POSIX shell cheat sheet

Reference syntax and commands for POSIX shell scripting.

Tuesday, 24 September 2019

1 Conditional expressions

For use in `[ ] if [ ]; then` and `test`.

Note: When writing a bash or zsh script, use `[[` instead of POSIX `[`.

1A Variable conditionals

Use these to check if a variable is empty or non-empty.

Expression	Value	Description
`-n`	`var`	If the length of string is non-zero.
`-z`	`var`	If the length of string is zero.

1B Variable comparisons

String:

Expression	Description
`var1 = var2`	Equal to.
`var1 != var2`	Not equal to.

Note: Use == instead of = inside of `[[` for bash, zsh scripts.

Numeric:

Expression	Description
`var1 -eq var2`	Equal to.
`var1 -ne var2`	Not equal to.
`var1 -gt var2`	Greater than.
`var1 -ge var2`	Greater than or equal to.
`var1 -lt var2`	Less than.

Expression	Description
`var1 -le var2`	Less than or equal to.

1C File conditionals

Common:

Expression	Value	Description
`-e`	`file`	If file exists and is any type.
`-f`	`file`	If file exists and is a regular file.
`-d`	`file`	If file exists and is a directory.
`-h/-L`	`file`	If file exists and is a symbolic link.
`-r`	`file`	If file exists and is readable.
`-w`	`file`	If file exists and is writable.
`-x`	`file`	If file exists and is executable.
`-s`	`file`	If file exists and has non-zero size (is non-empty).

Rare:

Expression	Value	Description
`-b`	`file`	If file exists and is a block special file.
`-c`	`file`	If file exists and is a character special file.
`-g`	`file`	If file exists and its set-group-id bit is set.
`-p`	`file`	If file exists and is a named pipe (*FIFO*).
`-t`	`fd`	If file descriptor is open and refers to a terminal.
`-u`	`file`	If file exists and its set-user-id bit is set.
`-S`	`file`	If file exists and is a socket.

2 Operators

2A Assignment

Operator	Description
`=`	Initialize or change the value of a variable.

2B Logical

Operator	Description
`!`	NOT
`&&`	AND
`||`	OR

2C Arithmetic

Operator	Description
`+`	Addition
`-`	Subtraction
`*`	Multiplication
`/`	Division
`**`	Exponentiation

Operator	Description
`%`	Modulo
`+=`	Plus-Equal
`-=`	Minus-Equal
`*=`	Times-Equal
`/=`	Slash-Equal
`%=`	Mod-Equal

3 Parameter expansion

Use these in place of `awk` or `sed` calls when possible.

3A Replacement

Parameter	Description
`${VAR//PATTERN/REPLACE}`	Substitute pattern with replacement.
`${VAR#PATTERN}`	Remove shortest match of pattern from start.
`${VAR##PATTERN}`	Remove longest match of pattern from start.
`${VAR%PATTERN}`	Remove shortest match of pattern from end.
`${VAR%%PATTERN}`	Remove longest match of pattern from end.

3B Length

Parameter	Description
`${#VAR}`	Length of var in characters.

3C Default Value

Parameter	Description
`${VAR:-STRING}`	If `VAR` is empty or unset, use `STRING` as its value.
`${VAR-STRING}`	If `VAR` is unset, use `STRING` as its value.
`${VAR:=STRING}`	If `VAR` is empty or unset, set the value of `VAR` to `STRING`.
`${VAR=STRING}`	If `VAR` is unset, set the value of `VAR` to `STRING`.
`${VAR:+STRING}`	If `VAR` is not empty, use `STRING` as its value.
`${VAR+STRING}`	If `VAR` is set, use `STRING` as its value.
`${VAR:?STRING}`	Display an error if empty or unset.
`${VAR?STRING}`	Display an error if unset.

4 Escape sequences

4A Text Colors

Note: Sequences using RGB values only work in 24-bit true-color mode.

Sequence	Description	Value
`\033[38;5;<NUM>m`	Set text foreground color.	`0-255`
`\033[48;5;<NUM>m`	Set text background color.	`0-255`
`\033[38;2;<R>;<G>;<B>m`	Set text foreground color to RGB color.	`R`, `G`, `B`
`\033[48;2;<R>;<G>;<B>m`	Set text background color to RGB color.	`R`, `G`, `B`

4B Text Attributes

Sequence	Description
`\033[m`	Reset text formatting and colors.
`\033[1m`	Bold text.
`\033[2m`	Faint text.
`\033[3m`	Italic text.
`\033[4m`	Underline text.
`\033[5m`	Slow blink.
`\033[7m`	Swap foreground and background colors.
`\033[8m`	Hidden text.
`\033[9m`	Strike-through text.

5 Internal and environment variables

Variable	Description
`$-`	Shell options
`$$`	Current shell PID

6 References

pure sh bible