

Perl is a simple, yet powerful, scripting language that can be very useful for automating repetitive tasks, processing text and acting as a glue between other programs. This 3-day class introduces Perl to both new programmers and experienced ones who want to learn a new language. It provides a host of real-world applications for today's environments so readers can get started immediately.

Course Objectives:

- Describe the fundamental data types for Perl
- Create and access arrays
- Program with branching and looping constructs
- Create and access hashes
- Input from the keyboard and output from the screen
- Utilize regular expressions with Perl
- Create and use functions

Audience: Programmers and/or system administrators.

Prerequisites: Programming experience in a structured language, such as C, C++, Java, or COBOL, is recommended.

Number of Days: 3 days

1 Introduction – Meet Perl

Origin of Perl
 Perl Versions
 Invoking Perl
 Perl Documentation
 Perl Resources

2 Scalar Variables

Numeric Literals
 Manipulating Numbers
 String Literals
 Manipulating Strings
 Single vs. Double Quotes
 Scalars Variables
 Undefined Variables
 Auto increment and decrement
 Reading data from the user
 Chomp & Chop
 Curly Braces
 Additional Resources

3 Array Variables

Referencing Array Elements
 Adding & Removing Elements

The Splice Function
 Using for Loops
 The Reverse Statement
 The Sort Operator
 The qw & qq statements
 Arrays used in scalar context
 Additional Resources

4 Associate Array Variables

Crating Associate Arrays
 The Values Statement
 Reverse searching an associative array exists vs. defined
 Special Variables
 The Environment Variables
 The Argument Variable
 Additional Resources

5 Flow Control

Blocks
 The if Statement
 The unless Statement
 The switch Statement
 The given Statement

	The while Statement		index
	The until Statement		rindex
	The do Statement		grep
	Loop Control: last		srand & rand
	Loop Control: next		sleep
	Additional Resources		Additional Resources
6	Conditional Expressions	11	Filesystem & Process Control
	Numeric Comparison		Controlling the Filesystem within Perl
	String Comparison		Working with Directories
	Pattern Matching		Working with Files
	Using the Outcome of a Statement		Back-Quoting
	File Test Conditions		The System Statement
	Complex Conditional Expressions		Additional Resources
	Understand and/or versus &&/	12	Functions
	Using Parentheses		Creating Functions
	Short Circuiting		Invoking Functions
	Additional Resources		Returning Values from Functions
7	Basic Input and Output		Passing Parameters
	Reading Input		Scope of Variables
	while & until Loops		local() vs. my()
	Record Separator Variable		Additional Resources
	The Diamond Operator	13	Using Modules
	The Default Variable		What are Modules?
	Using Parentheses		Loading Modules with use
8	Advanced Input and Output		Other Functions of use
	Filehandles		Additional Resources
	The die and warn Statements	14	Debugging Perl
	Opening and Reading from Files		The -w Switch
	Opening and Writing to Files		The Perl Debugger
	Reading a Block of a Filehandle		Debugger Commands
	Reading a Single Character		Additional Resources
	Piping in Perl		
	The format Statement		
	Here Documents		
	Additional Resource		
9	Pattern Matching		
	Pattern Matching vs. Wildcards		
	Matching, Substituting and Translation		
	Modifiers		
	Regular Expressions: Metacharacters		
	Regular Expressions: Classes		
	Regular Expressions: Backreferencing		
10	Perl Utilities		
	split		
	join		
	substr		