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

C
Cobol
Scheme
Tcl/Tk
Lisp
Simula
Rexx
Ada
Java
LOGO
Perl
SML
Pascal
Basic
POSTSCRIPT
Algol
C++
APL
PL/I
Fortran
SmallTalk
Modula

There
are
lots
of
languages

2,500
at
last
count!!

Popular Languages in Libraries

C

C++

Java

Python

PHP

Visual Basic

C#

Perl

Perl

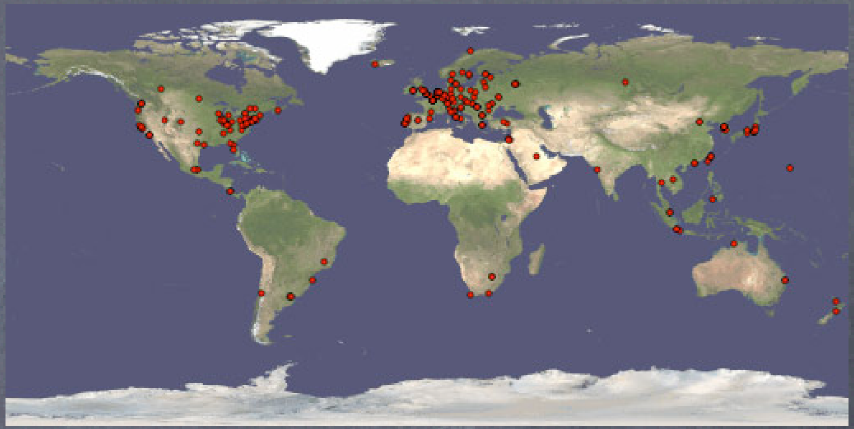
- Created by Larry Wall at JPL in 1987
- Open Source
- C, Lisp, Csh, Pascal, Awk, Sed, Smalltalk
- Practical Extraction and Reporting Language
- ... or Pathologically Eclectic Rubbish Lister

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.

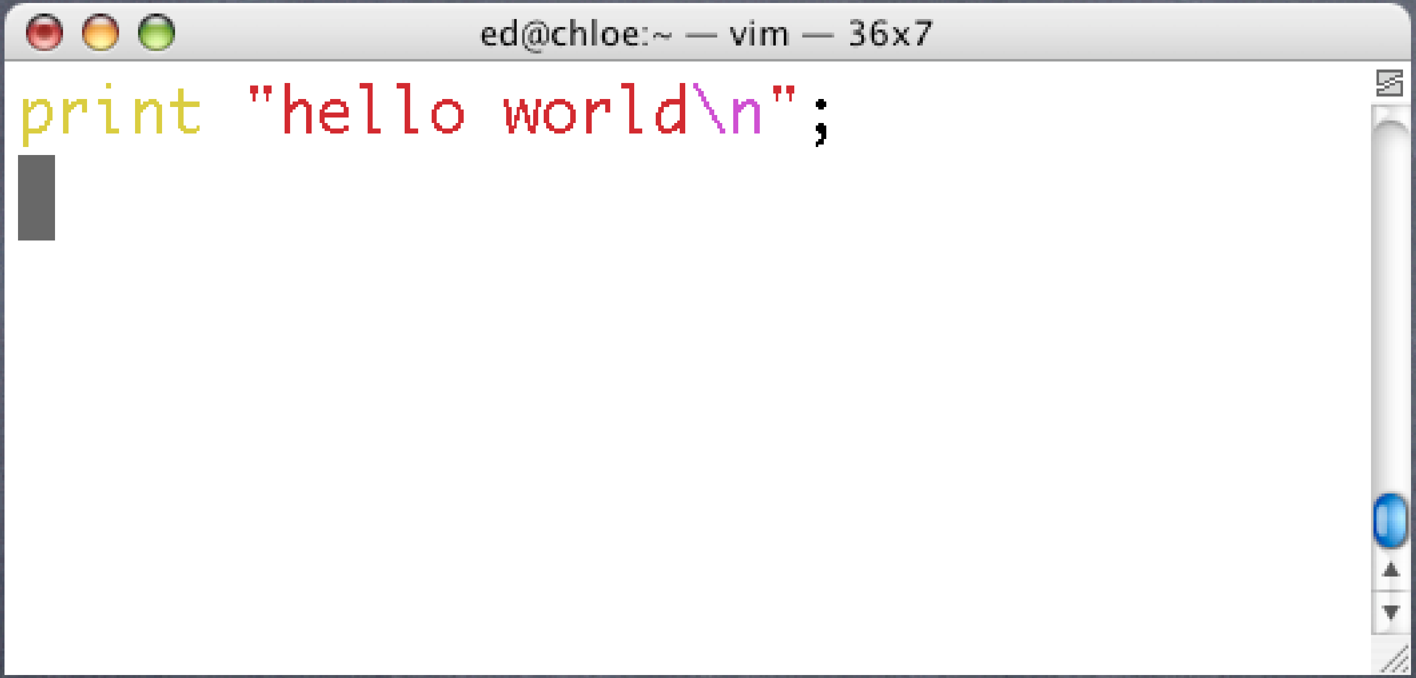


Perl in Libraries

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

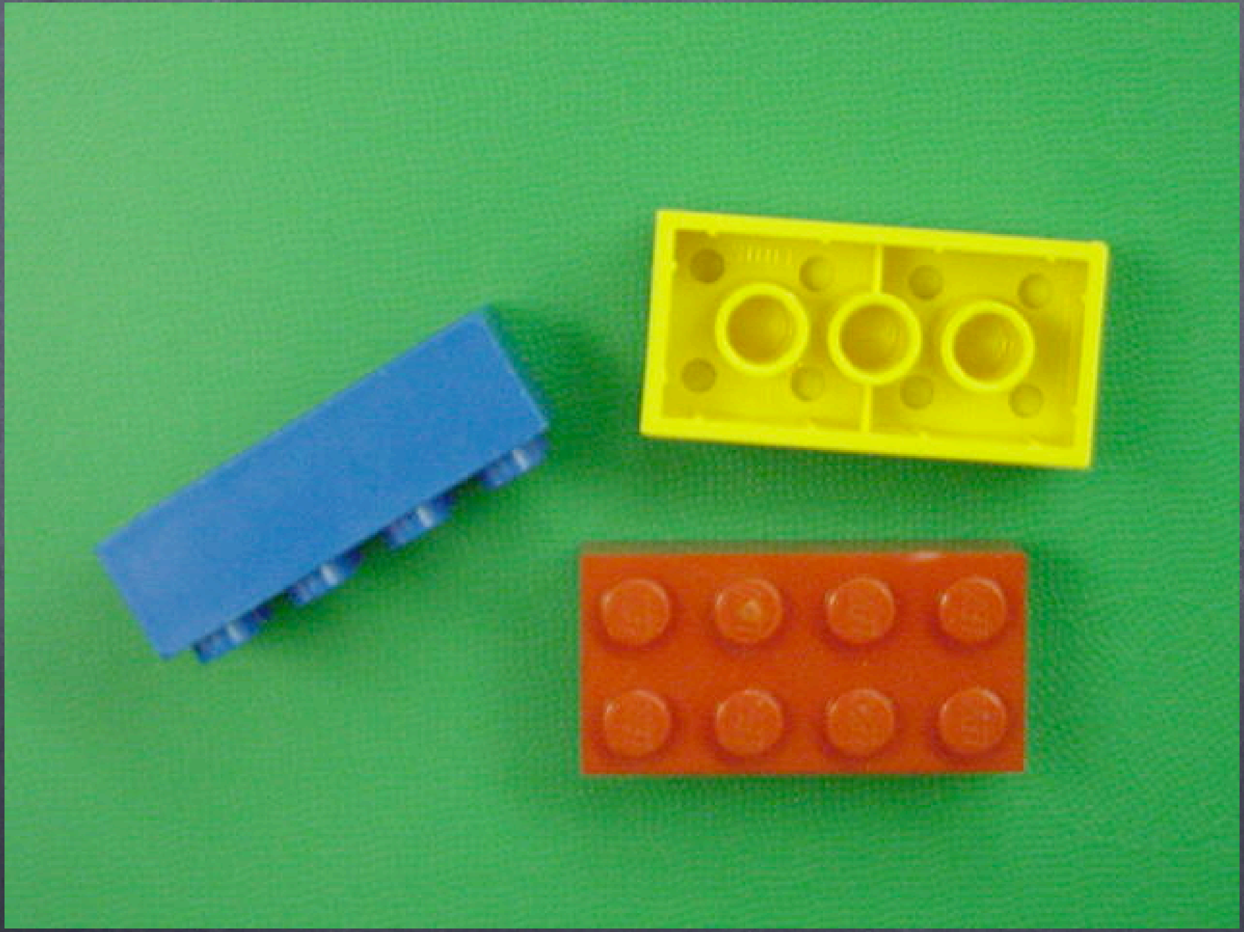
Perl
Basics

Hello World



```
print "hello world\n";
```

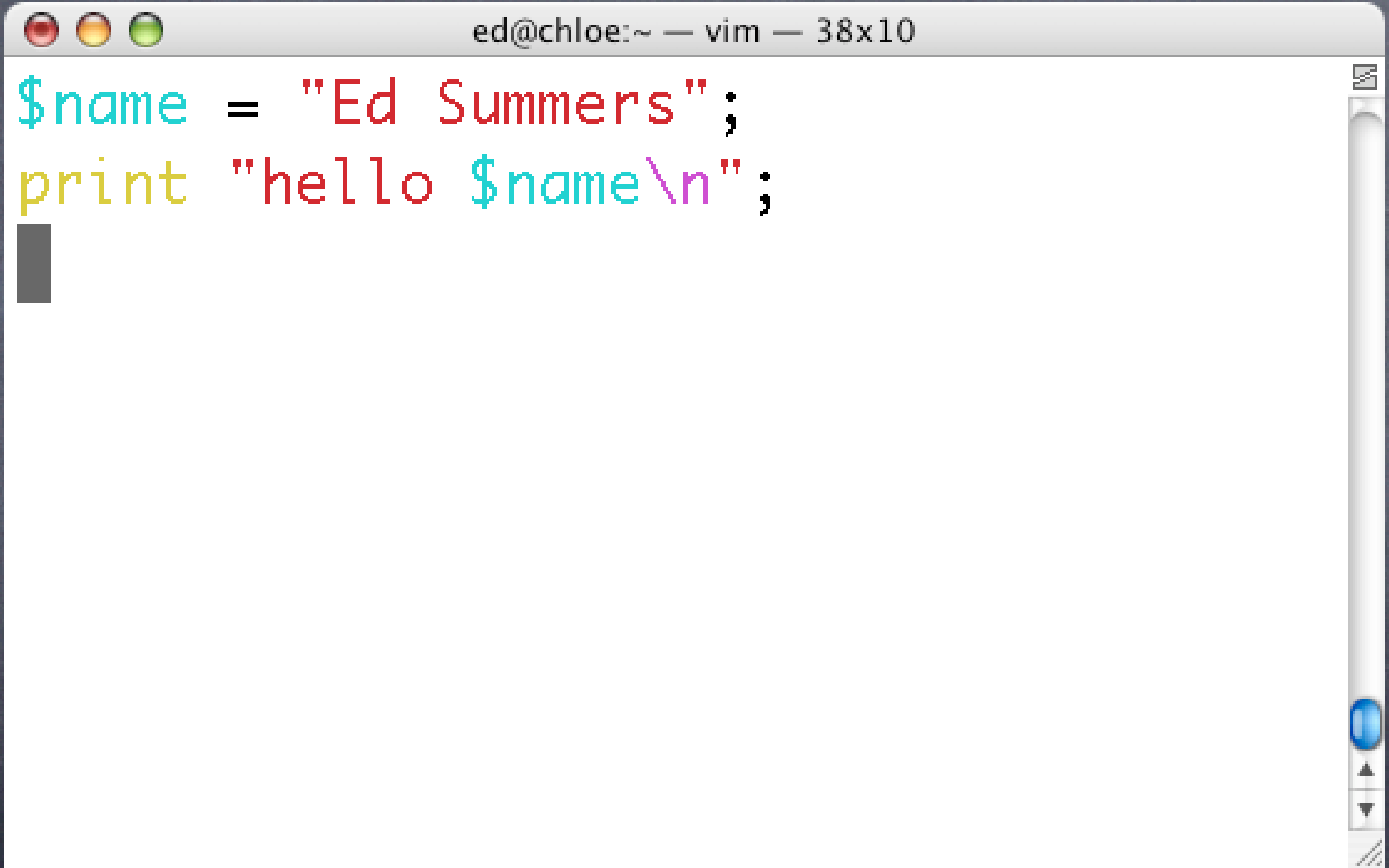
Building Blocks



Variables

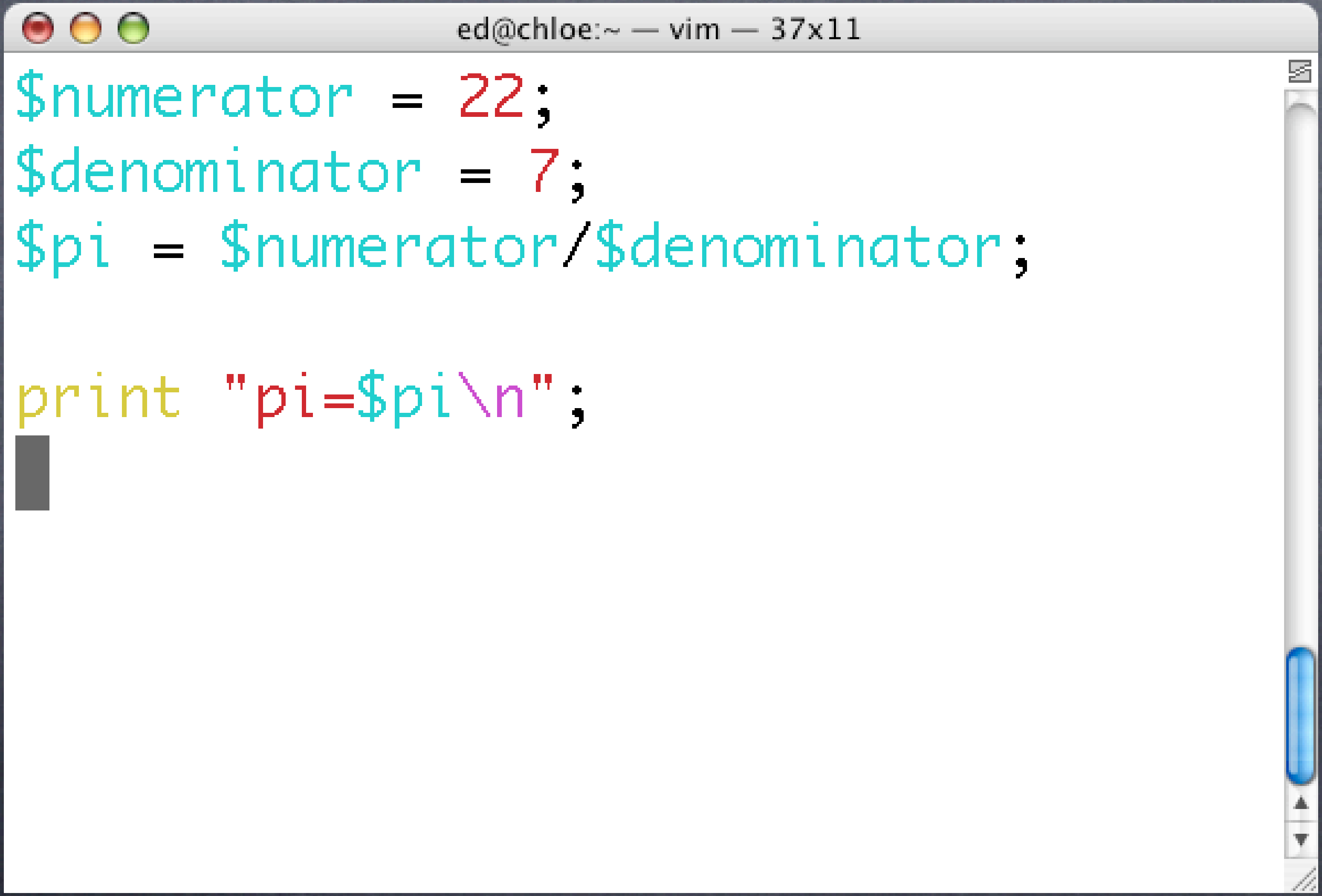
$ @   %

$ Strings $



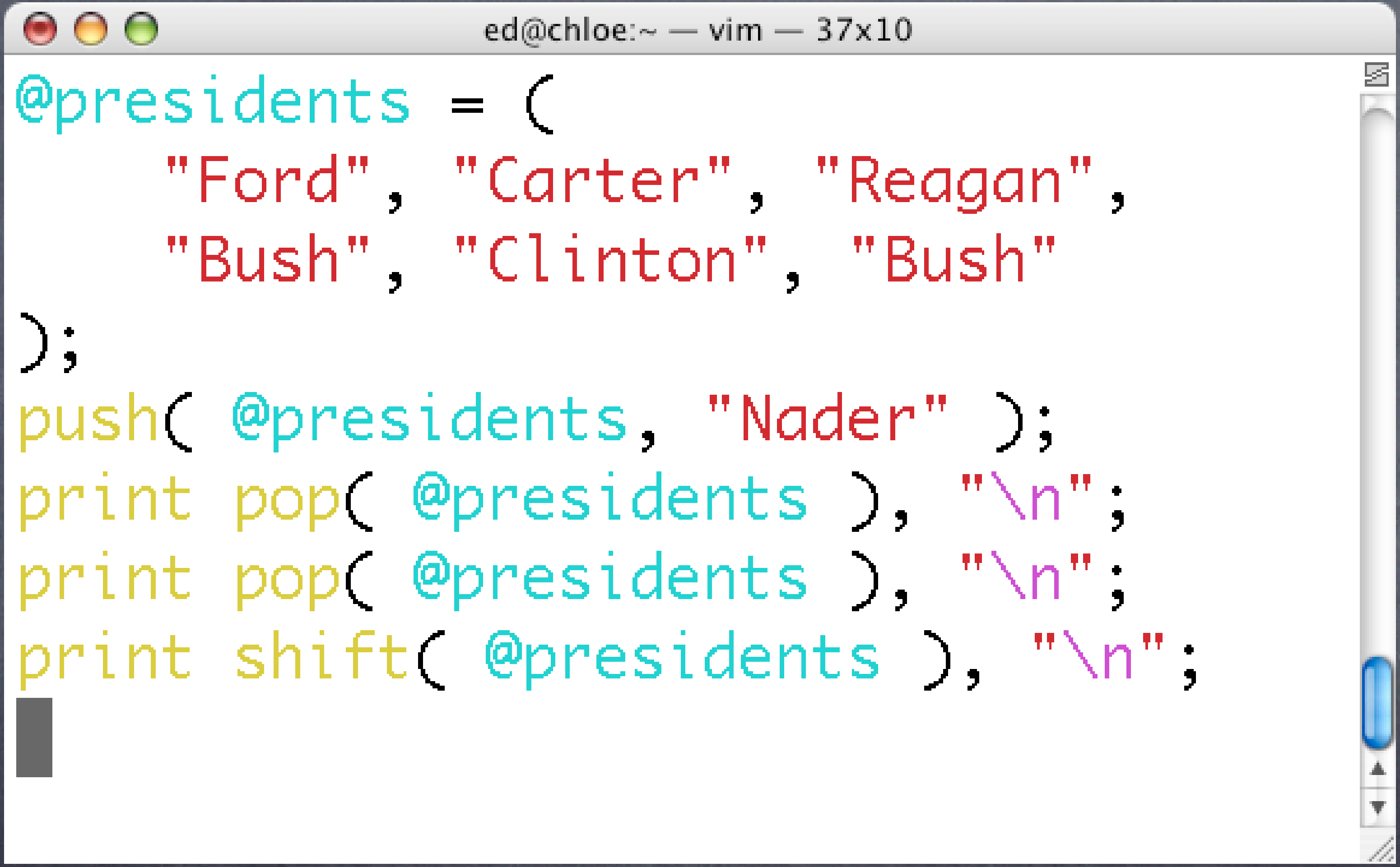
```
$name = "Ed Summers";
print "hello $name\n";
```

$ Numbers $



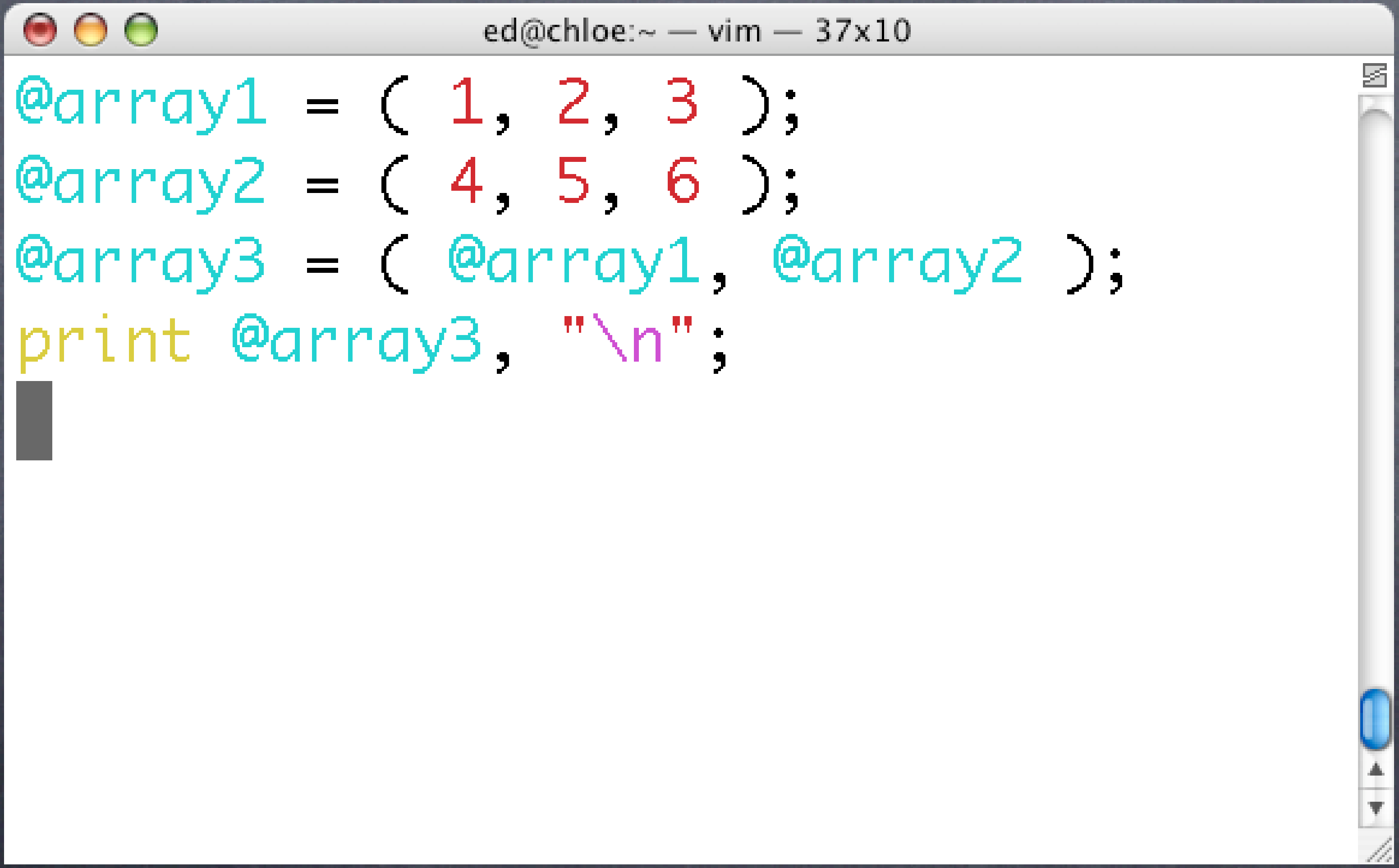
```
$numerator = 22;
$denominator = 7;
$pi = $numerator/$denominator;

print "pi=$pi\n";
```

@ Arrays @



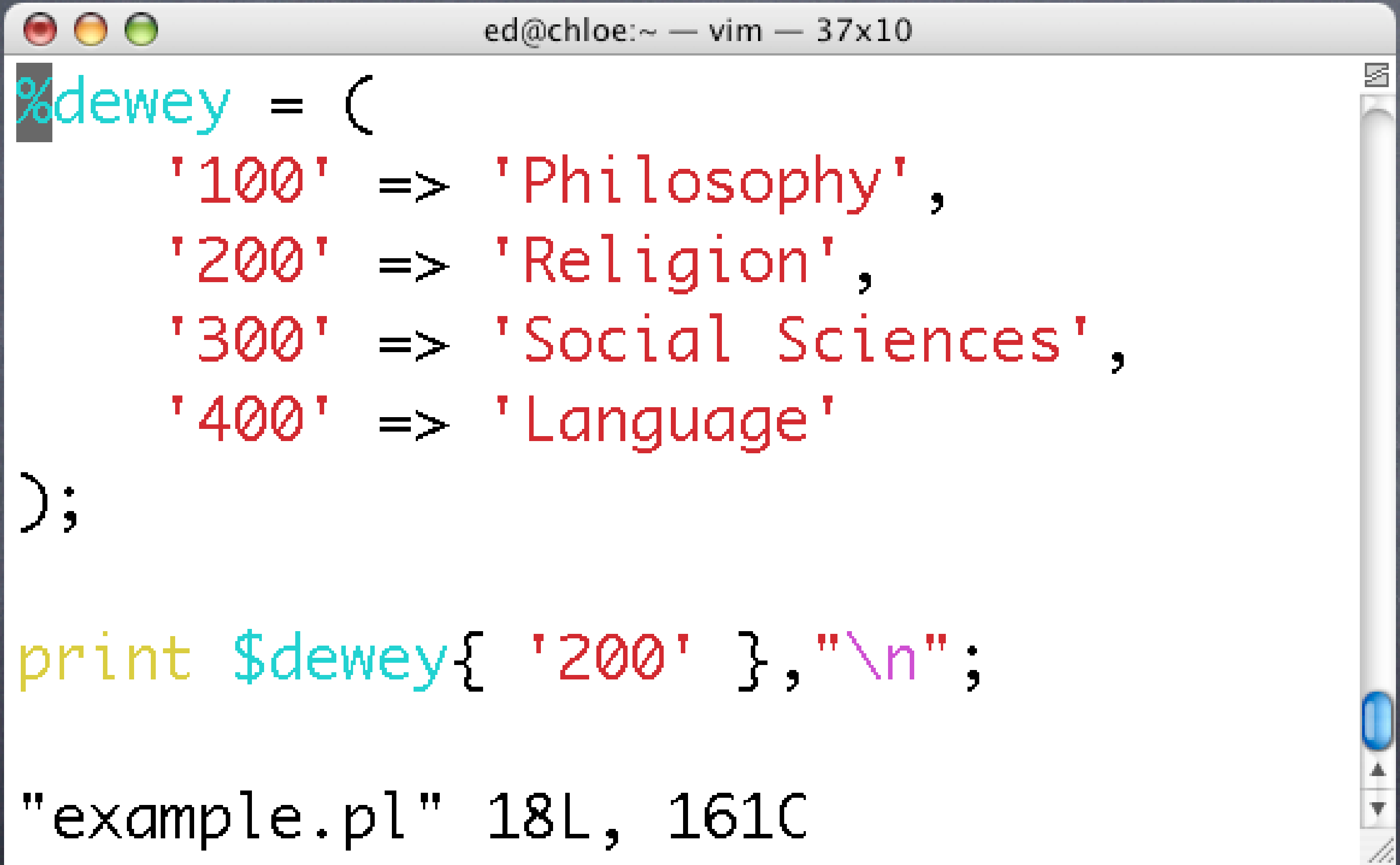
```
@presidents = (
    "Ford", "Carter", "Reagan",
    "Bush", "Clinton", "Bush"
);
push( @presidents, "Nader" );
print pop( @presidents ), "\n";
print pop( @presidents ), "\n";
print shift( @presidents ), "\n";
```

@ Combining @



```
@array1 = ( 1, 2, 3 );
@array2 = ( 4, 5, 6 );
@array3 = ( @array1, @array2 );
print @array3, "\n";
```

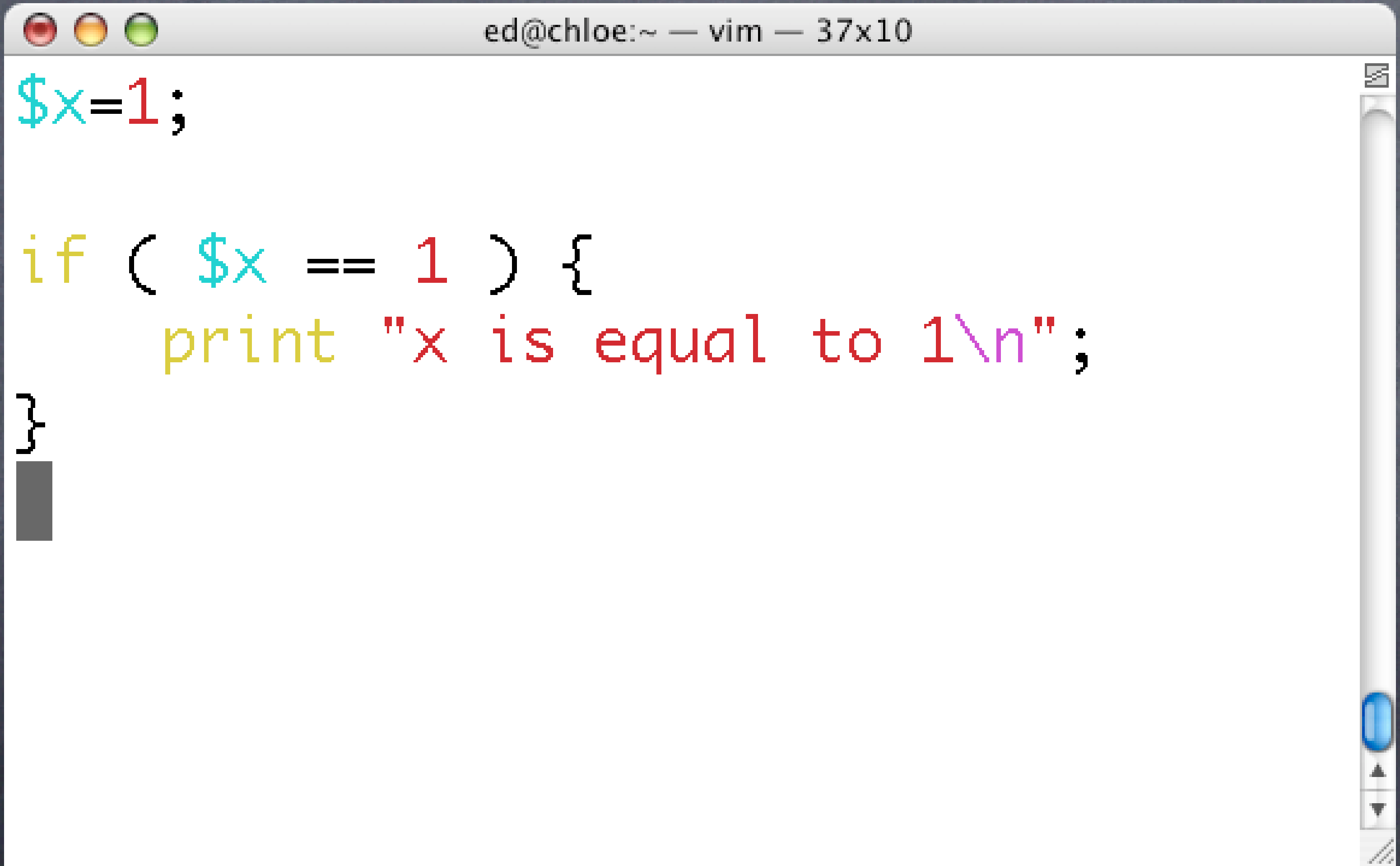
% Hashes %



```
%dewey = (
    '100' => 'Philosophy',
    '200' => 'Religion',
    '300' => 'Social Sciences',
    '400' => 'Language'
);

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

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

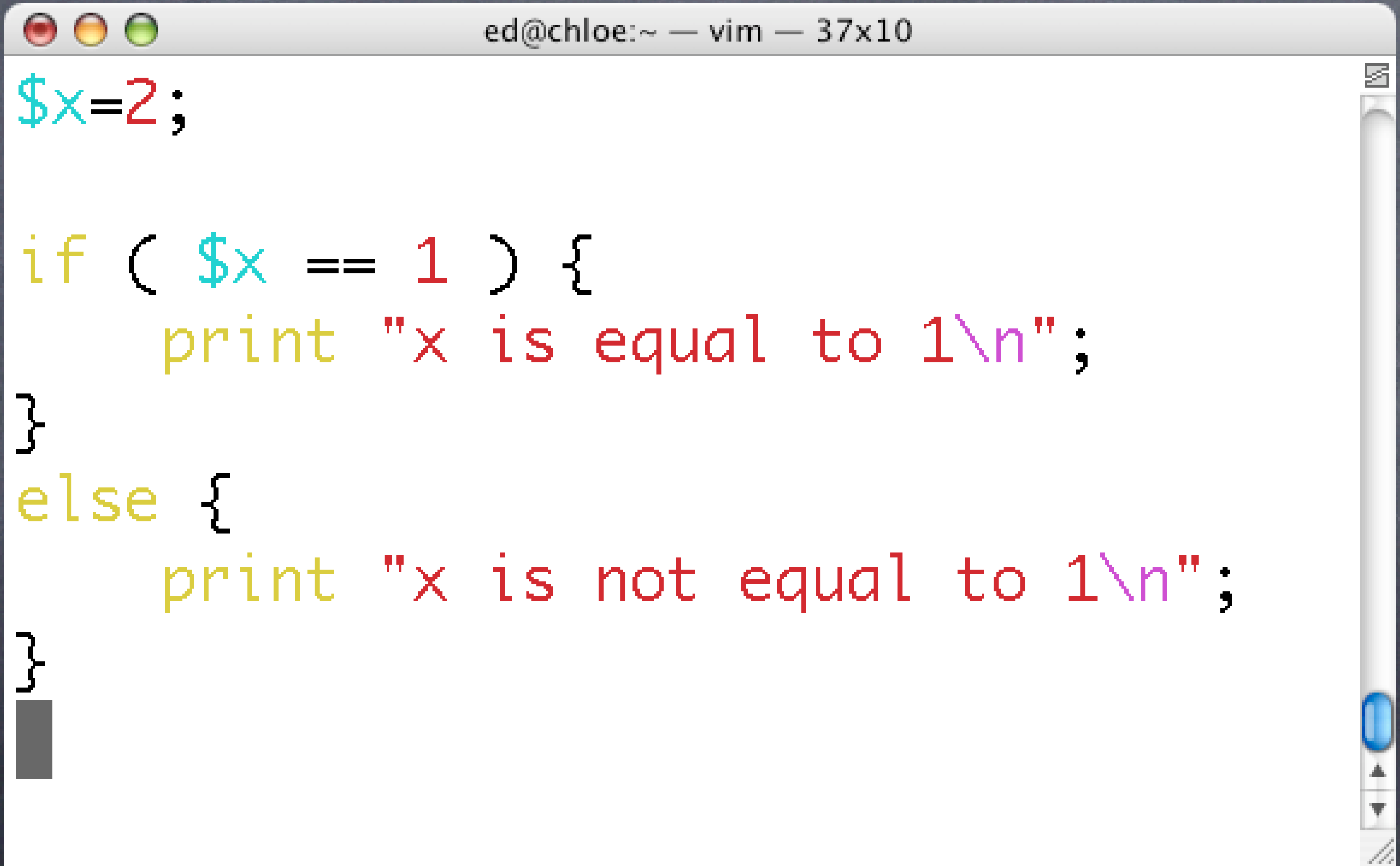
Flow Control

if



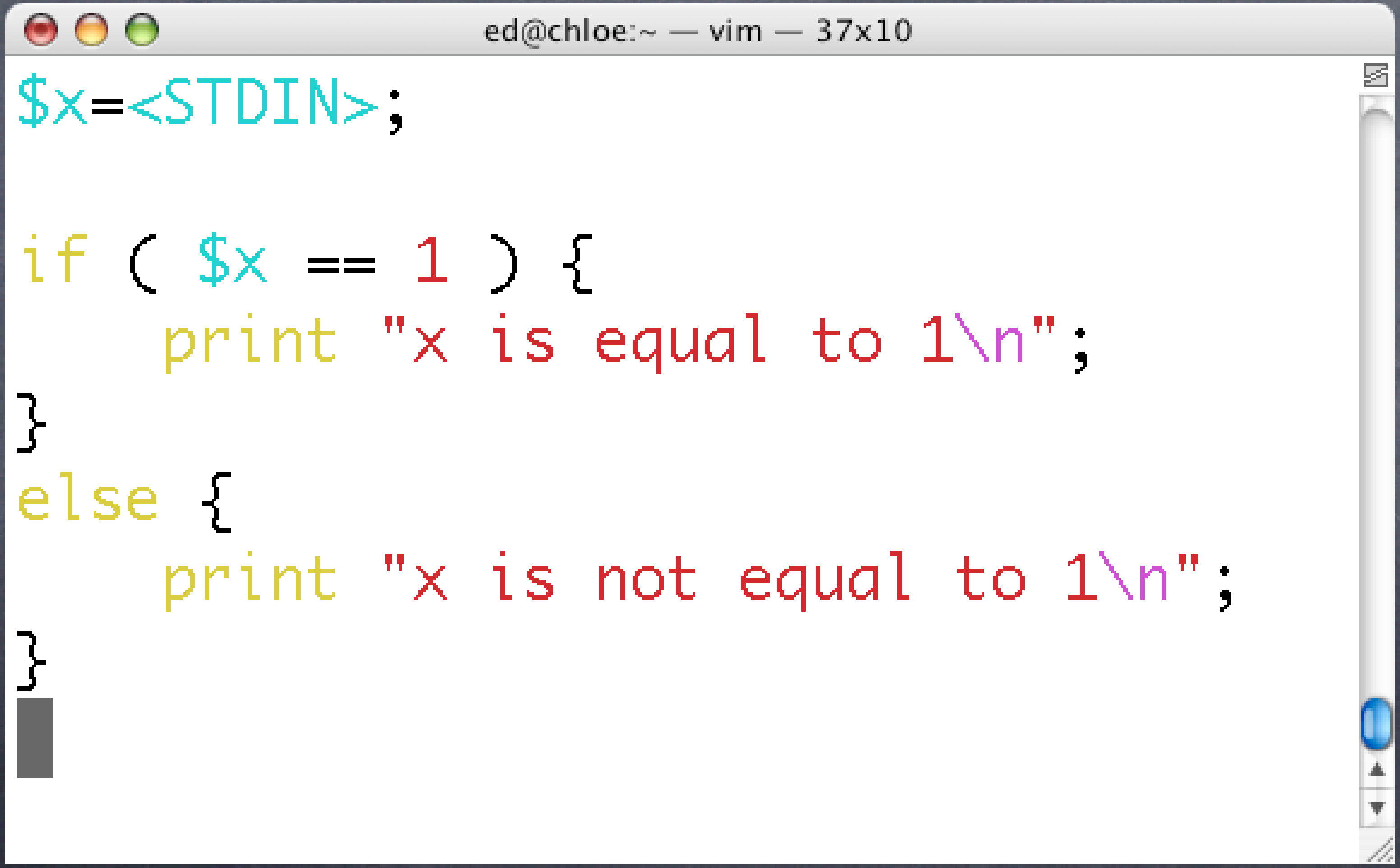
```
$x=1;

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

if/else



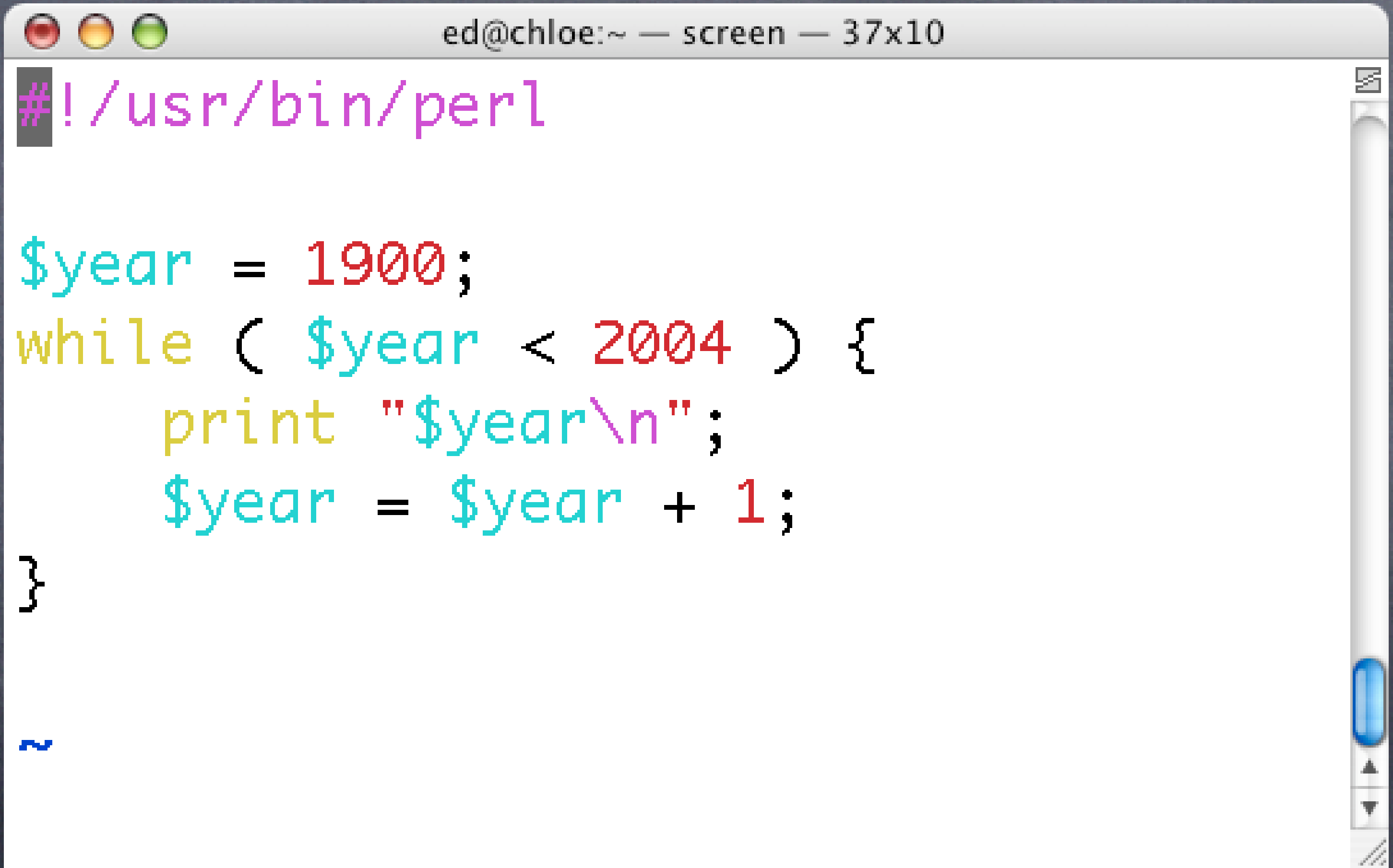
```
$x=2;

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

input



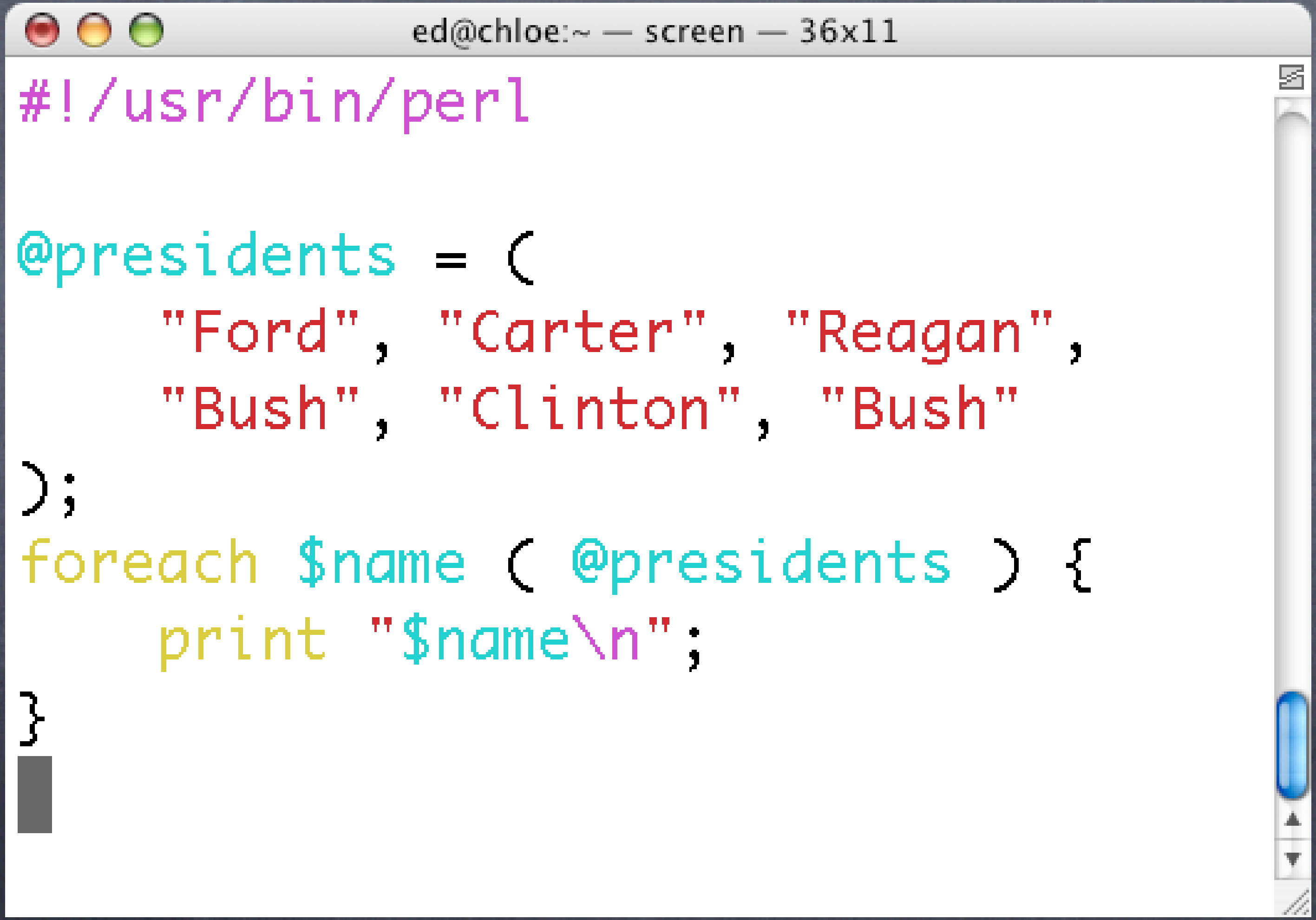
```
$x=<STDIN>;

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

while



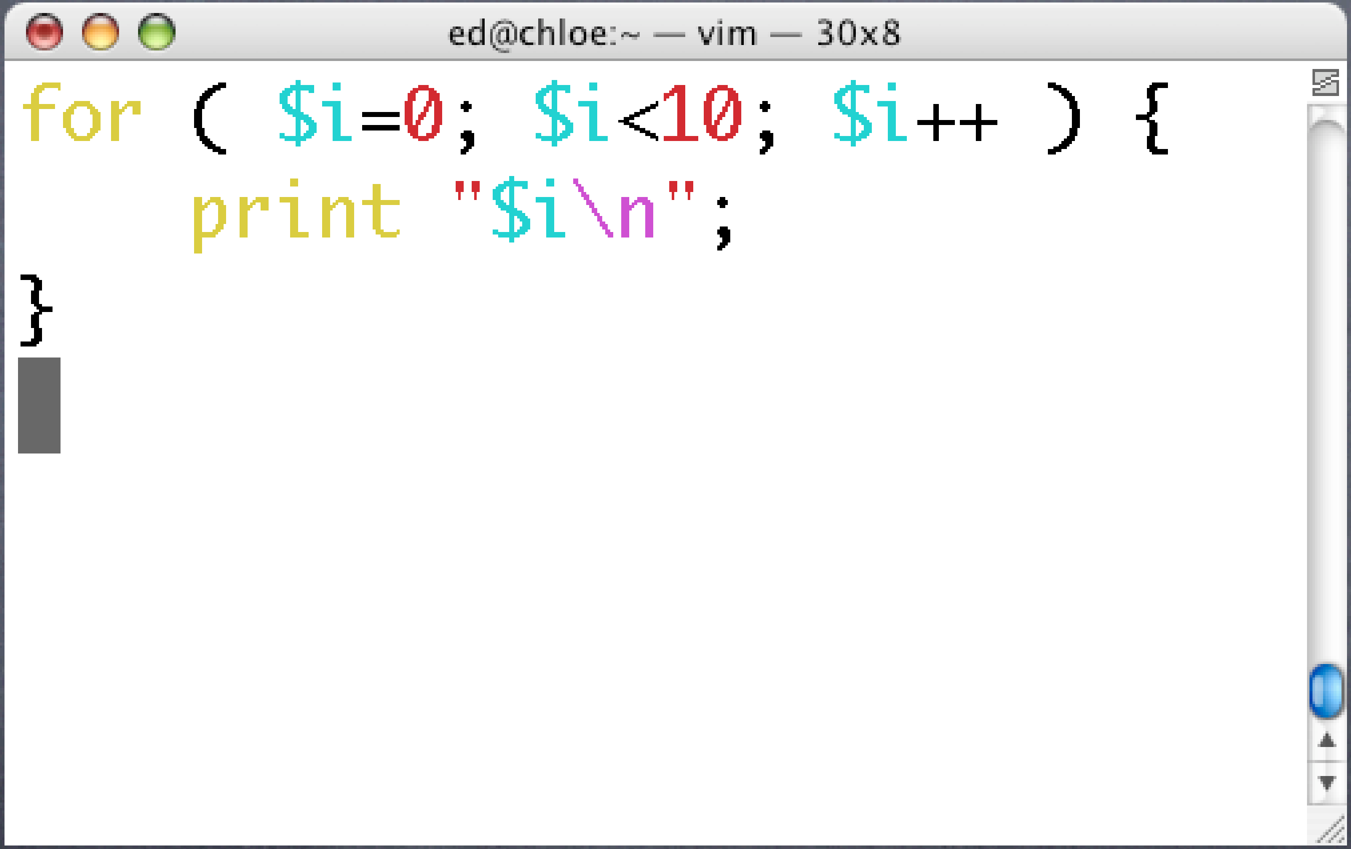
```
#!/usr/bin/perl

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


~
```

foreach



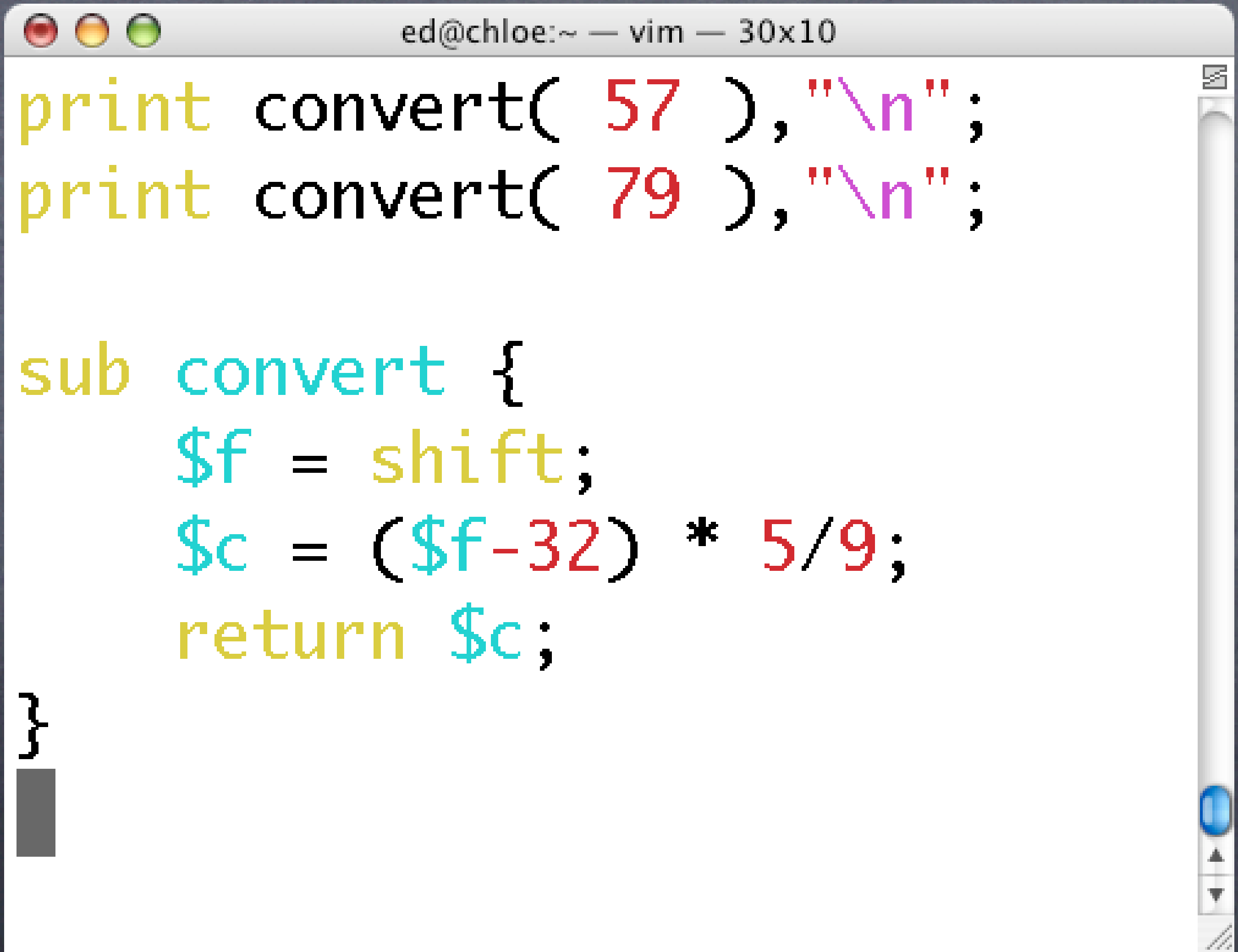
```
#!/usr/bin/perl

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

for



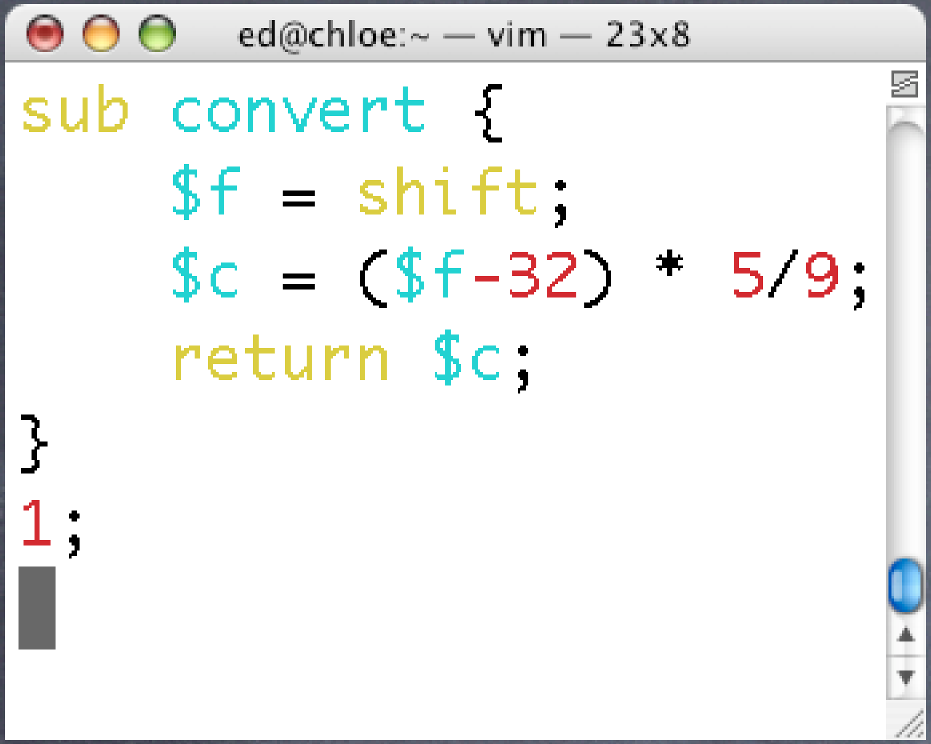
```
for ( $i=0; $i<10; $i++ ) {
    print "$i\n";
}
```

subroutines

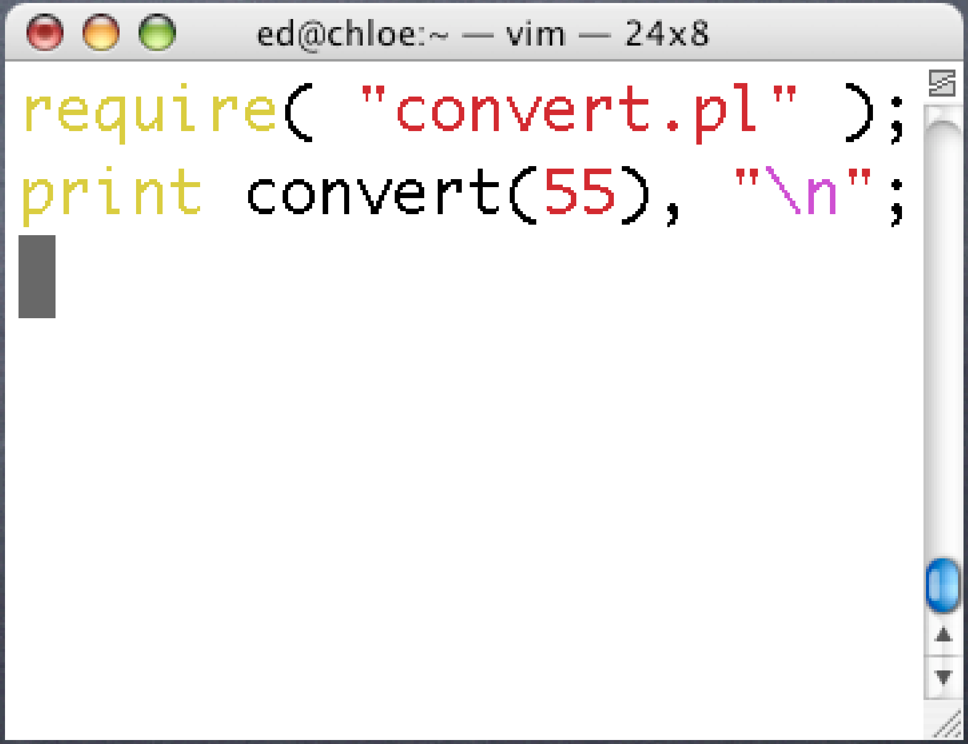


```
print convert( 57 ),"\n";
print convert( 79 ),"\n";

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

require

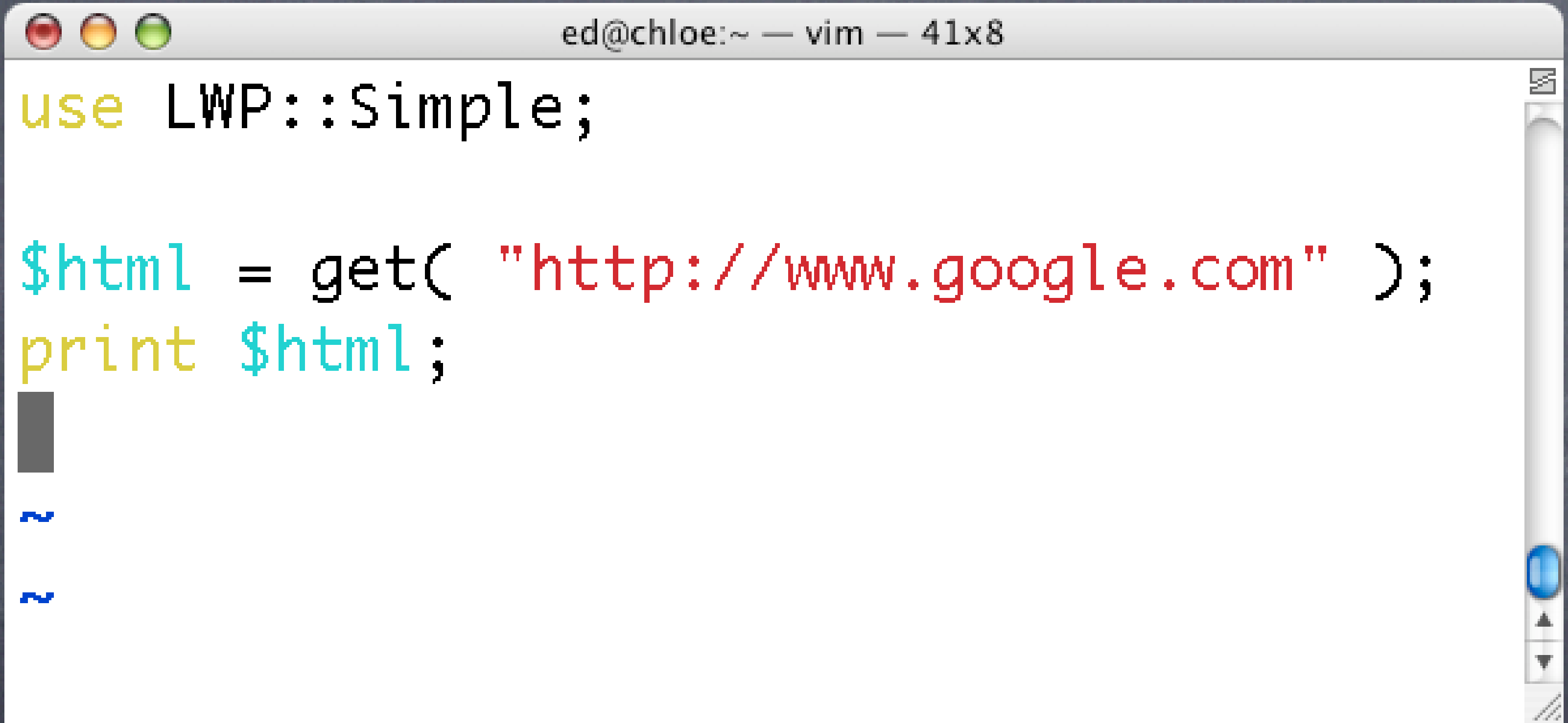


```
sub convert {
    $f = shift;
    $c = ($f-32) * 5/9;
    return $c;
}
1;
```



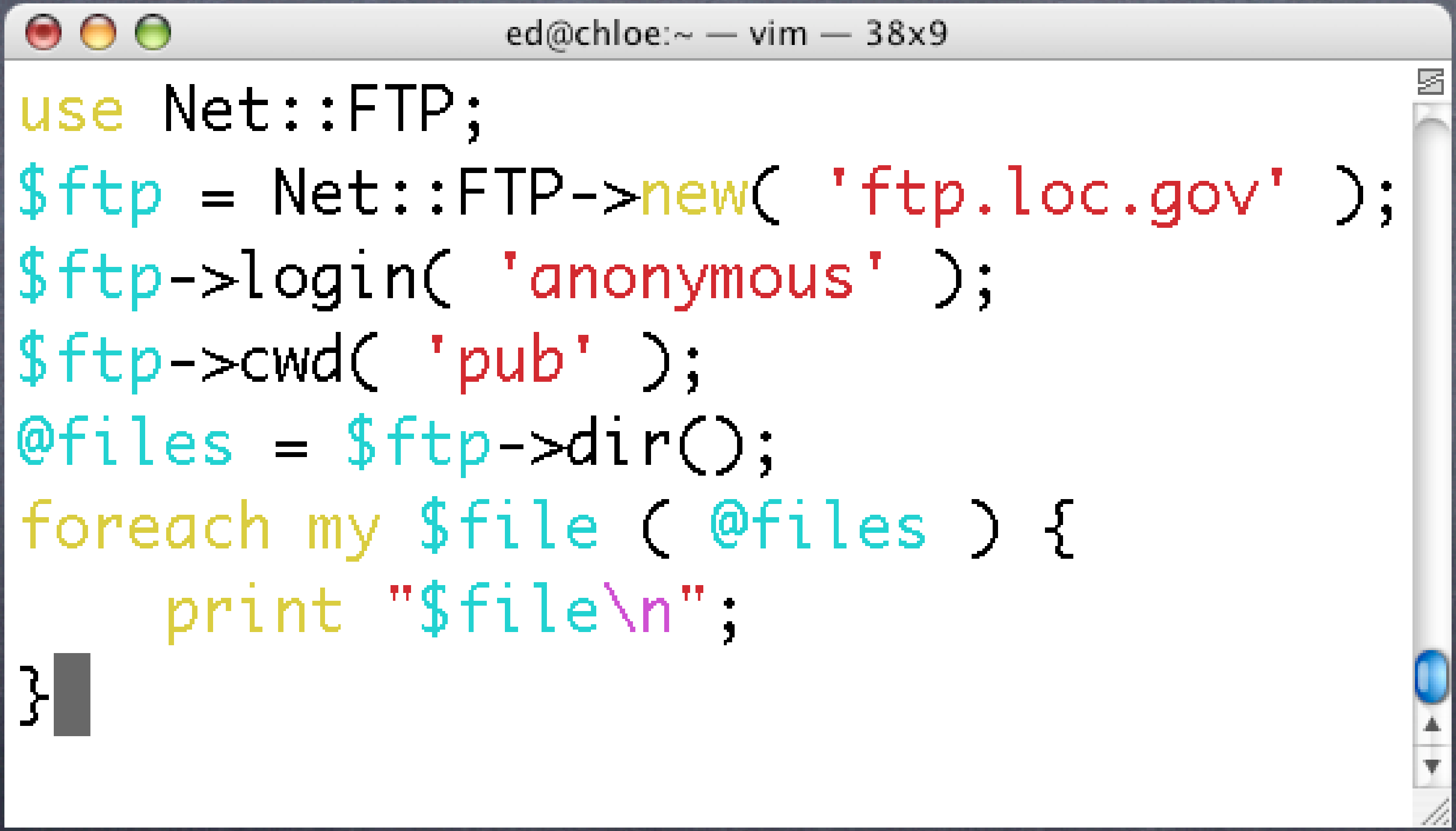
```
require( "convert.pl" );
print convert(55), "\n";
```

modules



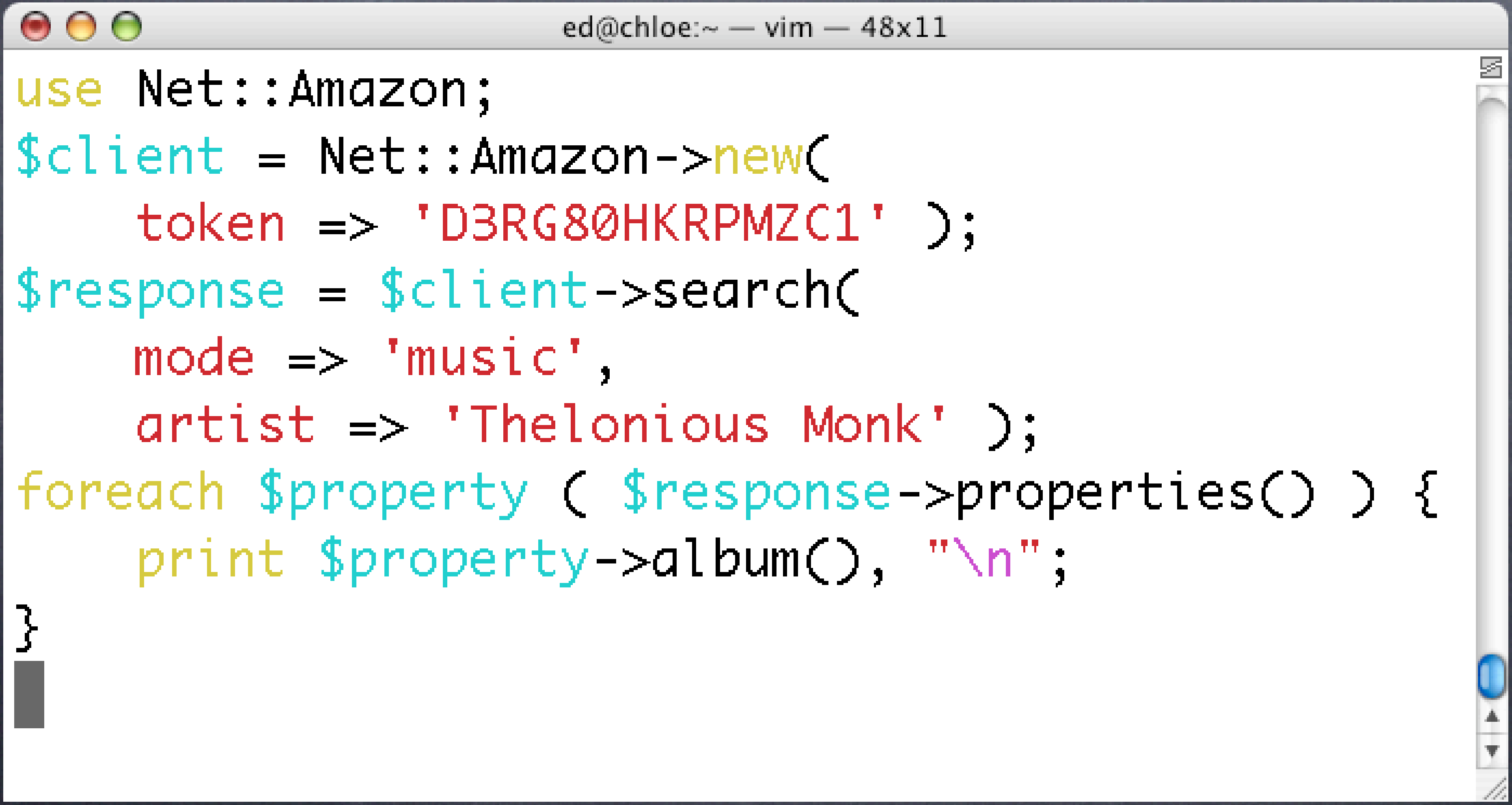
```
use LWP::Simple;

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

~
~
```

objects



```
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



```
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";
}
```