Modern Perl

“Perl was always designed to be an Evolving Language”

- **Larry Wall**, Original Perl Language Designer

Language Syntax

Standards and Conventions

Simple Object

```
package HelloName;
use Moose;

has name => (is=> "ro", required=>1);

sub hello {
    my ($self) = @_;
    "Hello ".$self->name;
}

1;
```

Using Your Object

```
#!/usr/bin/env perl

use Modern::Perl;
use HelloName;

my $obj = HelloName->new(name=> "John");
say $obj->hello;
```

Exercise

5, 8, 13, 21, 34

Basic Variables

```
my $name = "John";
my @friends = ("Chris", "Dave", "Joanne");
my %address = (
  street => "510 E 5th",
  state => "NY",
);
```

Scalars

```
my $name = "John";
say $name;
```

Arrays

```
my @friends = ("Chris", "Dave", "Joanne");
say $friends[0];
```

Hashes

```
my %address = (
  street => "510 E 5th",
  state => "NY",
);

say $address{state};
```

Sigils

$ @   %   &   * $#

What are the Values?

```
my @friends = ("Chris", "Dave", "Joanne");
my %address = (
  street => "510 E 5th",
  state => "NY",
);

my @some_friends = @friends[0..1];
my @elements = @address{qw(state ny)};
```

Packages and Scopes

```
package MyApp::Web;

our $website_name = "Awesome Site";
sub main_page { $website_name }
sub child_page {
  my $sub_title = shift;
  main_page. " [$sub_title]";
}
1;
```

Guess the STDOUT

```
use Modern::Perl;
use MyApp::Web;

say $MyApp::Web::website_name;
say MyApp::Web::main_page;
say MyApp::Web::child_page("Contact Us");
```

Variable Scopes

```
my $name = “Vanessa”;
our @friends = qw(mike joe jane);
local $name = “John”;
my $counting = sub { state $cnt = 0; $cnt++ };
```

Control Flow

```
my @friends = qw(mike joe jane);
for my $friend(@friends) {
  say $friend;
}
```

Control Flow

```
say $_ for qw(mike joe jane);
```

Control + Conditional

```
my @friends = (
  {name => ‘mike’, age => 20},
  {name => ‘joe’, age => 34},
);

for my $friend(@friends) {
  say $friend->{name}
    if $friend->{age} > 30;
}
```

Conditionals

```
for my $friend(@friends) {
  if ($friend->{age} > 30) {
    say $friend->{name};
  }
}
```

Logical Operators

&& ||
and   or

Complex Conditionals

```
if (($age > 30) and ($state eq ‘NY’)) {
  say “I am old and live in NY!”;
} elsif ($age > 67) {
 say “I am retired!”;
} else {
  say “Hey, I am still pretty young!”;
}
```

Negation Conditions

```
unless($age > 21) {
  say “I can’t drink :(“;
}

say “I’m not New Yorker”
  unless $state eq ‘NY’;
```

Careful!

```
if($age < 21) {
  say “I can’t drink :(“;
}

say “I’m not New Yorker”
  if $state ne ‘NY’;
```

Indeterminate Loops

```
while ( $work->has_jobs ) {
  $work->do_some_work;
}
```

Exercise Recap

5, 8, 13, 21, 34

Questions?