

Script Programming with Perl II

15-123

Systems Skills in C and Unix

Regex and Perl

<div class="comment-content">

lst

<p>PBS is more vital now than ever. THE NEWSHOUR WITH JIM LEHRER is the only fair and balanced news program on TV in the United States of America. The reporting is in-depth and all sides of an issue are represented. FRONTLINE and BILL MOYER'S JOURNAL are essential viewing to understanding, in even greater depth, the important issues of our times that are not covered (and sometimes not even mentioned) on cable news shows or the other corporate networks' news programs. PBS is one of the only places where I feel I can go to get the truth.

It should be obvious, but I'm afraid we all forget that commercial television is about the commercials and not about the shows. I love LOST ; DANCING WITH THE STARS and HOUSE are terrific, but when I want to know what's going on, what's really going on, I go to PBS. Try it before you deny it.</p>
<cite>'s Jeff Cummings</cite>

</div>

How do we parse this text?

\$\$

Subroutines

```
sub sum {  
  return $a + $b;  
}
```

global

So we can call this as:

```
$a = 12; $b = 10;
```

```
$sum = sum();
```

```
print "the sum is $sum\n";
```

Passing Arguments

Passing Arguments

A perl subroutine can be called with a list in parenthesis.

Example:

```
sub add {  
    $tmp = 0; # this defines a global variable  
    foreach $_ (@_) {  
        $tmp += $_;  
    }  
    return $tmp;  
}
```

\$d = add(\$a, \$b, \$c);
└───┬───┘
@_

Local variables

Local variables

Perl subroutines can define local private variables.

Example

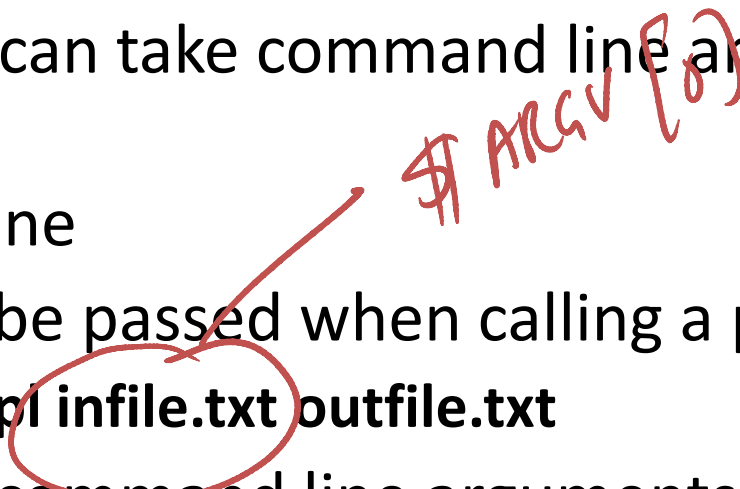
```
sub product {  
  my ($x); # defines the local variable x  
  foreach $_ (@_) { $x *= $_;}  
  return $x;  
}
```

You can have a list of local variables by simply expanding the list as:

```
my ($x, $y, @arr);
```

Command line arguments

Command Line Arguments in Perl

- A Perl program can take command line arguments. One or more command line arguments can be passed when calling a perl program.
 - `perl program.pl infile.txt outfile.txt`
 - The number of command line arguments is given by **`$#ARGV + 1`** and command line arguments are named **`$ARGV[0]`**, **`$ARGV[1]`**, etc
- 

LWP

Library for www in Perl

- LWP contains a collection of Perl modules
 - *use LWP::Simple;*
 - *\$_ = get(\$url);*
 - *print \$_;*
- *Good reference at*
 - *<http://www.perl.com/pub/a/2002/08/20/perlandlwp.html>*

Getopt

- The `Getopt::Long` module implements an extended `getopt` function called `GetOptions()`.
- Command line arguments are given as

– `-n 20` or `-num 20`

– `-n 20 -t test`

perl sl7.pl -n 10 -t test

- *use `Getopt::Long`;*
- *`$images_to_get = 20;`*
- *`$directory = ".";`*
- *`GetOptions("n=i" => \($images_to_get, "t=s" => \($directory));`*

References: <http://perldoc.perl.org/Getopt/Long.html>

Hashes or associative arrays

```
%hash = ( );
```

initializes a hash to empty set. We can
add elements later

```
$hash{'guna'} = "aa";
```

```
$hash{'neil'} = "ab";
```

```
$hash{'george'}="ac";
```

\$
@
%
$$\%H = \{ "guna", 10, "neil", 20 \}$$

each function

- Each function allows us to extract both value and key from a hash table

example

```
%table = {"guna", 10, "me", 20};  
while (($key, $value) = each(%table)) {  
    print "$key => $value\n";  
}
```

References

Creating a Reference to a scalar

```
$num = 10;
```

```
$numref = \u$num;
```

Creating a Reference to an array

```
@array = (guna, me, neil);
```

```
$arrayref = \@array;
```

Creating a Reference to a hashtable

```
%hash = {guna, aa, me, bb, him, cc};
```


```
$hashref = \%hash;
```

perl 5

Dereferences

Dereferencing a Reference to a scalar

```
$num = 10;  
$numref = \ $num;  
print $$numref; # prints the value 10
```



Dereferencing a Reference to an array using -> operator

```
@array = (guna, me, neil);  
$arrayref = \@array;  
print $$arrayref->[0]; # prints 'guna'
```

Dereferencing a Reference to an array using -> operator

```
@array = [guna, me, [blue, red]];  
$arrayref = \@array;  
print $$arrayref->[2][1]; # prints 'red'
```

Systems programming

```
opendir(DIR, ".");  
foreach $file (readdir(DIR)) {  
    print "$file \n";  
}  
close(DIR);
```

Examples

```
print "which directory to change to : ";
chomp($dir = <STDIN>);
if (chdir $dir){
    print "we are now in $dir \n";
} else {
    print "we could not change to $dir \n";
}
```

Modifying Permissions

```
foreach $file ("guna.c", "temp.o") {  
  unless chmod (0666, $file) {  
    warn "could not chmod the file $file \n";  
  }  
}
```

Running a perl script from another

```
#!/usr/local/bin/perl
```

```
system 'perl mkdir.pl file.txt';
```

System 'a.out';

'' _____ ''

Encoding pages

- Example

PAGE: for (\$page=1; ; \$page++){

\$url = "<http://www.cs.cmu.edu/>".\$page;

last PAGE if (\$count == \$maxcount);

}

```
'wget '#URL -O '#keyword'#i'.ip'
```

More Code Examples