PLCT S.A de C.V

CURSO PERL INTERMEDIO

Descripción General

Objetivo

Que el estudiante entienda y aplique de manera efectiva los conceptos de módulos, paquetes, referencias, programación orientada a objetos, así como prueba y distribución de sistemas desarrollados con Perl.

Duración

30 Horas

Requisitos

www.cursoslinux.com.mx
ventas@plct.com.mx
PLCT S.A. de C.V.
Tel.: 55 4522 7839/55 1800 7696/7224447684

CURSO 2023

PLCT S.A de C.V

TEMARIO

1. Intermediate Foundations
    1.1. List Operators
    1.2. Trapping Errors with eval
    1.3. Dynamic Code with eval
    1.4. Exercises
1. Using Modules
    1.1. The Standard Distribution
    1.2. Using Modules
    1.3. Functional Interfaces
    1.4. Selecting What to Import
    1.5. Object-Oriented Interfaces
    1.6. A More Typical ObjectOriented Module:
1. Math::BigInt
    1.1. The Comprehensive Perl Archive Network
    1.2. Installing Modules from CPAN
    1.3. Setting the Path at the Right Time
    1.4. Exercises
1. Introduction to References
    1.1. Performing the Same Task on Many Arrays
    1.2. Taking a Reference to an Array
    1.3. Dereferencing the Array Reference
    1.4. Getting Our Braces Off
    1.5. Modifying the Array
    1.6. Nested Data Structures
    1.7. Simplifying Nested Element References with Arrows
    1.8. References to Hashes
    1.9. Exercises
1. References and Scoping
    1.1. More Than One Reference to Data
    1.2. What If That Was the Name?
    1.3. Reference Counting and Nested Data Structures
    1.4. When Reference Counting Goes Bad
    1.5. Creating an Anonymous Array Directly
    1.6. Creating an Anonymous Hash
    1.7. Autovivification
    1.8. Autovivification and Hashes
    1.9. Exercises

www.cursoslinux.com.mx
ventas@plct.com.mx
PLCT S.A. de C.V.
Tel.: 55 4522 7839/55 1800 7696/7224447684

PLCT S.A de C.V

1. Manipulating Complex Data Structures
    1.1. Using the Debugger to View Complex Data
    1.2. Viewing Complex Data with Data::Dumper
    1.3. YAML
    1.4. Storing Complex Data with Storable
    1.5. Using the map and grep Operators
    1.6. Applying a Bit of Indirection
    1.7. Selecting and Altering Complex Data
    1.8. Exercises
1. Subroutine References
    1.1. Referencing a Named Subroutine
    1.2. Anonymous Subroutines
    1.3. Callbacks
    1.4. Closures
    1.5. Returning a Subroutine from a Subroutine
    1.6. Closure Variables as Inputs
    1.7. Closure Variables as Static Local Variables
    1.8. Exercises
1. Filehandle References
    1.1. The Old Way
    1.2. The Improved Way
    1.3. The Even Better Way
    1.4. IO::Handle
    1.5. Directory Handle References
    1.6. Exercises
1. Practical Reference Tricks
    1.1. Review of Sorting
    1.2. Sorting with Indices
    1.3. Sorting Efficiently
    1.4. The Schwartzian Transform
    1.5. Multi-Level Sort with the Schwartzian Transform
    1.6. Recursively Defined Data
    1.7. Building Recursively Defined Data
    1.8. Displaying Recursively Defined Data
    1.9. Exercises
1. Building Larger Programs
    1.1. The Cure for the Common Code
    1.2. Inserting Code with eval
    1.3. Using do
    1.4. Using require

PLCT S.A de C.V

1.6. Using Class Variables

1.7. Weakening the Argument

1.8. Exercise

1. Some Advanced Object Topics

1.1. UNIVERSAL Methods

1.2. Testing Our Objects for Good Behavior

1.3. AUTOLOAD as a Last Resort

1.4. Using AUTOLOAD for Accessors

1.5. Creating Getters and Setters More Easily

1.6. Multiple Inheritance

1.7. Exercises

1. Exporter

1.1. What use Is Doing

1.2. Importing with Exporter

1.3. @EXPORT and @EXPORT_OK

1.4. %EXPORT_TAGS

1.5. Exporting in a Primarily OO Module

1.6. Custom Import Routines

1.7. Exercises

1. Writing a Distribution

1.1. There's More Than One Way To Do It

1.2. Using h2xs

1.3. Embedded Documentation

1.4. Controlling the Distribution with Makefile.PL

1.5. Alternate Installation Locations (PREFIX=...)

1.6. Trivial make test

1.7. Trivial make install

1.8. Trivial make dist

1.9. Using the Alternate Library Location

1.10. Exercise

1.11. Essential Testing

1.12. More Tests Mean Better Code

1.13. A Simple Test Script

1.14. The Art of Testing

1.15. The Test Harness

1.16. Writing Tests with Test::More

1.17. Testing Object-Oriented Features

1.18. A Testing To-Do List

1.19. Skipping Tests

1.20. More Complex Tests (Multiple Test Scripts)

1.21. Exercise

1. Advanced Testing
   1.1. Testing Large Strings
   1.2. Testing Files
   1.3. Testing STDOUT or STDERR
   1.4. Using Mock Objects
   1.5. Testing POD
   1.6. Coverage Testing
   1.7. Writing Your Own Test::* Modules
   1.8. Exercises