Perl

An Introduction

Special Variables

- `$_`
  - Default source/destination for operators/functions
  - Current line (like awk $0)
  - Default pattern space (similar to sed)

- `ARGV`
  - `$ARGV`
    - Current file
  - `@ARGV`
    - List of command line elements

- `File Handles`
  - `STDIN`
  - `STDOUT`
  - `STDERR`
Opening Non-Standard Files

- `open(HANDLE, "myfilename");`
  - Reading / input
- `open(HANDLE, "> myfilename");`
  - Create / output
- `open(HANDLE, ">> myfilename");`
  - Append / output
- `open(HANDLE, "| program");`
  - Output filter
- `open(HANDLE, "program |");`
  - Input filter

Input Operator < >

- `<STDIN>`
  - Returns the next line from STDIN (keyboard)
- Context sensitive
  - Returns a scalar if used in a scalar context
  - Returns a list if used in an array context
  - `$line = <STDIN>  # next line of input`
  - `@lines = <STDIN>  # all of the rest of input lines`
- `<ARGV>`
  - # process command line
  - `$ARGV  # current command file / arg`
  - `@ARGV  # the rest of the list`
Pattern Matching Operator //

- Returns 1 or " (th null string: two adjacent ')
- /pattern/
- mspatterns
  - s is a non-alphanumeric symbol (except ?)
  - ?pattern? just one match
- options
  - i  case insensitive (like grep)
  - g  global - many times (like vi)
  - o  compile once

Common Operators

- print HANDLE “Hello world
”;
- chop;       # remove new line
- chop $_;     # remove new line
- print $_, “\n”;  # print new line
Lists and Arrays

- A list is an ordered set of scalars
  - Named lists often called arrays
  - List parameters called array values or lists
- Lists/arrays are indexed starting at 0
  - Extract a single single scalar: $ARGV[0]$ or $ARGV[$#ARGV$]

List Functions

- $a = \text{shift(@abz)}$
- $z = \text{pop(@abz)}$
- $\text{unshift(@abz, $a}$
- $\text{push(@abz, $z}$
- $\text{splice(ARRAY, OFFSET, LENTH [, LIST]})$