

[< Learn These Shortcuts \(/app/dojos/awk\)](#)

## AWK Cheat Sheet

### Basics I

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

### Variables I

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

### Basics II

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

### One-Line Exercises I

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

### Variables II

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

## Variables III

<b>CONVFMT</b>	Conversion format used when converting numbers (default %.6g)
<b>SUBSEP</b>	Separates multiple subscripts (default 034)
<b>OFMT</b>	Output format for numbers (default %.6g)
<b>ARGC</b>	Argument count, assignable
<b>ARGV</b>	Argument array, assignable
<b>ENVIRON</b>	Array of environment variables

## Functions I

<b>index(s,t)</b>	Position in string s where string t occurs, 0 if not found
<b>length(s)</b>	Length of string s (or \$0 if no arg)
<b>rand</b>	Random number between 0 and 1
<b>substr(s,index,len)</b>	Return len-char substring of s that begins at index (counted from 1)
<b>srand</b>	Set seed for rand and return previous seed
<b>int(x)</b>	Truncate x to integer value

## Functions II

<b>split(s,a,fs)</b>	Split string s into array a split by fs, returning length of a
<b>match(s,r)</b>	Position in string s where regex r occurs, or 0 if not found
<b>sub(r,t,s)</b>	Substitute t for first occurrence of regex r in string s (or \$0 if s not given)
<b>gsub(r,t,s)</b>	Substitute t for all occurrences of regex r in string s

## Functions III

<b>system(cmd)</b>	Execute cmd and return exit status
<b>tolower(s)</b>	String s to lowercase
<b>toupper(s)</b>	String s to uppercase
<b>getline</b>	Set \$0 to next input record from current input file.

## One-Line Exercises II

<b>awk 'NR!=1{print \$1}' file↵</b>	Print first field for each record in file excluding the first record
<b>awk 'END{print NR}' file↵</b>	Count lines in file
<b>awk '/foo/{n++}; END {print n+0}' file↵</b>	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

[< Learn These Shortcuts \(/app/dojos/awk\)](/app/dojos/awk)

[Blog \(/blog\)](/blog)

[About \(/about\)](/about)

[Privacy Policy \(/privacy\)](/privacy)

[Terms of Service \(/terms\)](/terms)

[Pricing \(/app/pricing\)](/app/pricing)

---