local

Assignment

Write the code for the following assignment and execute it

1. **Declare a package as FirstModule**
2. **Declare a variable $age as our and assign the value 42**
3. **Declare another package as SecondModule**

Assignment (cont.)

4. Declare a variable $ageword as our and assign the value "Forty-Two"
5. Declare a subroutine First()
6. Inside the subroutine declare two variables with local and my keyword as below:

```
local $age=52;
my $ageword="Fifty-Two";
```

Assignment (cont.)

7. Call another subroutine as Result()
8. Print the values of $age and $ageword inside this function.
9. End the subroutine
10. Declare the subroutine Result()
11. Again print the values of $age and $ageword
12. End the subroutine

Assignment (cont.)

13. Call the function First()
14. Print the Package First and package Second as below:

```
print "Package First :
$FirstModule::age \n";
print "Package Second :
$SecondModule::ageword \n";
```

About the Spoken Tutorial Project

- Watch the video available at http://spoken-tutorial.org/What_is_a_Spoken_Tutorial
- It summarises the Spoken Tutorial project
- If you do not have good bandwidth, you can download and watch it

Spoken Tutorial Workshops

The Spoken Tutorial Project Team

- **Conducts workshops using spoken tutorials**
- **Gives certificates to those who pass an online test**
- **For more details, please write to contact@spoken-tutorial.org**

Acknowledgements

- Spoken Tutorial Project is a part of the Talk to a Teacher project
- It is supported by the National Mission on Education through ICT, MHRD, Government of India
- More information on this Mission is available at http://spoken-tutorial.org/NMEICT-Intro