

# Learning Perl

Ed Summers



# The Plan

- What is Perl?
- Perl Basics
- Perl Modules
- Perl and MARC



What  
is  
Perl?



# Everyday Programming

1. Preheat oven to 375° F.
2. Combine flour, baking soda and salt in small bowl.
3. Beat butter, granulated sugar, brown sugar and vanilla extract in large mixer bowl until creamy.
4. Add eggs one at a time, beating well after each addition.
5. Gradually beat in flour mixture.
6. Stir in morsels and nuts.
7. Drop by rounded tablespoon onto ungreased baking sheets.
8. Bake for 9 to 11 minutes or until golden brown.
9. Cool on baking sheets for 2 minutes.



# Programming in Libraries?

- Online Catalogs
- Web Sites
- Homegrown Databases
- Data Services: Census Data, GIS, etc...
- Public Access Computing



There  
are  
lots  
of  
languages

2,500  
at  
last  
count!!



# Popular Languages in Libraries

C

C++

Java

Python

PHP

Visual Basic

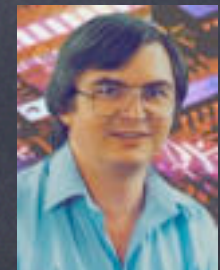
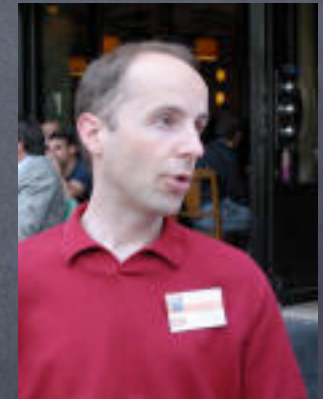
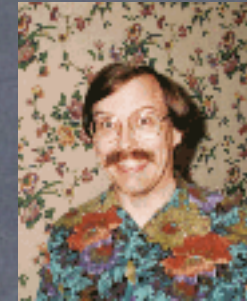
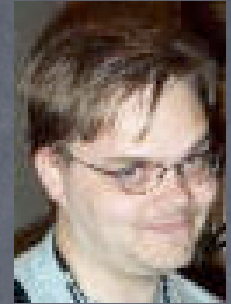
C#

Perl



# Perl

- Created by Larry Wall at JPL in 1987
- Open Source
- C, Lisp, Csh, Pascal, Awk, Sed, Smalltalk
- Practical Extraction and Reporting Language
- ... or Pathologically Eclectic Rubbish Lister





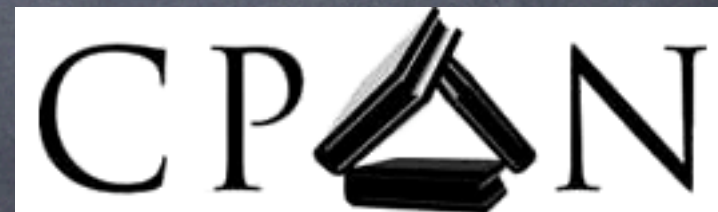
# Natural Languages

- Learn it once, use it many times.
- Learn as you go.
- Many acceptable levels of competence.
- Multiple ways of saying the same thing.
- No shame in borrowing.
- Cooperative design.



# CPAN

- Comprehensive Perl Archive Network
- Thousands of free extensions to Perl available.





# Perl in Libraries

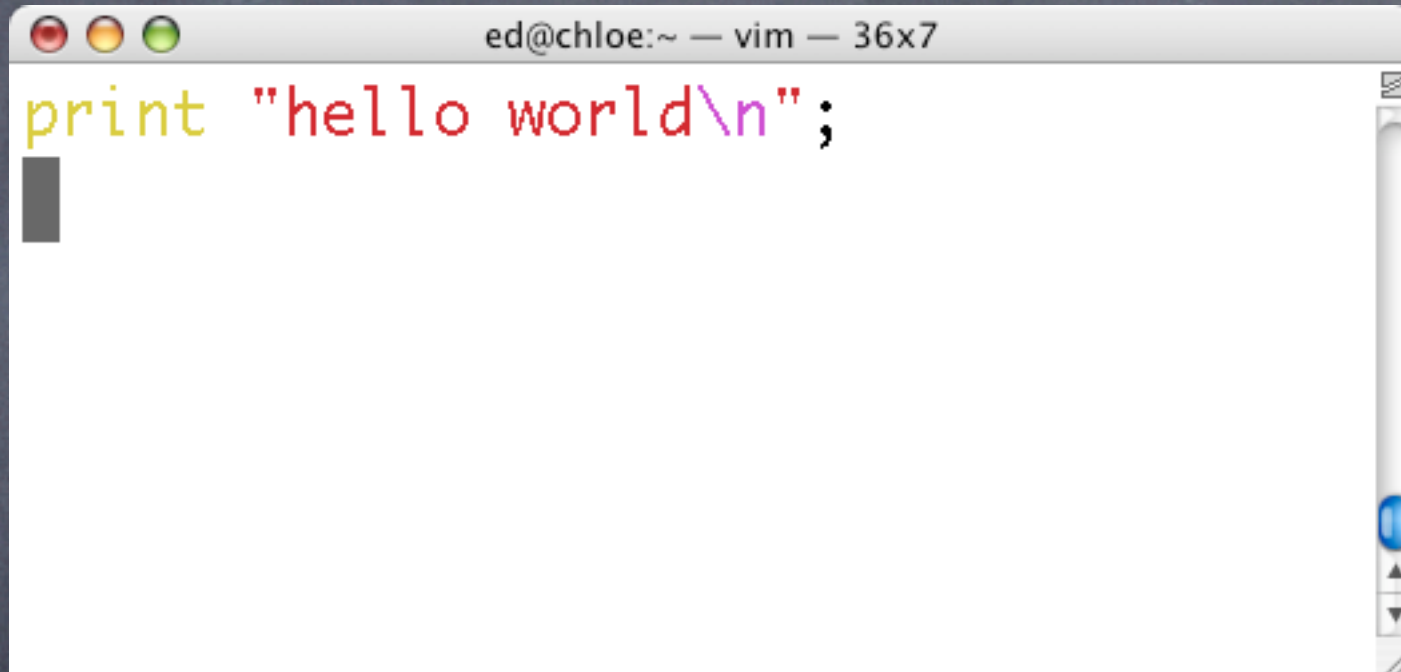
- Web Development
- Homegrown Databases
- Systems Administration:  
email, monitoring tools,  
proxies, log analysis ...
- Data Processing: MARC,  
Z39.50, XML, RDF, Web  
Services ...



# Perl Basics



# Hello World

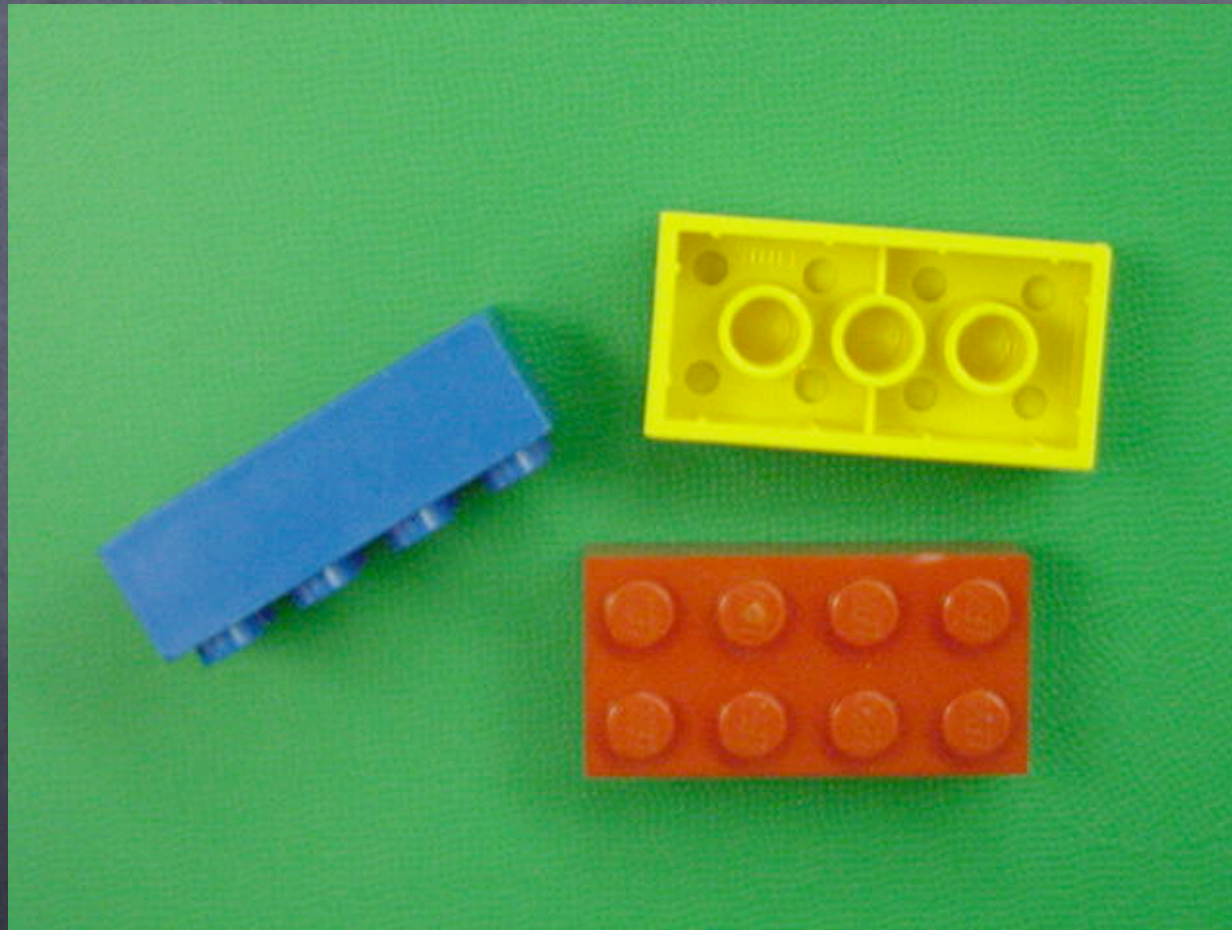


The image shows a terminal window with a title bar that reads "ed@chloe:~ — vim — 36x7". The window contains a single line of C code: `print "hello world\n";`. The code is color-coded: "print" is yellow, "hello world\n" is red, and ";" is purple. A black cursor is positioned at the start of the second line. The window has standard macOS window controls (red, yellow, green buttons) and a scrollbar on the right side.

```
ed@chloe:~ — vim — 36x7
print "hello world\n";
█
```



# Building Blocks





# Variables

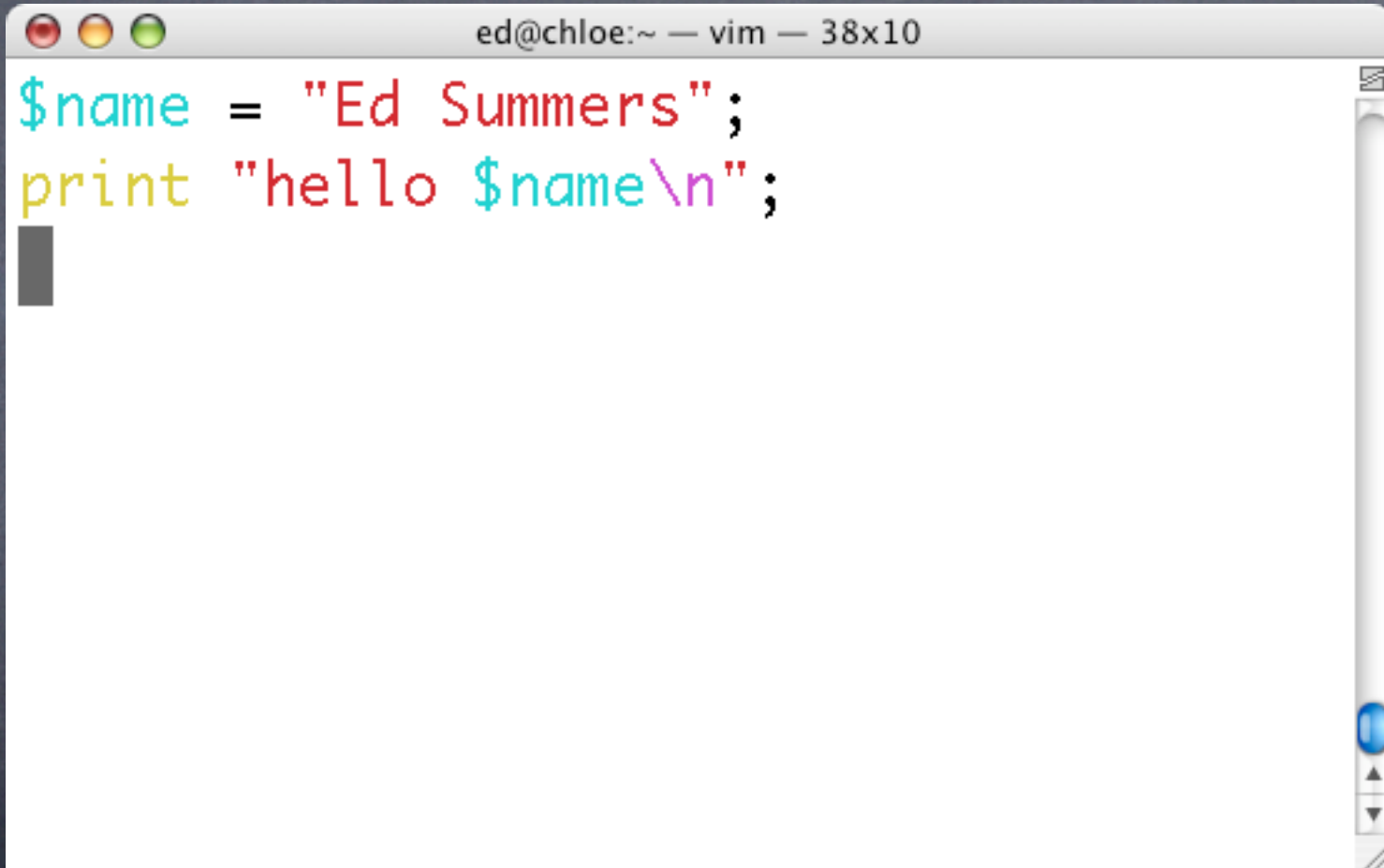
\$

@

%



# \$ Strings \$



```
ed@chloe:~ — vim — 38x10
$name = "Ed Summers";
print "hello $name\n";
```

The image shows a terminal window with a vim editor. The window title is "ed@chloe:~ — vim — 38x10". The editor contains two lines of C code: `$name = "Ed Summers";` and `print "hello $name\n";`. The code is color-coded: `$name` is cyan, `=` is black, `"Ed Summers"` is red, `;` is black, `print` is yellow, `"hello $name\n"` is red, and `;` is black. A black cursor is visible on the first line of the second line of code.



# \$ Numbers \$

```
ed@chloe:~ -- vim -- 37x11
$numerator = 22;
$denominator = 7;
$pi = $numerator/$denominator;

print "pi=$pi\n";
█
```



# @ Arrays @

```
ed@chloe:~ — vim — 37x10
@presidents = (
    "Ford", "Carter", "Reagan",
    "Bush", "Clinton", "Bush"
);
push( @presidents, "Nader" );
print pop( @presidents ), "\n";
print pop( @presidents ), "\n";
print shift( @presidents ), "\n";
█
```



# @ Combining @

```
ed@chloe:~ -- vim -- 37x10
@array1 = ( 1, 2, 3 );
@array2 = ( 4, 5, 6 );
@array3 = ( @array1, @array2 );
print @array3, "\n";
█
```



# % Hashes %

```
ed@chloe:~ -- vim -- 37x10
%dewey = (
    '100' => 'Philosophy',
    '200' => 'Religion',
    '300' => 'Social Sciences',
    '400' => 'Language'
);

print $dewey{ '200' }, "\n";

"example.pl" 18L, 161C
```



# Flow Control





# if

```
ed@chloe:~ — vim — 37x10
$X=1;

if ( $X == 1 ) {
    print "x is equal to 1\n";
}
█
```

# if/else

```
ed@chloe:~ — vim — 37x10
$x=2;

if ( $x == 1 ) {
    print "x is equal to 1\n";
}
else {
    print "x is not equal to 1\n";
}
█
```



# input

```
ed@chloe:~ — vim — 37x10
$x=<STDIN>;

if ( $x == 1 ) {
    print "x is equal to 1\n";
}
else {
    print "x is not equal to 1\n";
}
█
```

# while

```
ed@chloe:~ — screen — 37x10
#!/usr/bin/perl

$year = 1900;
while ( $year < 2004 ) {
    print "$year\n";
    $year = $year + 1;
}

~
```

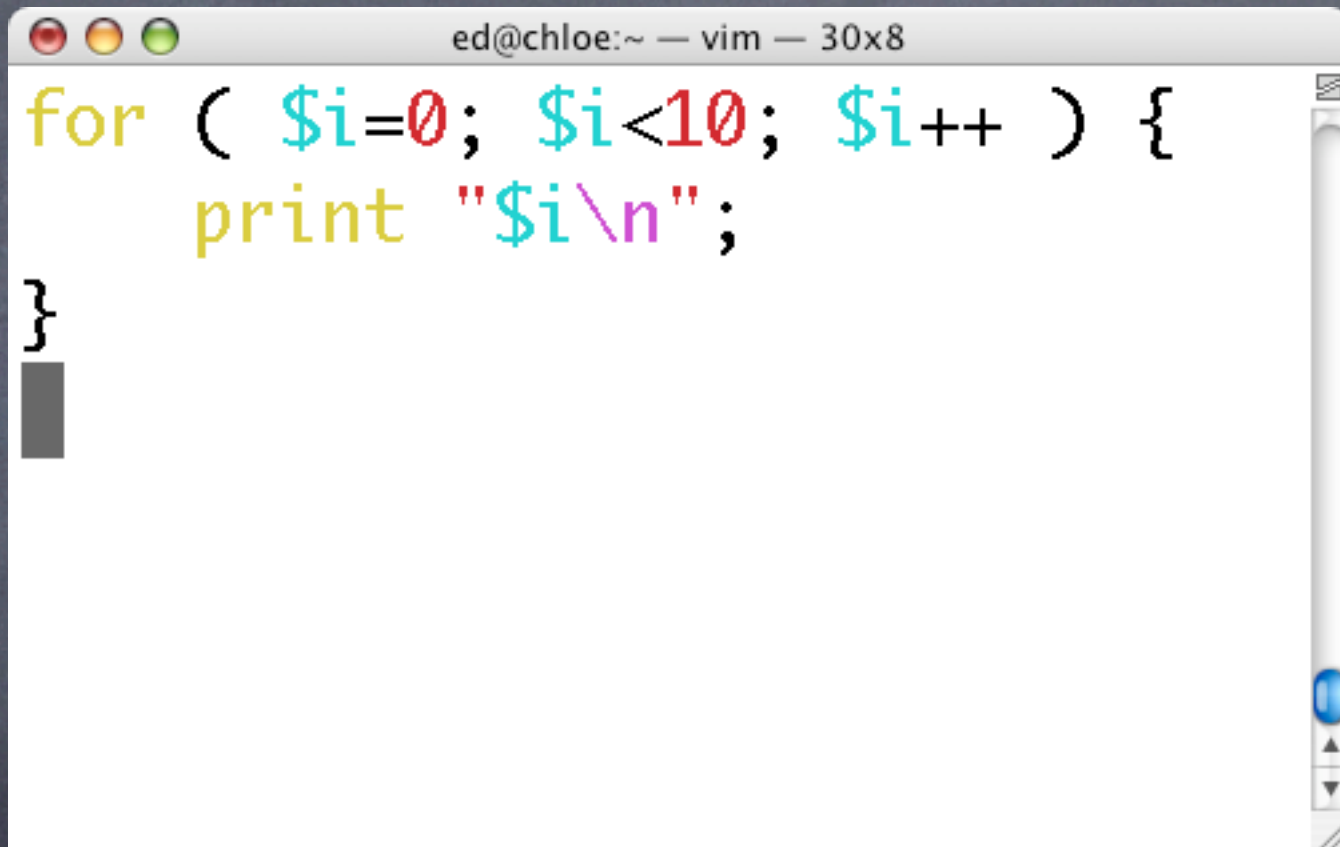


# foreach

```
ed@chloe:~ — screen — 36x11
#!/usr/bin/perl

@presidents = (
    "Ford", "Carter", "Reagan",
    "Bush", "Clinton", "Bush"
);
foreach $name ( @presidents ) {
    print "$name\n";
}
█
```

# for

A screenshot of a terminal window with a light gray title bar. The title bar contains three colored window control buttons (red, yellow, green) on the left and the text "ed@chloe:~ — vim — 30x8" on the right. The main area of the window is white and contains C code. The code is: 

```
for ( $i=0; $i<10; $i++ ) {  
    print "$i\n";  
}
```

 The code is color-coded: "for" is yellow, "(" is black, "\$i=0" is cyan, ";" is black, "\$i<10" is cyan, ";" is black, "\$i++" is cyan, ")" is black, "{" is black, "print" is yellow, "\$i\n" is cyan, ";" is black, and "}" is black. A black cursor is positioned at the end of the first line. On the right side of the window, there is a vertical scrollbar with a blue slider and a small icon at the top.



# subroutines

```
ed@chloe:~ — vim — 30x10
print convert( 57 ), "\n";
print convert( 79 ), "\n";

sub convert {
    $f = shift;
    $c = ($f-32) * 5/9;
    return $c;
}
█
```

```
ed@chloe:~ — vim — 23x8
sub convert {
    $f = shift;
    $c = ($f-32) * 5/9;
    return $c;
}
1;
```

require

```
ed@chloe:~ — vim — 24x8
require( "convert.pl" );
print convert(55), "\n";
```



# modules



The image shows a screenshot of a vim editor window. The title bar at the top reads "ed@chloe:~ — vim — 41x8". The editor contains the following Perl code:

```
use LWP::Simple;

$html = get( "http://www.google.com" );
print $html;
```

The code is color-coded: "use" is yellow, "LWP::Simple;" is black, "\$html" is cyan, "get(" is black, "http://www.google.com" is red, and "print \$html;" is cyan. There is a grey cursor block on the line following "print \$html;". At the bottom left, there are two blue question marks. On the right side, there is a vertical scrollbar and a blue play button icon.

# objects

```
ed@chloe:~ — vim — 38x9
use Net::FTP;
$ftp = Net::FTP->new( 'ftp.loc.gov' );
$ftp->login( 'anonymous' );
$ftp->cwd( 'pub' );
@files = $ftp->dir();
foreach my $file ( @files ) {
    print "$file\n";
}
```



# combing some stuff

```
ed@chloe:~ -- vim -- 48x11
use Net::Amazon;
$client = Net::Amazon->new(
    token => 'D3RG80HKRPMZC1' );
$response = $client->search(
    mode => 'music',
    artist => 'Thelonious Monk' );
foreach $property ( $response->properties() ) {
    print $property->album(), "\n";
}
```



