PERL TUTORIAL

컨텐츠검색플랫폼팀
조영일

소개

- 이름의 유래
  - “Practical Extraction and Report Language”
- 특징
  - Script Language
    - “glue language”
  - 수많은 모듈
    - http://www.CPAN.org
  - 자연어에 가까운 문법
    - 자유로운 문법
    - 표현의 다양성
  - 정규식(Regular Expression)

HELLO, WORLD

- print
  - print “Hello, world₩n”;

CODING STYLE

- #!/usr/bin/env perl

- use strict;
- use warnings;
- use English;

- sub main
- {

- }

- main();

VARIABLE

- scalar
  - my $scalar;
- array or list
  - my @array;
- hash
  - my %hash;
  - * my
    - local variable

VARIABLE

- 꺼내 쓸 때는?
  - $scalar;
  - $array[3];
  - $hash[“key”];

VALUE

- scalar
  - my $phrase = “Howdy, world!₩n”;
  - print $phrase;

VALUE

- array or list
  - my @home = (“couch”, “chair”, “table”, “stove”);
  - my $furniture = shift @home; print $furniture;
  - print $home[0];

VALUE

- hash
  - my %day = (“Sun” => “Sunday”, “Mon” => “Monday”);
  - my $day{“Wed”} = “Wednesday”;
  - print $day{“Tue”};

VALUE

- String
  - “hello₩n”
  - string concatenation operator: .
- Integer
  - octal: 052
  - hexadecimal: 0x20
- Float
  - 0.007
  - 1.02e14

VALUE

- undef
  - unassigned
  - undefined
  - unknown

VALUE

- empty vector
  - ()
  - my @arr = ();
  - my %h = ();

LIST

- initialization
  - my @list1 = (1, 2, 3, 4, 5, 6);
  - my @list2 = 1 .. 6;
  - my @list3 = ‘a’ .. ‘z’;
  - my @list4 = qw(Larry Curly Moe Shemp Joey Kenny);
    - by whiespace
- Operations
  - push @meals, qw(hamburgers pizza lasagna turnip);
  - pop @meals;
  - unshift @meals, qw(tofu curry spanakopita taquitos);
  - shift @meals;

HASH

- initialization
  - my %favorite_flavors = (
    - Gabi=〉 'Mint chocolate chip',
    - Annette =〉 'French vanilla',
  -);
- Operations
  - say "Have Leonardo's address" if exists $addresses{Leonardo} and defined $addresses{Leibniz};
  - for my $addressee (keys %addresses) { }
  - for my $address (values %addresses) { }

CONTROL FLOW

- if
  - if ($str eq “world”) {
  - } elsif ($str ne “hello”) {
  - } else {
  - }
- unless
  - print “not ok” unless ($status == 0);

CONTROL FLOW

- for
  - for (my $i = 0; $i < 10; $i++) {
  - }
  - for my $i (0 .. 9)
- foreach
  - foreach my $item (sort @list) {
  - }
  - for my $item (sort @list)

CONTROL FLOW

- while
  - while (1) { }
  - while (@list) { }
  - while (my $line = <$file_handle>) { }
- until
  - until ($finished_running) { }

CONTROL FLOW

- loop control
  - while (⟨$fh⟩) {
    - next if /₩A#/;
    - ...
  - }

  - while (⟨$fh⟩) {
    - next if /₩A#/;
    - last if /₩A__END__/;
    - ...
  - }

CONTROL FLOW

- loop control
  - while (my $line = ⟨$fh⟩) {
    - chomp $line;
    - # match backslash at the end of a line
    - if ($line =~ s{₩₩$} {}) {
      - $line .= ⟨$fh⟩;
      - redo;
    - }
  - }

OPERATOR

- string
  - eq, ne, gt, ge, lt, le
- logical
  - and, or, not, xor
  - // (defined-or)
- repetition
  - print “H” x 10;

FUNCTION

- definition
  - sub greet_me { … }
- call
  - greet_me(‘hello’, ‘world’);
- parameter
  - sub greet_me
  - {
    - my $first = shift;
    - my $second = shift;
  - }

FUNCTION

- vector-type parameter
  - greet_me(@list);
  - greet_me(%hash);

- but, in reference
  - greet_me(₩@list);
  - greet_me(₩%hash);
  - sub greet_me
  - {
    - my $arg = shift;
    - my @list = @$arg;
    - or
    - my %hash = %$arg;
  - }

FILE

- open(my $infile, “data1.txt”)
- or die “can’t open ‘data1.txt’₩n”;
- open(my $outfile, “〉data2.txt”)
- or die “can’t open ‘data2.txt’₩n”;
- while (my $line = 〈$infile〉) {
  - print $outfile $line;
- }
- close($outfile);
- close($infile);

QUOTATION

- double quote
  - my $quote= qq{"Ouch", he said. "That hurt!"};
- single quote
  - my $reminder = q^Didn't need to escape the single quote!^;
  - my $complaint = q{It's too early to be awake.};

QUOTATION

- multi-line
  - my $text =〈〈‘END_QUOTE’;
  - He looked up. "Time is never on our side, my child. Do you see the irony? All they know is change. Change is the constant on which they all can agree. Whereas we, born out of time to remain perfect and perfectly self-aware, can only suffer change if we pursue it. It is against our nature. We rebel against that change. Shall we consider them greaterfor it?"
  - END_QUOTE