

Perl Notes

Table of Contents

1. Argument parsing.	1
2. Working with directories.	1
3. Pipe	2

1. Argument parsing

Get and use string / bool from CLI

```
#!/usr/bin/env perl
use warnings;
use strict;
# module
use Getopt::Long;
# variables
my $input = '';
my $output = '';
my $help = 0;
# parse args
GetOptions(
    'u|url=s' => \$input, # string value
    'd|dir=s' => \$output, # string value
    'h|help' => \$help, # bool value
) or die "Error in command line arguments. Use -h for help.\n";
# check args
if ($help || !$input) {
    print_help();
    exit($help ? 0 : 1); # sugar
}
```

2. Working with directories

List of directories

```
sub dirlist {
    my $path = $_[0];
    $path =~ s,/,$,;
    die "Please specify which directory to search" unless -d $path;
    my @dirs;
    opendir(my $dir, $path);
    while (my $entry = readdir $dir) {
        next unless -d $path . '/' . $entry;
        next if $entry eq '.' or $entry eq '..';
        # print "\tdirlist: $entry\n";
        unless ($entry =~ /^.bare/) {
            push(@dirs, $entry)
        }
    }
}
```

```
    return @dirs;
    closedir $dir;
}
```

3. Pipe

Get list of tmux session and pipe it to the fzf command

```
#!/usr/bin/env perl
use warnings;
use strict;
# work
my @tmux_sessions = `tmux list-sessions 2>/dev/null`;
chomp @tmux_sessions;
unless (@tmux_sessions) {
    log_print(4, "tmux: No active session(s)");
    exit 1;
}
my $session_list = join("\n", map { (split /:/)[0] } @tmux_sessions);
my $session_name = `echo "$session_list" | fzf`;
chomp $session_name;
die("No session selected\n") unless $session_name;
```