

Quick and Dirty Perl

What you can do with one line

Using Perl

- Web Applications
- GUI Applications
- Server Maintenance Applications
- Overlooked: Everyday Life!

The Perl Interpreter

- Our new best friend
 - `perl myscript.pl`
- Power of command-line options
 - `-M`: Load a module
 - `-w`: Turn on warnings
 - `-e`: Run some code
- But that's not all!

Run some code

- The `-e` switch allows you to run a piece of Perl code from the command-line
 - `perl -e'print 3856*24, "\n";'`
 - `perl -e'print "$^O\n"'` # see `perlvar`
- Use the right quotes for your shell

The Magic Filehandle

- `<>` - The diamond operator.
- Either `STDIN` or a concatenation of all the files specified on the command-line.
- `cat lol.pl label.pl | perl -e'print while <>'`
is identical to
`perl -e'print while <>' lol.pl label.pl`

The Magic Filehandle

- Also known as ARGV
 - `<>` and `<ARGV>` are identical
 - ARGV is only magical when used with the readline operator (`<...>`), using `*ARGV` may not work as expected.
 - Not to be confused with `@ARGV` and `$ARGV`

Looping Over Input

- -n - Loop over input
while (<>) {
 ...
}
perl -ne'...'
- -p - Loop over input and print
while (<>) {
 ...
}
perl -pe'...'

Looping Over Input

- `perl -pe'' lol.pl`
- `perl -ne'print if /^#/ ' lol.pl`
- `perl -ne'print if 1..5' lol.pl`
- `perl -ne'print if /^=/. / ^=cut/' lol.pl`

Handle Lines

- The `-l` switch automatically handles lines.
 - `chomp` input and add a `\n` to output
 - `perl -ne'chomp; print; print "\n"`
 - and --
 - `perl -nle'print'`
 - are identical

Handle Lines

- Calculate the ratio of comments to code
 - `perl -lne '$comments++ if /^#/; END { print $comments / $. }'`
- Remove HTML tags from some text
 - `perl -lpe 's{<[^\<]*\>}{}g' index.html`

More About \$.

- The current line number of the currently selected filehandle (perldoc perlvar).
- `perl -lpe'print "$.:"' lol.pl label.pl`
- If you only want lines per-file, use `eof()`
- `perl -lne'print "$.: $_"; close ARGV if eof' lol.pl label.pl`

Doing Things

- If you take anything from this: -e, -n/-p, -l
- Combine perl with other system utilities
 - `find /home | perl -lne '$total = (stat $_)[7]; END { print "Total: $total bytes" }`
 - `find | egrep '.html$' | perl -lne '$o = $_; s/\.html$/\.htm/; rename $o, $_'`

Autosplit Fields

- The -a switch autosplits fields
 - Use with -n/-p
 - Splits into @F
- The -F switch lets you specify the delimiter.
 - Default is ' ' (a single space)

Autosplit Fields

- `perl -ane'print pop @F' lol.pl`

- Equivalent to:

```
while (<>) {  
    @F = split " ";  
    print pop @F;  
}
```

Autosplit Fields

- A list of usernames
- `perl -anl -F':' -e'print shift @F' /etc/passwd`
- A list of usernames with valid shells
- `perl -anl -F':' -e'print "$F[0]:$F[-1]" if -e $F[-1] and $F[-1] !~ /false$/' /etc/passwd`

Alter Files Inline

- The `-i` switch allows you to edit data in place
 - Argument to `-i` allows you to specify a backup suffix: `-i'.bak'`
- Clean HTML
 - `perl -pl -i'.bak' -e's/\<[^<]*\>//g' index.html`
- Clean code
 - `perl -nl -i'.bak' -e'print if /^=/ .. /^=cut/'
lol.pl label.pl`

Arguments

- `@ARGV` holds the arguments to your script
 - splits the stuff remaining after `-e` on whitespace
- `perl -le'print +(stat $_)[7], " bytes" for @ARGV' lol.pl index.htm`
- `perl -le'print $total += $_ for @ARGV' | 2
3 4`

Modules

- Where the universe begins
- Check a module's version
 - `perl -MCGI -ple'$_=CGI->VERSION'`
- CPAN
 - `perl -MCPAN -e'install IO::All'`

Modules

- File::Path
 - `perl -MFile::Path -e'mkpath("/some/new/path")'`
- LWP::Simple
 - `perl -MLWP::Simple -e'getprint("http://google.com")' > google.html`

Modules

- HTML::Entities
 - perl -MHTML::Entities -
ple'encode_entities(\$_)' index.html
- File::Slurp -- more efficient than -n/-p
 - perl -MFile::Slurp -e'for (@ARGV) { \$text
= read_file(\$_);

Modules

- File::Find
 - `perl -MFile::Find -e'find(sub { print if /\.pl $/; }, ".")'`

Things I Missed

- Automatic options parsing for your script, see ``perldoc perlrun``
- Slurping with `-0` (zero, not oh)

See Also

- perldoc perlrun and perldoc perlvar
- <http://sial.org/howto/perl/one-liner/>