

# Rexx <--> Bash

by H. Fosdick © 2024

CC BY-ND [RexxInfo.org](http://RexxInfo.org)

These charts enumerate equivalences between Rexx and Bash. They're for those who know one of these languages and want to learn about the other. (They're **not** about "which language is better".)

## The Basics

|                                  | Rexx            | Bash                                  |
|----------------------------------|-----------------|---------------------------------------|
| Easy to learn, use, and maintain | Yes             | No (unfriendly syntax)                |
| Very powerful                    | Yes             | Yes                                   |
| Open source                      | Yes             | Yes                                   |
| Portable                         | Yes             | Yes                                   |
| Runs on all platforms            | Yes             | Yes                                   |
| Runs as the OS Shell             | No *            | Yes                                   |
| Interfaces to tons of tools      | Yes             | Yes                                   |
| ANSI or ISO Standard             | Yes (ANSI-1996) | Yes (POSIX compliant with extensions) |

\*Regina Rexx can run in the same process as the Z shell (zsh)

## Profiles

|   | Rexx   | Bash  |
|---|--|---|
| Dialects  | TRL-2, ANSI, Mainframe, ooRexx, NetRexx  | Bash (a superset of the Bourne shell)   |
| Unique Usage  | * Default scripting language for mainframes and several minor platforms<br>* Interfaces to <a href="#">all mainframe environments</a> and address spaces | * Default scripting language for Linux (including on Windows and mainframe Linux systems)<br>* <a href="#">Sometimes the default</a> for BSD, Oracle Solaris, older Apples, other systems |
| Programming paradigms   | Procedural, scripting, functional, object-oriented (ooRexx and NetRexx)  | Procedural, scripting, functional   |
| OOP: classes, objects, multi-inheritance, polymorphism, encapsulation | In ooRexx and NetRexx  | Unsupported   |
| User Group  | <a href="#">Rexx Language Association</a>  | <a href="#">Free Software Foundation</a>  |
| Quick Online Lookup   | <a href="#">Quick Lookup</a>   | <a href="#">DevHints</a> , <a href="#">LinuxTutorials</a>   |
| Cheat Sheet (printable PDF)   | <a href="#">ANSI Rexx</a> , <a href="#">Mainframe Rexx</a>   | <a href="#">LinuxSimply</a> , <a href="#">Cheatography</a>  |
| Forum   | <a href="#">RexxLA forum</a>   | <a href="#">LinuxQuestions.org</a> , <a href="#">Linux.org</a>  |
| Further information   | <a href="#">RexxInfo.org</a>   | <a href="#">Free Software Foundation: GNU Bash</a>  |

# Language Comparison (Covers ANSI-standard "classic Rexx")

|                         | ANSI Rexx  | Bash   |
|-------------------------|--|--|
| Formatting              | Free form  | Free form  |
| Case-sensitive          | No   | Yes  |
| Comments                | Enclose inside /* and */   | Start comment with: # Or, use a Here document for commenting out multiple lines  |
| Line Continuation       | , (comma)  | \ (backslash)  |
| Statement Separator     | ; (semi-colon)   | ; (semi-colon)   |
| Code Blocks             | Define by do - end   | Define by do - done, if - fi, or case - esac   |
| Undefined Variables     | Allowed. Use SYMBOL to determine if a variable has been defined  | Allowed. To error on undefined variables, code: set -u   |
| Assignment Operators    | =  | = += -= *= /= %= Compound<br>Bitwise: &=  = <<= >>= ^=   |
| Arithmetic Operators    | + - * / % ** //  | + - * / % **<br>Compound: += -= *= /= %=<br>Evaluate arithmetic expression:<br>\$(( expression ))<br>Increment/decrement a value:<br>i++ i-- ++i --i |
| Comparison Operators    | == \== >> << >>= \<< <<= \>> = \<br>= <> >< > < >= \< <= \><br>( \ can be replaced with ~ in any of these) | Integers: -eq -ne -gt -ge -lt -le<br>Strings: = == != > >= < <= =~   |
| Logical Operators       | &   && \ (prefix) ~ (prefix)   | &&    ! (prefix)   |
| Concatenation Operators | Or, concatenate with blank between<br>Or, concatenate by abuttal (no blank)                                | += Or, use abuttal:<br>VAR3="\$VAR1\$VAR2"   |
| Bitwise Operators       | Use built-in functions   | &   ^ << >> ~<br>Compound: &=  = <<= >>= ^=  |
| Membership Operators    | Unsupported  | Unsupported  |
| String Parsing          | PARSE instruction  | Use regular expressions  |
| Regular Expressions     | Use RexxRE Regular Expressions external Library  | Yes  |
| Built-in Functions      | About 70 functions   | None in the traditional sense, designed to issue shell and line commands   |
| Data Types              | Everything's a string, types are reflected in usage  | By default, variables are untyped. Or use "declare" to explicitly type them as: -a, -A, -i, -l, -n, -r, -t, -u, -x                                   |
| To Check Data Type      | datatype   | declare -p   |
| Associative Arrays      | Use compound variables   | declare -A array_name  |

|                                 |   |   |
|---------------------------------|---|---|
| Multidimensional Arrays         | Use compound variables  | Unsupported   |
| Stack & Queue Operations        | Yes (push, pull, parse pull, queue, queued)   | No generalized facility (offers a directory stack with pushd, popd, and dirs) |
| Decimal Arithmetic              | Default   | Install and use: bc   |
| Flow of Control                 | if, do, select, call, exit, return, iterate, leave, signal, nop                                       | until, while, for, break, continue, return, exit                              |
| Trace Script Execution          | trace (instruction), trace (function)   | Use the -x option.<br>Run the script: bash -x<br>or inside the script: set -x |
| Terminate Process               | exit  | exit  |
| Get User Input                  | say "Enter your name:"<br>parse pull name   | echo -n "Enter your name: "<br>read name                                      |
| Exception Handling              | signal  | trap  |
| Standard Exceptions             | novalue, error, failure, halt, notready, syntax, lostdigits   | trap covers 64 signal specs<br>(list them by: trap -l)                        |
| Run an Operating System Command | Just issue the command string<br>(Rexx passes unrecognized strings to the default active environment) | Just issue the command string   |

Based on [Rexx Programmer's Reference](#) and [Bash Reference Manual](#) version 5.2. Thank you to several RexxLA members who reviewed and improved this cheat sheet.