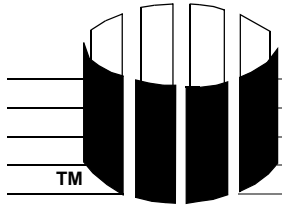


STONEHENGE CONSULTING SERVICES, Inc. 121 SW Morrison St, Suite 830 Portland, OR 97204-3138 USA

# Learning Perl

an introductory course by Randal L. Schwartz and Tom Phoenix  
Stonehenge Consulting Services, Inc.

Version 3.1.2 (1/23/06)[L]



## Table of Contents

### Introduction

What this course is about .....	3
Larry Wall .....	4
History of Perl .....	5
Modern Perl .....	6
Purpose of Perl .....	7
Availability .....	8
Support .....	9
Perl books .....	10
Making the program .....	11
Whitespace and comments .....	12
General concepts .....	13
Notes on compilation .....	14
A second example program .....	15

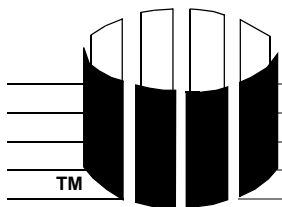
### Scalars

Scalar data .....	17
Numbers .....	18
Numeric operators .....	19
Strings .....	20
Double-quoted string literals .....	21
String operators .....	22
Automatic conversion between numbers and strings .....	23

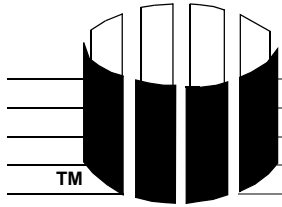
-w warnings .....	24
Scalar variables .....	25
Assignment operators .....	26
Generating output with print .....	27
Variable interpolation of scalars into strings .....	28
Controlling variable interpolation .....	29
Operator precedence and associativity .....	30
Arithmetic and string comparisons .....	31
The if control structure .....	32
Boolean values .....	33
Logical “not” .....	34
<STDIN> as a scalar value .....	35
The chomp operator .....	36
Combining chomp with <STDIN> .....	37
The while control structure .....	38
The undef value .....	39
More about undef .....	40

### Lists and Arrays

What is an array? .....	42
Array element access .....	43
More on array element access .....	44
Special array indexing .....	45



List literals and the <code>qw//</code> shortcut	46	A better <code>max</code> routine	71
List assignment	47	Notes on lexical ( <code>my</code> ) variables	72
The array as a whole	48	The use <code>strict pragma</code>	73
Assigning to an array	49	The <code>return</code> operator	74
Array names inside literal lists	50	Omitting the ampersand	75
The <code>pop</code> and <code>push</code> operators	51	<b>Hashes</b>	
The <code>shift</code> and <code>unshift</code> operators	52	What is a hash?	77
Interpolation of arrays into strings	53	Why use a hash?	78
The <code>foreach</code> control structure	54	Hash element access	79
More on the <code>foreach</code> control structure	55	Modifying hashes	80
Perl's favorite default: <code>\$_</code>	56	The hash as a whole	81
The <code>reverse</code> operator	57	Hash assignment	82
The <code>sort</code> operator (simple version)	58	Hash operators	83
Scalar and list context	59	Typical use of a hash	84
More on scalar and list context	60	The <code>exists</code> operator	85
Context summary	61	The <code>delete</code> operator	86
Converting a scalar to a list	62	Hash element interpolation	87
<code>&lt;STDIN&gt;</code> as a list	63	Summary of Variable Access Syntax	88
<b>Subroutines</b>		<b>Basic I/O</b>	
Defining and invoking a user function (subroutine)	65	Input from standard input	90
Return values	66	Input from the diamond ( <code>&lt;&gt;</code> ) operator	91
Arguments	67	The invocation arguments	92
Private variables in subroutines	68	Output to standard output	93
The <code>local</code> operator	69	Notes on <code>print</code>	94
The difference between <code>local</code> and <code>my</code>	70		



Formatted output with <code>printf</code> .....	95
More <code>printf</code> formats .....	96

## Concepts of Regular Expressions

Regular expressions .....	98
Using simple patterns .....	99
Repeat operators .....	100
Alternatives .....	101
A pattern test program .....	102

## More about Regular Expressions

Character classes .....	104
Character class shortcuts .....	105
Quantifiers .....	106
Anchoring patterns .....	107
Word-boundary anchors .....	108
Memory parentheses .....	109
Regular Expression metacharacters .....	110
Regular expression precedence .....	111

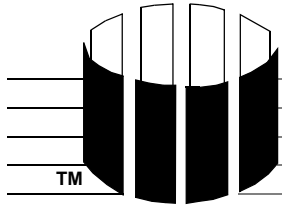
## Using Regular Expressions

Matching with regular expressions .....	113
More on matching .....	114
Interpolating into patterns .....	115
The match variables .....	116
Automatic match variables .....	117

Substitutions .....	118
More on substitutions .....	119
Case shifting .....	120
The <code>split</code> operator .....	121
The <code>join</code> operator .....	122

## Additional Control Structures

Reversed logic control structures .....	124
Expression modifiers .....	125
The “naked block” control structure .....	126
The <code>if</code> control structure’s <code>elsif</code> clause .....	127
Autoincrement and autodecrement .....	128
The value of autoincrement .....	129
The <code>for</code> control structure .....	130
Notes on the <code>for</code> control structure .....	131
More notes on the <code>for</code> control structure .....	132
The <code>last</code> operator .....	133
The <code>next</code> operator .....	134
The <code>redo</code> operator .....	135
The looping control statements, as a picture .....	136
Labeled blocks .....	137
Combining logical values .....	138
Partial-evaluation operators .....	139
Control structures using partial-evaluation operators .....	140



## Filehandles and File Tests

What is a filehandle? .....	142
Opening and closing a filehandle .....	143
Bad filehandles .....	144
A slight diversion: die .....	145
Using filehandles .....	146
Some advanced output techniques .....	147
The -X filetests .....	148
More filetests .....	149
The stat operator .....	150
Bitwise operators .....	151

## Directory Access

Moving around the directory tree .....	153
Globbering .....	154
An alternate syntax for globbing .....	155
Directory handles .....	156
Skipping some files with dirhandles .....	157

## Manipulating Files and Directories

Removing a file .....	159
Renaming a file .....	160
Creating a link to a file .....	161
Notes on file links .....	162
Making and removing directories .....	163
Modifying permissions and ownership .....	164

Changing timestamps .....	165
---------------------------	-----

## Process Management

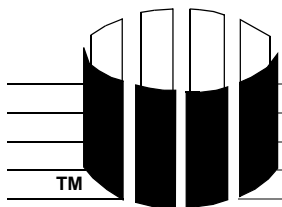
The system operator .....	167
The exec operator .....	168
Backquotes in scalar context .....	169
Backquotes in list context .....	170
Processes as filehandles .....	171
The fork and wait operators .....	172
The fork and wait operators in Perl .....	173
Sending and receiving signals .....	174

## Strings and Sorting

Finding a substring .....	176
Extracting and replacing a substring .....	177
Formatting data with sprintf .....	178
Advanced sorting .....	179
Making a sort subroutine .....	180
Notes on making a sort subroutine .....	181
Sorting by values in a hash .....	182
Sorting by multiple keys .....	183

## Simple Databases

DBM files and DBM hashes .....	185
Opening and closing DBM hashes .....	186
Using a DBM hash .....	187



Packing and unpacking binary data . . . . .	188
Fixed-length random access databases . . . . .	189
Variable-length (text) databases . . . . .	190
Updating text files . . . . .	191

### **Some Advanced Perl Techniques**

Selecting a subset from a list . . . . .	193
Transforming a list. . . . .	194
Minimal pattern matching . . . . .	195
Matching multiple lines at once. . . . .	196
Introduction to slices . . . . .	197
Pulling an item from a list . . . . .	198
List slice. . . . .	199
Array slice . . . . .	200
Hash slice. . . . .	201
Slice subscripting. . . . .	202
Slice interpolation . . . . .	203
Using <code>eval</code> to trap errors . . . . .	204
Typical use of <code>eval</code> . . . . .	205
Notes on using <code>eval</code> . . . . .	206

### **Miscellany**

Converting <code>find</code> command lines and other programs to Perl. . . . .	208
Very brief overview of additional features not covered . . . . .	209
Very brief overview, part two . . . . .	210

### **Conclusion**

Questions and answers. . . . .	212
--------------------------------	-----