Introduction to Perl

Gang Chen
chengang@genomics.cn

October 10, 2015

Outline

Next

What is Perl?

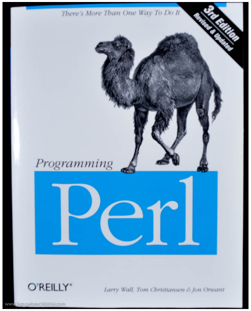
Perl

Perl is a family of high-level, general-purpose, interpreted, dynamic programming languages.

- Practical Extraction and Report Language
- Pathologically Eclectic Rubbish Lister

Perl: History 1

- 1.0: December 18, 1987, Larry Page
- 2.0: 1988, a better regular expression
- 3.0: 1989, support binary data streams
- 4.0, 1991
- Programming Perl, Camel Book, for Perl 4.0


There's More Than One Way To Do It
3rd Edition
Revised & Updated
Programming
Perl
O'REILLY
Larry Wall, Tom Christiansen & Jon Orwant

Perl: History 2

Perl 5

- 5.000: October 17, 1994, rewrite of the interpreter Objects, lexical variables, modules and references are added.
- 5.002: new prototypes feature.
- Comprehensive Perl Archive Network(CPAN), 1995.
- 5.004: May 15, 1997, UNIVERSAL package and CGI.pm module.
- 5.8: July 18, 2002, unicode, a new I/O, thread
- 5.10: Dec 18, 2007
- 5.20: May 27, 2014, subroutin signature, slice.
- 5.22: Jun 1, 2015
- perdoc perlhist

Perl: History 3

Perl 6

- Perl 6 design process was first announced on July 19, 2000
- As of 2015, none of Perl 6 implementation are considered ``complete''.
    - Rakudo Perl: Perl 6 for virtual machines.
    - Pugs: Perl 6 written in Haskell.
    - v6.pm: a pure Perl 5 implementation of Perl 6.
    - Yapsi: a Perl 6 compiler and runtime written in Perl 6 itself.

Perl 5 will be used in the rest of this course.

Applications

Applications

- text processing
- CGI programming: Craigslist, IMDb, Slashdot and so on;
- graphics programming: Perl/Tk, WxPerl
- system administration
- network programming
- bioinformatics

References

Books

- Learning Perl sixth Edition;
- Mastering Perl;
- Advanced Perl;
- Programming Perl;
- perldoc command;

Official Website

http://www.perl.org/

Next

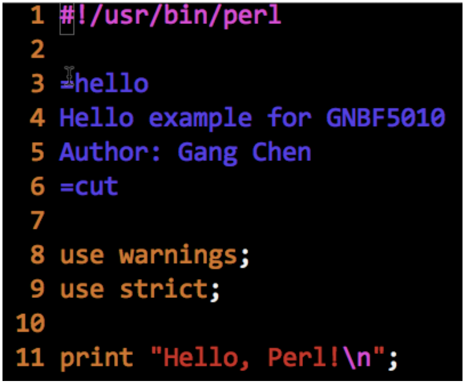
Install

Download and Install

Download: http://www.perl.org/get.html

- Unix/Linux: preinstalled
- Mac OS: preinstalled
- Windows:
  - ActiveState Perl: A binary distribution for Win
  - StrawBerry Perl: Open source
  - DWIM Perl: based on StrawBerry and include many useful CPAN modules

Hello Perl!



```
#!/usr/bin/perl

=hello
Hello example for GNBF5010
Author: Gang Chen
=cut

use warnings;
use strict;

print "Hello, Perl!\n";
```

see hello.pl

Input and Run

1. Input the source codes by using a editor
2. Save the source codes to a file named hello.pl
3. Execute the file:
    - Add execution permission to the file and execute directly
    - Execute the file by using perl interpreter

Just for Fun

Open and execute fun.pl

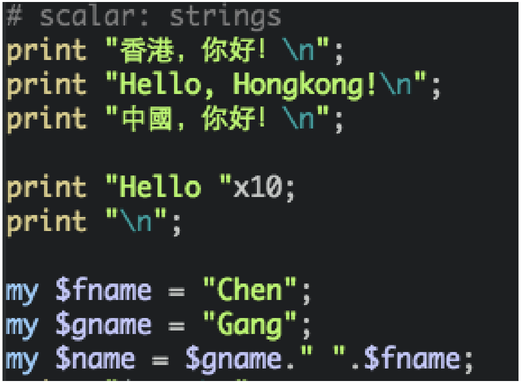
Next

Next

Scalar Data: Number

```
$a = 1;
$b = 1.2;
print $a + $b, "\n";
```

Scalar Data: String



```
# scalar: strings
print "香港, 你好! \n";
print "Hello, Hongkong!\n";
print "中國, 你好! \n";

print "Hello "x10;
print "\n";

my $fname = "Chen";
my $gname = "Gang";
my $name = $gname." ".$fname;
```

Conversion between Numbers and Strings

```
# conversion between numbers and strings
print 1 + 2, "\n";
print "1" + 2, "\n";
print "1" + "2", "\n";
print "1 + 2", "\n";

print "00001" + "002", "\n";

print "one" + 2, "\n";
print "one" + "two", "\n";
```

if Control Structure

```
# if control structure
my $num1 = 5;
my $num2 = 3;

if($num1 > $num2){
  print "Success\n";
}else{
  print "failed\n";
}

$num1 > $num2 ? print "Success\n" : print "Failed\n";

print "Success\n" if ($num1 > $num2);
```

while and for

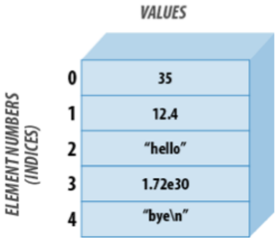
```
# while and for
my $num = 1;
while($num < 10){
  print $num, "\n";
  $num++;
}

for($num = 1;$num<10;$num++){
  print $num, "\n";
}
```

List and Array

List and Array

List	A list is an ordered collection of scalars.
Array	An array is a variable that contains a list.



List and Array

```
# list and array
my @list1 = (1,2,3,4,5);
my @list2 = ("one", "two", "three");
my @list3 = (1..10);
print $list1[0], "\n";
print $list2[1], "\n";
print $list3[3], "\n";
```

List and Array

- Operate to the start of the array: shift, unshift
- Operate to the end of the array: pop, push
- Any place: splice

foreach

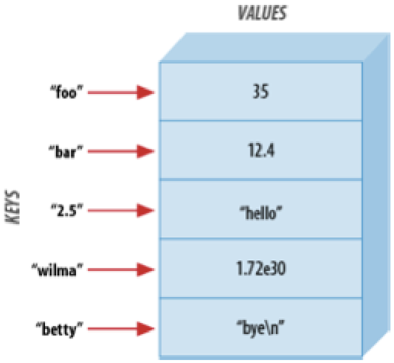
```
foreach (@list1){
  print $_, "\n";
}

for(@list1){
  print $_, "\n";
}

print $_, "\n" for(@list1);

print $_, "\n" for(1..10);
```

Hash



Hash

```
# hash
my %scores = (
  'Gang' => 60,
  'Chen' => 70,
  'Xu' => 80,
  'Lu' => 90,
);

print $scores{'Chen'}, "\n";

for (keys %scores){
  print $_,":",$scores{$_},"\n";
}

print $_,":",$scores{$_},"\n" for (keys %scores);
```

Next

Input from User

```
print "What's your name?\n";

my $name = <STDIN>;
print "Hello ", $name;

my @names = <STDIN>;

print "Hello ", $_ for(@names);
```

Interact with Filesystem

see io.pl

Next

Regular Expression

Regular Expression

Regular Expression is a template or pattern of strings.

Match

see regex.pl

References

- Mastering Regular Expressions
- 精通正则表达式
- 正则指引

Next

Next

Command Options of perl

Options

- -e
- -n

  ```
  while (<>) {
    # your code goes here
  }
  ```

- -p

  ```
    while (<>) {
        # your code goes here
      } continue {
        print or die "-p destination: $!\n";
      }
  ```

Process file content

Adding Line Number to file content

```
perl -ne 'print "$. $_"' names.txt
perl -pe '$_ = "$. $_"' names.txt
```

Next

CGI Programming

CGI

Common Gateway Interface (CGI) is a standard environment for web servers to interface with executable programs installed on a server that generate web pages dynamically.

Thanks!