

POSIX-2024

Shell & Utilities

Utilities

NAME

awk – pattern scanning and processing language

SYNOPSIS

awk [-F *sepstring*] [-v *assignment*]... *program* [*argument*...]
awk [-F *sepstring*] -f *progfile* [-f *progfile*]... [-v *assignment*]... [*argument*...]

DESCRIPTION

OPTIONS

OPERANDS

STDIN

INPUT FILES

ENVIRONMENT VARIABLES

LANG	LC_MESSAGES
LC_ALL	LC_NUMERIC
LC_COLLATE	NLSPATH
LC_CTYPE	PATH

ASYNCHRONOUS EVENTS

STDOUT

STDERR

OUTPUT FILES

EXTENDED DESCRIPTION

Overall Program Structure

pattern { *action* }

Expressions in awk

(<i>expr</i>)	<i>expr</i> < <i>expr</i>
<i>\$expr</i>	<i>expr</i> <= <i>expr</i>
	<i>expr</i> != <i>expr</i>
<i>lvalue</i> ++	<i>expr</i> == <i>expr</i>
<i>lvalue</i> --	<i>expr</i> > <i>expr</i>
	<i>expr</i> >= <i>expr</i>
++ <i>lvalue</i>	
-- <i>lvalue</i>	
<i>expr</i> ^ <i>expr</i>	<i>expr</i> ~ <i>expr</i>
	<i>expr</i> !~ <i>expr</i>
! <i>expr</i>	
+ <i>expr</i>	<i>expr</i> in <i>array</i>
- <i>expr</i>	(<i>index</i>) in <i>array</i>
<i>expr</i> * <i>expr</i>	<i>expr</i> && <i>expr</i>
<i>expr</i> / <i>expr</i>	
<i>expr</i> % <i>expr</i>	<i>expr</i> <i>expr</i>
	<i>expr1</i> ? <i>expr2</i> : <i>expr3</i>
<i>expr</i> + <i>expr</i>	<i>lvalue</i> ^= <i>expr</i>
<i>expr</i> - <i>expr</i>	<i>lvalue</i> %= <i>expr</i>
	<i>lvalue</i> *= <i>expr</i>
<i>expr</i> <i>expr</i>	<i>lvalue</i> /= <i>expr</i>
	<i>lvalue</i> += <i>expr</i>
	<i>lvalue</i> -= <i>expr</i>
	<i>lvalue</i> = <i>expr</i>

Variables and Special Variables

ARGC	NR
ARGV	OFMT
CONVFMT	OFS
ENVIRON	ORS
FILENAME	RLENGTH
FNR	RS
FS	RSTART
NF	SUBSEP

Regular Expressions

(XBD Extended Regular Expressions)

(XBD File Format Notation)

Patterns

Special Patterns

BEGIN
END

Expression Patterns

Pattern Ranges

Actions

(Concepts Derived from the ISO C Standard)

{ statements }
if (*expression*) *statement* [**else** *statement*]
while (*expression*) *statement*
do *statement* **while** (*expression*)
for (*expression*; *expression*; *expression*) *statement*
for (*variable in* *array*) *statement*
break
continue
delete *array*[*index*]
next
nextfile
exit [*expression*]

Output Statements

print
printf

Functions

Arithmetic Functions

atan2(*y,x*)
cos(*x*)
sin(*x*)
exp(*x*)
log(*x*)
sqrt(*x*)
int(*x*)
rand()
srand([*expr*])

String Functions

gsub(*ere, repl*[, *in*])
index(*s, t*)
length([*arg*])
match(*s, ere*)
split(*s, a*[, *fs*])
sprintf(*fmt, expr, expr, ...*)
sub(*ere, repl*[, *in*])
substr(*s, m*[, *n*])
tolower(*s*)
toupper(*s*)

Input/Output and General Functions

close(*expression*)
fflush([*expression*])
expression | **getline** [*var*]
getline
getline *var*
getline [*var*] < *expression*
system(*expression*)

User-Defined Functions

function *name*([*parameter, ...*]) { *statements* }

Grammar

Lexical Conventions

EXIT STATUS

CONSEQUENCES OF ERRORS