

Perl Programming

Duration:	3 days
Type:	beginner

Description

This course provides a complete introduction to the Perl scripting language. It includes both the latest features in Perl 5.10 and details of how the language is being redesigned in Perl 6. As the course progresses delegates learn to read data from files, filter it using regular expressions and store the result in data structures built from anonymous arrays and hashes. Best practises for Perl coding are introduced in each chapter, including techniques for creating reusable modules, classes and CPAN libraries.

Prerequisites

Delegates should have some previous programming experience, ideally in C, C++, C#, Java or JavaScript. By default the course is delivered on Windows using ActivePerl and the Komodo IDE, but other configurations are possible if required.

List of Modules

Introduction to Perl

- The evolution of Perl from Sed, Awk and Shell Scripting
- Installing Perl and downloading modules from CPAN
- Understanding how the Perl interpreter runs your script
- Differences between versions 5 and 6 of Perl
- Comparing Perl to Python and Ruby

Key Characteristics of Perl

- Sigils are used to denote variables
- How barewords are interpreted
- Packages, symbol tables and typeglobs
- Strings, comments and using heredoc
- Printing to and reading from the console

Basic Procedural Programming

- Declaring and initializing scalar variables
- Operators for strings and numbers
- What values count as true in Perl
- Variables created by the interpreter
- Performing tests using *if* and *unless*
- Different types of loops supported by Perl

File Access in Perl

- Creating handles and testing file paths
- Storing file handles in scalar variables
- Reading lines, characters and binary data
- Different techniques for slurping files
- File globbing and changing permissions

Arrays and Lists

- Initializing arrays based on lists
- Initializing arrays implicitly
- Initializing list values based on arrays
- Slicing, expanding and shrinking arrays
- Built-in functions for manipulating arrays
- Introducing anonymous arrays

Working With Hashes

- Initializing hashes based on lists
- Initializing hashes implicitly
- Using the 'fat comma' notation
- Built-in functions for manipulating hashes
- Initializing an array based on a hash
- Using slices to add and retrieve multiple values
- Creating hashes of anonymous arrays

The Perl Regular Expression Syntax

- How a Regular Expression Engine operates
- Creating character classes and specifying multiplicities
- Shortcuts for character classes provided by Perl and POSIX
- Meta-characters for specifying positions in the input
- Using parenthesis for grouping and submatches

Running Regular Expressions in Perl

- Using the `=~` operator for pattern matching
- Iterating over all the matches for an expression
- Modifiers that can be used during matching
- Using the extended regular expression syntax
- Using the `=~` operator for making substitutions
- Using the *e* modifier to run code during substitutions

Regular Expressions - Advanced Concepts

- The non greedy versions of `*`, `+` and `?`
- Using parenthesis that do not capture
- Applying modifiers only to part of the expression
- Using look-around assertions to match without capturing
- Building regular expressions dynamically
- Using Unicode properties in expressions
- Named submatches in Perl 5.10

Subroutines in Perl

- Declaring subroutines and passing parameters
- Using hashes to implement named parameters
- Creating local variables with *my*, *our* and *local*
- Using anonymous subroutines for meta-programming

Perl Error Handling

- Using *eval* to catch and handle errors
- Enhanced error handling provided by *Carp*

References in Perl

- Introducing references as memory addresses
- Declaring references to named and anonymous variables
- The different syntaxes available for working with arrays
- Using references to create complex data structures
- Best practises for navigating complex data structures

Creating Perl Modules

- Understanding how packages work in Perl
- The relationship between packages and modules
- How the *Exporter* module exports symbols
- Using the *EXPORT*, *EXPORT_OK* and *EXPORT_FAIL* arrays
- Implementing versioning in your modules
- Publishing modules to CPAN

Object Orientation in Perl

- Reviewing the key concepts of OO
- The effect of the *bless* keyword in Perl
- Declaring classes with fields and methods
- Creating class hierarchies and overriding methods

Using Perl Libraries (Optional)

- Manipulating XML with SAX/DOM/XPath
- Database access with DBI
- Creating Web Applications

New Features in Perl 6 (Optional)

- Sigils are now invariant
- Better syntax for indexing
- Modifications to operators
- Changes to regular expressions
- Changes to Object Orientation
- Changes to references