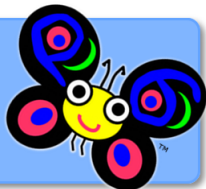


# Introduction To Perl 6 Modules

Jonathan "Duke" Leto

**Perl 6 Modules**



# What is Perl 6?

- It is a specification for a language.
- There are many implementations.
- NOT the successor to Perl 5 (more like a kid sister).

# What are Perl 6 Modules?

- Just like Perl 5 modules, Perl 6 modules are units of distributable and useful code.
- The CPAN of Perl 6 is called <http://modules.perl6.org>
- How many modules does your unreleased language have?

# Which flavor of Perl 6?

- Different flavors of Perl 6 have implemented different feature sets.
- Rakudo Perl 6 currently has the largest feature set and the most number of current contributors.
- Most Perl 6 modules worked on Rakudo at least some time in the past.

# Anatomy of a Perl 6 Module

- It looks just about the same!
- META.info (like a Build.PL or Makefile.PL)
- README\*
- lib/
- t/

# What does META.info look like?

It is just a chunk of JSON with project metadata.

```
{  
  "name"       : "Algorithm::Soundex",  
  "version"    : "*",  
  "description" : "Soundex Algorithm in Perl 6",  
  "depends"     : [],  
  "source-url" : "git://github.com/leto/perl6-Algorithm-Soundex.git"  
}
```

# How Do I Install a Perl 6 Module

You must enlist the help of a panda!



# Installing Panda

First, we grab panda (the cpanminus of Perl 6):

- `git clone git://github.com/tadzik/panda.git`
- `cd panda`
- `sh bootstrap.sh` # needs a perl6 binary in PATH



# Installing a Perl 6 Module with Panda

So simple, even your grandma could do it:

- `panda install Algorithm::Soundex`

# How Do I Start Writing a Perl 6 Module?

- `git clone https://github.com/tadzik/module-starter`
- `cd module-starter`
- `module-starter --description="some junk"`  
Some::Junk

Now you have directory `Some-Junk/` with a `META.info!`

# Show me the code!

This is the essence of all Perl 6 modules

```
1  use v6;  
2  
3  class Algorithm::Soundex {  
4      method soundex($string -> Str) {  
5          # stuff  
6      }  
7  }  
8
```

# Writing Tests for a Perl 6 Module

Very similar to Perl 5 tests:

```
use v6;  
use Test;  
  
# because prove -l doesn't work for perl 6 modules  
BEGIN { push @INC, './lib'; }  
  
plan 3;  
  
use Algorithm::Soundex;  
  
my Algorithm::Soundex $s .= new();  
  
isa_ok($s, Algorithm::Soundex);  
  
pass("Loaded Algorithm::Soundex");  
  
my $soundex = $s.soundex("Robert");  
  
is($soundex, 'R163');
```

# Running Tests for a Perl 6 Module

The easiest way is still prove.

- `prove -e perl6 -rv t/`

# Getting Involved

# Thanks

- Larry
- Eric Wilhelm
- Patrick Michaud
- The Perl Foundation
- Everyone working on Parrot, Rakudo and Perl 6
- PDX.pm for listening to my rants

# Resources

- <http://perl6.org>
- <http://modules.perl6.org>
- TODO: perl 6 planet
- #perl6 on irc.freenode.net
- #parrot on irc.perl.org