

CPAN

The Comprehensive Perl
thingie thingie

Tonight's meandering path

- The interactive shell
- Looking around
- Configuring
- Installing modules
- Using CPAN . pm in perl scripts.

The CPAN shell

- `perl -MCPAN -e shell`
- Browse or install based on authors, bundles, distros or modules
- Prefers LWP to access outside world. If not, `Net::FTP`, or as a last resort, `Lynx` or other external command

Looking around

- a, b, d and m can be used to browse authors, bundles, distros and modules
- You can specify either an exact name to search for, or a case insensitive regex-like pattern
- Commands like `ls` and `look` can show more information

Configuring CPAN.pm: MyConfig.pm

- Defaults stored in `Config.pm` found inside perl installed files
- Users can set up
`~/ .cpan/CPAN/MyConfig.pm`
- Session specific configuration can be done within the CPAN shell using `o conf` or `o debug`

Useful Config options

- `makepl_arg`, `make_arg`,
`make_install_arg`: Set to pass
parameters on to the `perl`
`Makefile.pl`, `make` and `make`
`install` processes.
- `urllist`: Lists locations of nearby CPAN
mirrors, or even to specify a non-networked
source.

Installing modules

- `get -> make -> test -> install`
- use `makepl_arg` to set variables like `PREFIX` or `LIB` to install to a non-standard location
- `look` allows you to make changes to a package before building
- Adding `UNINST=1` to `make_install_arg` can help keep packages current
- use `force` to explicitly perform an action

Using CPAN.pm in scripts, an example

```
sub list_modules()
{
    for my $module ( CPAN::Shell->expand( "Module", "/./" )) {
        next unless $module->inst_file;
        print( $module->id, "\t", $module->inst_version, "\t",
            ( $module->uptodate ? ''
              : $module->cpan_version ), "\n" )
        || show_error( "Error processing modules." );
    }
    return;
}
```