

Perl for Web Development (5 days)

Synopsis

A specialist course designed to enable web developers to build dynamic (database-driven) e-commerce web sites using the Perl programming language, its specialist web modules and CGI (the common gateway interface).

The course includes a thorough grounding in generic Perl programming before moving on to specialised web development tools and techniques.

Who Should Attend

Programmers who need to develop serious web sites on any of today's major platforms and in multi-platform environments.

Web Developers with limited scripting experience (such as JavaScript, VBScript, Unix/Linux shell scripting, etc rather than pure programming) will benefit from this course, but may find it demanding. Some may choose to extend the same material over more than four days, with additional hands-on practical exercises.

Delegates with backgrounds in C like languages will recognise elements of Perl syntax and can use existing knowledge to progress quickly, but they must beware the temptation to translate literally from C/C++. Perl's native way of doing things is invariably more efficient and better adapted to specialist web development tasks.

Like all of our Perl courses, this course is designed for cross-platform application, but Unix/Linux developers will benefit particularly from Perl variants of popular Unix tools and from the abundance of Perl modules for use with Unix and the Apache web server.

Course Contents

Preparing to Learn Perl

- Things you need to know and do in order to run Perl and learn Perl programming
- A module designed for complete beginners
- The perl compiler/interpreter
- Perl under Unix/Linux
- Perl under MS Windows 2000/NT/95/98/ME (perl.exe)
- ActiveState Perl
- Making programs executable
- Perl from the command line (% perl)
- Specify the Perl compiler/interpreter (!)
- Using plain text
- Writing a very simple program
- Running a very simple program
- Basic syntax

Perl: The Absolute Minimum

- Enough of the language to get started
- The print command
- Variables
- Scalars — numbers and 'strings'
- Assignment
- Simple conditional tests — if "strings" "\n"

- Lists
- Arrays — for storing lists
- foreach
- Hashes
- Other loops: while, for, do, until
- Arrays — the rest
- Simple input, e.g. while(<>)
- Functions overview — recognising, writing, using
- Regular expressions — perlre
- Simple file handling — open, print
- Subroutines — parameters in and out, listification, local variables
- Help — perldoc, books, web

Perl: Beyond the Basics

- More flow control
- Statement modifiers
- Quoting mechanisms — qq(), etc
- Here documents
- Uppercase/lowercase conversion
- Splitting strings into lists
- Joining lists into strings
- Filtering lists with map
- Sorting lists
- The importance of context
- Assignment shortcuts
- Scoping rules
- Special variables

Complex Data Structures & References

- Limits of flat lists
- Nesting arrays
- Array references
- Anonymous arrays
- Named array references
- Passing multiple arrays to/from functions
- Hashes of arrays
- Hash references
- Arrays of hashes
- Hashes of hashes
- Complex nested data structures
- Code references
- Dispatch tables

Finding Out More For Yourself

- How to read Perl's documentation
- Where to find more information
- Knowing what's out there to look for
- FAQs

The Perl Debugger & Debugging Perl

- Avoiding bugs
- Perl's built-in debugger
- Invoking the debugger
- What you can do with the debugger
- Understanding the debugger's command line interface
- Knowing the debugger's basic command set
- Exploring some extended functions
- Graphical debuggers
- Alternative debugging techniques

Using Perl Modules from CPAN

- The Comprehensive Perl Archive Network (CPAN)
- Why effective Perl programmers are efficient CPAN users
- CPAN's philosophy
- Finding modules
- Installing Modules
- Using modules
- Some particularly useful modules

Command-Line Perl

- General principles
- Using Perl as a filter
- Awk-like Perl
- Many real world examples
- Many examples using regular expressions
- Command line flags

A Whistle-Stop Tour of the World of Perl

- Wheels you don't need to re-invent
- Common recipes
- Common pitfalls

Advanced File Processing with Perl

- Types of open
- Filehandles
- Reading line by line
- Reading paragraph by paragraph
- Reading entire files
- Special variables
- The flip-flop operator (..)
- File test functions
- Pipes

Perl Security Issues

- Potential security pitfalls
- Coding for security
- Taint checking
- Dangerous environment variables
- File input
- Set-user-id Perl programs
- Permissions and users

Handling Databases with Perl

- Interacting with SQL databases from Perl
- Using DBI
- Using ODBC
- Connection/disconnection, log in/log out
- Doing simple queries
- Doing complex queries, e.g. hash and array references
- Linking with HTML:Template
- Monitoring database servers
- Database related modules

Introduction to CGI

Programming With Perl

- Printing a CGI header for a simple command-line program
- Running the simple program as a CGI script
- Form handling — CGI::Lite.pm for parameter parsing
- GET
- POST
- Debugging CGI programs
- Running CGI programs at the command line

More Advanced CGI

Techniques

- Hidden fields
- Maintaining state
- Cookies
- CGI & HTTP headers
- Redirection to other pages
- File uploading

Web Security Issues

- CGI scripts and security
- User input on the web
- Guest books
- Dangerous strings
- Unsafe process calls
- File upload exploits
- Safe database access
- Permissions for CGI programs