

UNIX SHELL PROGRAMMING

TABLE OF CONTENTS

- **1 - INTRODUCTION:**
 - What you should already know... 1-2
 - What is the shell? 1-5
 - The role of the shell 1-6
 - Determining your shell type 1-7
 - What is a shell program? 1-8
 - Creating a shell script 1-10
 - Comments 1-13
 - The subshell type descriptor record 1-15
 - Exercises 1-17
 - Exercise solutions 1-18

- **2 - I/O AND VARIABLES:**
 - Using echo 2-2
 - Using print and printf 2-4
 - Variables: overview 2-6
 - Variable assignment 2-8
 - Variable substitution 2-9
 - Variable "dump": set 2-11
 - Variable assignment: read 2-12
 - Removing a variable: unset 2-14
 - Exporting variables: export 2-15
 - "Sourcing" a script with the dot (.) 2-18
 - The backslash as a quote: \ 2-19
 - Single quotes: ' ' 2-22
 - Double quotes: " " 2-24
 - Exercises 2-26
 - Exercise solutions 2-28

UNIX SHELL PROGRAMMING

TABLE OF CONTENTS (continued)

- **3 - CONDITIONAL BRANCHING:**
 - Overview 3-2
 - Return values 3-3
 - Basic decision making: if blocks 3-4
 - The opposite of if: else blocks 3-5
 - Contracted nesting: elif 3-8
 - The test command 3-11
 - The test operators 3-14
 - Short circuit operators 3-17
 - The exit command 3-19
 - Command substitution: `command` 3-20
 - Simple math: expr and \$(()) 3-21
 - Exercises 3-23
 - Exercise solutions 3-26

- **4 - WHILE LOOPS:**
 - Overview 4-2
 - The while loop 4-3
 - The until loop 4-7
 - Loop control: continue and break 4-9
 - Nested loops as branch points 4-12
 - Exercises 4-14
 - Exercise solutions 4-16

UNIX SHELL PROGRAMMING

TABLE OF CONTENTS (continued)

- **5 - FOR LOOPS AND POSITIONAL PARAMETERS:**

➤ Overview	5-2
➤ The for loop	5-3
➤ Positional parameters	5-5
➤ Accessing positional parameters: shift	5-6
➤ \$* versus \$@	5-8
➤ Loading up the positional parameters	5-15
➤ The for loop without a list	5-17
➤ Examples of for loops	5-18
➤ Exercises	5-20
➤ Exercise solutions	5-22

- **6 - DEBUGGING:**

➤ Overview	6-2
➤ Execution trace: set -x	6-3
➤ Verbose trace: set -v	6-8
➤ Combining -x and -v	6-10
➤ Error exit: set -e	6-12
➤ No execution: -n	6-14
➤ No unset: set -u	6-16
➤ Temporarily "commenting out" code	6-17
➤ Removing the # characters	6-19
➤ Temporary exits and breakpoints	6-20
➤ Bypasses	6-21
➤ Exercises	6-22
➤ Exercise solutions	6-24

UNIX SHELL PROGRAMMING

TABLE OF CONTENTS (continued)

- **7 - THE case AND select STATEMENTS:**
 - The case statement 7-2
 - The patterns 7-4
 - A case example 7-5
 - The select statement 7-7
 - A select example 7-9
 - LINES and COLUMNS 7-10
 - Exercises 7-12
 - Exercise solutions 7-13

- **8 - THE 'HERE' DOCUMENT:**
 - Overview 8-2
 - The 'here' document: << 8-3
 - Allowing indentation with tabs 8-5
 - Preventing substitutions 8-6
 - Self-extracting scripts 8-7
 - Exercises 8-8
 - Exercise solutions 8-10

- **9 - SPECIAL VARIABLE SUBSTITUTIONS:**
 - Overview 9-2
 - Default substitute value 9-3
 - Assign default value 9-6
 - Print error and exit 9-9
 - Alternate value substitution 9-12
 - Pattern substring deletion 9-15
 - Variable value length 9-18
 - Exercises 9-19
 - Exercise solutions 9-20

UNIX SHELL PROGRAMMING

TABLE OF CONTENTS (continued)

- **10 - ARRAYS AND typeset:**
 - Overview 10-2
 - The typeset built in 10-3
 - Typeset options 10-4
 - Examples of typeset 10-5
 - Removing attributes with typeset 10-7
 - Indexed arrays 10-8
 - Array example 10-9
 - The IFS variable 10-11
 - Exercises 10-19
 - Exercise hints 10-20
 - Exercise solutions 10-22

- **11 - SIGNALS AND TRAPS:**
 - Keyboard "interrupts" 11-2
 - Signals, PID's and kill 11-4
 - Terminating a process: kill 11-6
 - Protecting your scripts: trap 11-11
 - Setting an ignore trap 11-12
 - Setting a catch trap 11-16
 - Setting a catch trap for the exit signal 11-19
 - A trap within a trap 11-20
 - Exercises 11-21
 - Exercise solutions 11-22

UNIX SHELL PROGRAMMING

TABLE OF CONTENTS (continued)

- **12 - FUNCTIONS:**
 - Overview 12-2
 - Creating a function 12-3
 - Function definition files 12-6
 - Removing functions 12-8
 - Autoloading functions 12-9
 - Variable scope in functions 12-12
 - Positional parameters and functions 12-14
 - Exercises 12-16
 - Exercise solutions 12-20

- **13 - INTRODUCTION TO sed:**
 - Automating changes: sed 13-2
 - How sed works 13-3
 - sed commands 13-4
 - sed command examples 13-5
 - sed example 13-6
 - sed example walk through 13-7
 - The -n option 13-10
 - Exercises 13-11
 - Exercise hints 13-12
 - Exercise solutions 13-14

UNIX SHELL PROGRAMMING

TABLE OF CONTENTS (continued)

- **14 - MORE sed:**
 - Overview 14-2
 - Reading in a file: r 14-3
 - Writing to a file: w 14-5
 - Quitting: q 14-10
 - Grouped commands: { } 14-13
 - Multiline read next: N 14-14
 - Hold and get: h H g G 14-18
 - Do not execute: ! 14-22
 - Comments in a sed script: # 14-23
 - Exercises 14-24
 - Exercise hints 14-26
 - Exercise solutions 14-28

 - **15 - INTRODUCTION TO awk:**
 - A programming language: awk 15-2
 - Command line syntax 15-3
 - How awk works 15-4
 - Patterns and actions 15-5
 - Old, new or GNU? 15-6
 - Records, fields and special variables 15-7
 - Simple actions 15-8
 - Simple patterns 15-10
 - Regular expressions as patterns 15-12
 - More about printing 15-14
 - Literal values 15-15
 - Special patterns: BEGIN and END 15-18
- (continued)**

UNIX SHELL PROGRAMMING

TABLE OF CONTENTS (continued)

➤	Better output format: printf	15-20
➤	User defined variables	15-24
➤	Operators	15-26
➤	The END pattern	15-27
➤	Exercises	15-28
➤	Exercise hints	15-29
➤	Exercise solutions	15-30
•	16 - MORE awk:	
➤	Overview	16-2
➤	Conditional logic	16-3
➤	The while loop	16-7
➤	The for loop	16-11
➤	awk functions	16-13
➤	String manipulation	16-14
➤	Other awk things not covered here	16-20
➤	Exercises	16-21
➤	Exercise solutions	16-22
•	17 - ADVANCED I/O:	
➤	Overview	17-2
➤	exec and file descriptors 3 through 9	17-4
➤	Co-processes	17-8
➤	Named pipes	17-14
➤	Exercises	17-19
➤	Exercise solutions	17-20
•	APPENDIX: WHAT YOU SHOULD ALREADY KNOW:	

➤	What you should know	A-1
➤	vi editor summary	A-7
➤	Regular expression metacharacter summary	A-12
➤	Finding patterns with grep	A-13
➤	Extracting fields with cut	A-17

THIS PAGE IS INTENTIONALLY BLANK