

AWK

AWK Cheatsheet

Start learning for free



ShortcutFoo uses a Spaced Repetition System that adapts to your training.



Basics I

\$1	Reference first column
awk '/pattern/ {action}' file	Execute action for matched pattern 'pattern' on file 'file'
;	Char to separate two actions
print	Print current record line
\$0	Reference current record line



Variables I

\$2	Reference second column
FS	Field separator of input file (default whitespace)
NF	Number of fields in current record
NR	Line number of the current record



Basics II

^	Match beginning of field
---	--------------------------

<code>,</code>	Match operator
<code>!</code>	Do not match operator
<code>-F</code>	Command line option to specify input field delimiter
<code>BEGIN</code>	Denotes block executed once at start
<code>END</code>	Denotes block executed once at end
<code>str1 str2</code>	Concat str1 and str2



One-Line Exercises I

<code>awk '{print \$1}' file</code>	Print first field for each record in file
<code>awk '/regex/' file</code>	Print only lines that match regex in file
<code>awk '!/regex/' file</code>	Print only lines that do not match regex in file
<code>awk '\$2 == "foo" file</code>	Print any line where field 2 is equal to "foo" in file
<code>awk '\$2 != "foo" file</code>	Print lines where field 2 is NOT equal to "foo" in file
<code>awk '\$1 ~ /regex/' file</code>	Print line if field 1 matches regex in file
<code>awk '\$1 !~ /regex/' file</code>	Print line if field 1 does NOT match regex in file



Variables II

<code>FILENAME</code>	Reference current input file
<code>FNR</code>	Reference number of the current record relative to current input file
<code>OFS</code>	Field separator of the outputted data (default whitespace)
<code>ORS</code>	Record separator of the outputted data (default newline)

RS

Record separator of input file (default newline)



Variables III

CONVFMT

Conversion format used when converting numbers (default %.6g)

SUBSEP

Separates multiple subscripts (default 034)

OFMT

Output format for numbers (default %.6g)

ARGC

Argument count, assignable

ARGV

Argument array, assignable

ENVIRON

Array of environment variables



Functions I

index(s,t)

Position in string s where string t occurs, 0 if not found

length(s)

Length of string s (or \$0 if no arg)

rand

Random number between 0 and 1

substr(s,index,len)

Return len-char substring of s that begins at index (counted from 1)

srand

Set seed for rand and return previous seed

int(x)

Truncate x to integer value



Functions II

split(s,a,fs)

Split string s into array a split by fs, returning length of a

match(s,r)

Position in string s where regex r occurs, or 0 if not found

sub(r,t,s)

Substitute t for first occurrence of regex r in string s (or \$0 if s not given)

gsub(r,t,s)

Substitute t for all occurrences of regex r in string s



Functions III

system(cmd)

Execute cmd and return exit status

tolower(s)

String s to lowercase

toupper(s)

String s to uppercase

getline

Set \$0 to next input record from current input file.



One-Line Exercises II

awk 'NR!=1{print \$1}' file

Print first field for each record in file excluding the first record

awk 'END{print NR}' file

Count lines in file

awk '/foo/{n++}; END {print n+0}' file

Print total number of lines that contain foo

awk '{total=total+NF};END{print total}' file

Print total number of fields in all lines

awk '/regex/{getline;print}' file

Print line immediately after regex, but not line containing regex in file

awk 'length > 32' file

Print lines with more than 32 characters in file

awk 'NR==12' file

Print line number 12 of file

© 2023 ShortcutFoo, LLC. All rights reserved.