

Modern Perl

Images Provided
Courtesy of
Shutterstock, LLC

Summary and Next Steps



One Liner

```
perl -MMModern::Perl -ne 'my @hi = ("Hi", $_);  
say @hi'
```

```
John  
Hello John
```

Simple Object

```
package HelloName;  
use Moose;  
  
has name => (is=> "ro", required=>1);  
  
sub hello {  
    my ($self) = @_;  
    "Hello ".$self->name;  
}  
  
1;
```

Simple Test Case

```
use HelloName;  
use Test::Most;
```

```
my $obj = HelloName->new(name=>'John');
```

```
is $obj->name, 'John';
```

```
is $obj->hello, 'Hello John!';
```

```
done_testing;
```

Using Your Object

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

```
use Modern::Perl;
```

```
use HelloName;
```

```
my $obj = HelloName->new(name=> shift);
```

```
say $obj->hello;
```


Simple Website

```
use HelloName;
```

```
my $obj = HelloName->new( name =>  
    $ENV{HelloName_name} );
```

```
sub {  
    return [  
        200, [ "Content-Type" => "text/plain"],  
        [ $obj->hello ],  
    ];  
};
```

Things to Remember

Some Programmer Rule and Style Guidelines

Maintainable Perl

Follow Community Standards

Do the most simple thing that doesn't suck

Avoid excessive Perl Golf

Avoid deprecated idioms

Community Standards

Use the CPAN toolchain

Use CPAN!

Class, Method and Attribute Naming Conventions

Moose, Test::Most, Plack (and look at Task::Kensho)

Don't Suck

```
use HelloName;
```

```
my $obj = HelloName->new( name =>  
    $ENV{HelloName_name} );
```

```
sub {  
    return [  
        200, [ "Content-Type" => "text/plain"],  
        [ $obj->hello ],  
    ];  
};
```

Golfing

```
$_='while(read+STDIN,$_ ,2048){$a=29;$b=73;$c=142;$t=255;@t=map{$_%16or$t^=$c^=( $m=(11,10,116,100,11,122,20,100)[$_/16%8])&110;$t^=(72,@z=(64,72,$a^=12*($_%16 -2?0:$m&17)), $b^=$_%64?12:0,@z)[$_%8]}(16..271);if((@a=unx"C*",$_)[20]&48){$h=5;$_=unxb24,join"",@b=map{xB8,unxb8,chr($_^$a[--$h+84])}@ARGV;s/...$/1$&/;$ d=unxV,xb25,$_;$e=256|(ord$b[4])<<9|ord$b[3];$d=$d8^($f=$t&($d12^$d4^ $d^$d/8))<<17,$e=$e8^($t&($g=($q=$e14&7^$e)^$q*8^$q<<6))<<9,$_=$t[$_]^ (($h=8)+=$f+(~$g&$t))for@a[128..$#a]}print+x"C*",@a}';s/x/pack+/g;eval
```

Deprecated Idioms

```
open (FH, "< file.txt") || die $@;
```

New Hot-ness

```
use autodie;  
open(my $fh, '<', 'file.txt');
```

```
use Path::Class::File;  
my $file = Path::Class::File->new(@dirs, $file);
```


How to Learn More

Places to go and check out!

Books

<http://learn.perl.org/books/>



Search the CPAN

I'm Feeling Lucky

Hint: Press shift and enter if you are feeling lucky

[API](#)

[About MetaCPAN](#)

[Fork metacpan.org](#)

<http://metacpan.org/>

The Web

<http://learn.perl.org/>

<http://blogs.perl.org/>

<https://metacpan.org/recent>

http://szabgab.com/perl_tutorial.html

The Web (part 2)

<http://www.udemy.com/beginner-perl-maven/>

<http://www.udemy.com/advanced-perl-maven/>

IRC

<irc://irc.perl.org>

My Favorite Channels

#moose

#catalyst

#dbix-class

#plack

#web-simple

#formhandler

Our Code + Lessons

[https://github.com/jjn1056/
GeneralAssembly-Guestbook](https://github.com/jjn1056/GeneralAssembly-Guestbook)

“Perl was always
designed to be an
Evolving Language”

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



Questions?