# Contribution:Dancer

## Perl 5

Carsten Hartenfels

2014-06-23

http://101companies.org/wiki/Contribution:dancer

# Contents

# Contents

1. Perl

# Contents

1. Perl

2. Moops

# Contents

1. Perl

2. Moops

3. Dancer

# Contents

1. Perl
2. Moops
3. Dancer
4. Showcase

# 1. Perl

# 1. Perl

- Scripting Language of the Perl/Ruby/Python Family

# 1. Perl

- Scripting Language of the Perl/Ruby/Python Family

- Multi-Paradigm, TMTOWTDI

# 1. Perl

- Scripting Language of the Perl/Ruby/Python Family
- Multi-Paradigm, TMTOWTDI
- Implementation-Only, No Specification

# 1. Perl

- Scripting Language of the Perl/Ruby/Python Family
- Multi-Paradigm, TMTOWTDI
- Implementation-Only, No Specification
- Extensive, Centralized Module Repository (CPAN)

# 2. Moops

# 2. Moops

- Perl's OO is Orthogonal

# 2. Moops

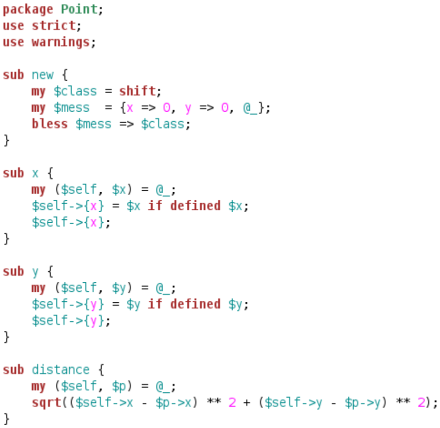
- Perl's OO is Orthogonal, Meaning "Stupid and Verbose"

# 2. Moops

- Perl’s OO is Orthogonal, Meaning “Stupid and Verbose”
- Luckily, Perl is Extensible

# 2. Moops

- Perl's OO is Orthogonal, Meaning "Stupid and Verbose"
- Luckily, Perl is Extensible



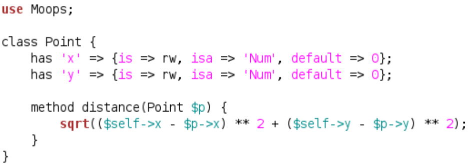
```
package Point;
use strict;
use warnings;

sub new {
    my $class = shift;
    my $mess  = {x => 0, y => 0, @_};
    bless $mess => $class;
}

sub x {
    my ($self, $x) = @_;
    $self->{x} = $x if defined $x;
    $self->{x};
}

sub y {
    my ($self, $y) = @_;
    $self->{y} = $y if defined $y;
    $self->{y};
}

sub distance {
    my ($self, $p) = @_;
    sqrt(($self->x - $p->x) ** 2 + ($self->y - $p->y) ** 2);
}
```



```
use Moops;

class Point {
    has 'x' => {is => rw, isa => 'Num', default => 0};
    has 'y' => {is => rw, isa => 'Num', default => 0};

    method distance(Point $p) {
        sqrt(($self->x - $p->x) ** 2 + ($self->y - $p->y) ** 2);
    }
}
```

# 3. Dancer

# 3. Dancer

- Web Application Framework

# 3. Dancer

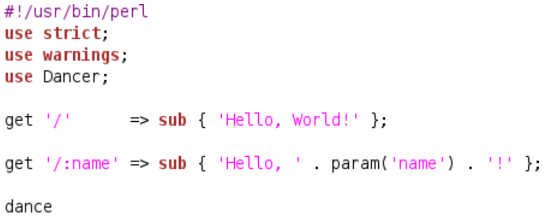
- Web Application Framework
- Inspired by Sinatra, like Bottle, Flask, Spark etc.

# 3. Dancer

- Web Application Framework
- Inspired by Sinatra, like Bottle, Flask, Spark etc.
- Embedded Domain-Specific Language

# 3. Dancer

- Web Application Framework
- Inspired by Sinatra, like Bottle, Flask, Spark etc.
- Embedded Domain-Specific Language



```
#!/usr/bin/perl
use strict;
use warnings;
use Dancer;

get '/'      => sub { 'Hello, World!' };

get '/:name' => sub { 'Hello, ' . param('name') . '!' };

dance
```

# 4. Showcase

Demonstration

# Thank You All For Listening

Visit `http://75.136.142.69:3000/`