Welcome to Charlotte Perl Mongers!

Jason A. Crome Senior Perl Developer, Empowered

Who are we?

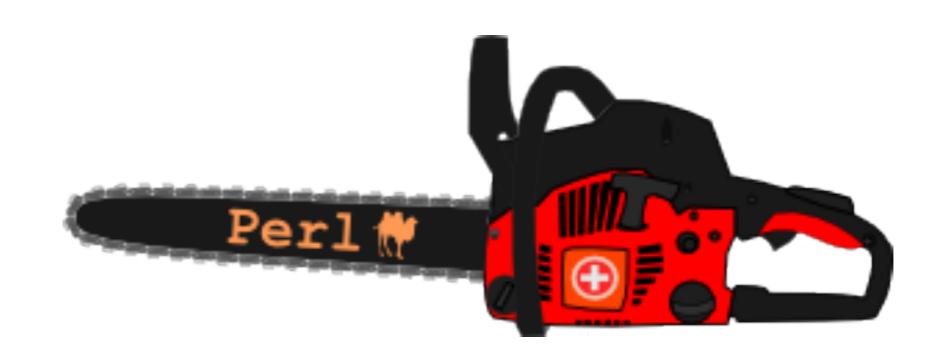
What is Perl Mongers?

- Grassroots organization for promoting Perl
- Run by volunteers (and zealots like me!)
- Share and disseminate information about Perl and the community
- Welcome to Charlotte.pm!

Engage with us!

- Us = all of you!
- Use our Meetup
- charlottedevs.slack.com#perl
- irc.perl.org#charlotte.pm

What is Modern Perl?



What is Modern Perl?

- First-class OO programming
- Great Testing Infrastructure
- Deployment and Configuration Toolchain
- Active user community
- Modern language features
- Modern frameworks

Who's Using Modern Perl?

- ZipRecruiter
- Booking.com
- Craigslist
- Ticketmaster
- Amazon
- BBC
- NASA

- Rent.com
- DuckDuckGo
- IMDB
- Empowered
- ShutterStock
- ServerCentral
- Pinboard

What is Modern Perl?

- First-class OO programming
- Great Testing Infrastructure
- Deployment and Configuration Toolchain
- Active user community
- Modern language features
- Modern frameworks

Object Oriented

- Moose
- Moo
- Mouse

```
package Point;
sub new {
    my ($class, $x, $y) = 0_;
    my \$self = {
         x => $x
        y \Rightarrow \$y
    };
    return bless $self, $class;
sub get x {
    return $[0] -> \{x\};
sub set_x {
    return [0] \rightarrow \{x\} = [1];
sub get y {
    return $ [0] -> {y};
sub set y {
    return [0] -> \{y\} = [1];
1;
```

```
package Point;
use Moose;
has 'x' => (isa => 'Int', is => 'rw', required => 1);
has 'y' => (isa => 'Int', is => 'rw', required => 1);
sub clear {
    my $self = shift;
    self->x(0);
    self->y(0);
package Point3D;
use Moose;
extends 'Point';
has 'z' => (isa => 'Int', is => 'rw', required => 1);
after 'clear' => sub {
    my $self = shift;
    self->z(0);
};
```

Using the Point Class

```
# hash or hashrefs are ok for the constructor my $point1 = Point->new(x => 5, y => 7); my $point2 = Point->new(\{x => 5, y => 7\}); my $point3d = Point3D->new(\{x => 5, y => 42, z => -5);
```

Roles with Moose

```
package MyApp::CLI::SetPassword;
use strictures 2;
use MooseX::App::Command;
extends qw( MyApp::CLI );
with qw(
   MyApp::Role::CLI::WithPassword
    MyApp::Role::CLI::WithUsername
   MyApp::Role::CLI::WithUser
);
use MyApp::Module;
sub run ( $self ) {
    $self-> say( "Changing password for " . $self->username . '...' );
    $self->user->password( $self->password );
    $self->user->update;
    say "Done!";
1;
```

Testing Infrastructure

- Test2
- Test::Class::Moose
- prove
- TAP

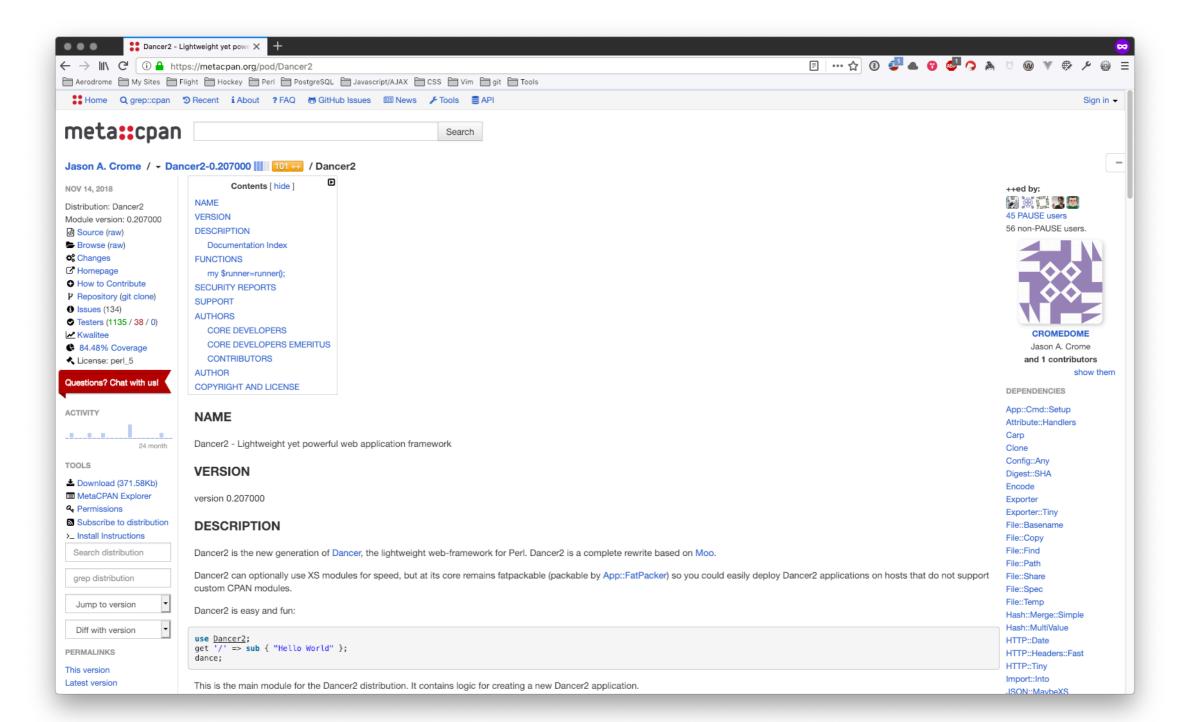
Deployment/Configuration

- Plack/PSGI
- cpanfile
- Carton
- Pinto
- perlbrew
- plenv

Community

- Perl's greatest strength
- Perl Mongers
- YAPC / TPC
- (Meta)CPAN

MetaCPAN



Modern Language Features/Tools

- Named parameters/sub signatures
- Meta-object protocol
- XS
- Pod
- Perlcritic
- Perltidy

Named params/sub signatures

```
sub clear( $self ) {
   self->x(0);
   self->y(0);
after 'clear' => sub( $self ) {
  self->z(0);
};
sub set($self, $x, $y, $z?) {
   self->x(sx);
   $self->y( $y );
   self->z(sz) if z;
```

Modern Frameworks

- Object Frameworks
- ORMs
- Templating Frameworks
- Web App Frameworks

ORMs

• Class::DBI

• DBIx::Class

• Rose::DB

DBIx::Class

```
$user = $schema->resultset('User')->find( { email => $self->email } );
$user = $schema->resultset('User')->create(
    { username => $self->username,
       password => $self->password,
        email => $self->email,
my $user rs = $schema->resultset('User')->search({
        email => { like => '%example.com' },
    },{
        order by => { username => 'ASC' },
```

Templating Frameworks

- HTML::Template
- Mason
- Template Toolkit
- Mustache
- XSlate
- And many, many others...

Web App Frameworks







And now, live coding and swag!



Questions?

Thanks!

https://empowered.net

jcrome@empoweredbenefits.com

Twitter: @cromedome