# awk quick reference

Awk is a useful unix command/script language that turns text files into records and fields which can be selected to display as kind of an ad hoc database. With *awk* you can perform many manipulations to these fields or records before they are displayed.

**Fields**:

- separated by whitespace, or by regex FS.
- The fields are denoted $1, $2, ..., while $0 refers to the entire line.
- If FS is null, the input line is split into one field per character.

**Records**:

records are separated by \n (new line), or by regex RS..

**Command Forms:**

A pattern-action statement has the form

**pattern** { *action* }

- A missing **pattern** always matches
- A missing { *action* } means print the line
- Pattern-action statements are separated by newlines or semicolons.
- There can be three separate action blocks:

BEGIN { *action* }<br>
{ *action* }<br>
END { *action* }

**Pattern Forms**:

relational expression != ==

/regular expressions/

- Ex: /start/, /stop/

  Print all lines between start/stop pairs.

**Action Forms:**

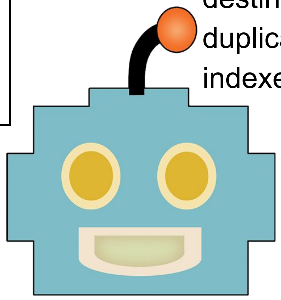
- print *statement*
- If (expression) *statement* [ else *statement* ]
- while (expression) *statement*
- for (expression ; expression) *statement*
- for (var in array) *statement*
- do *statement* while (expression)

**Operators**:

Mathematical: + , - , * , / , cos(), sin(), sqrt(), % modulus

Assignment: = += -= *= /= %= ^=.

FAS RC
38 Oxford St, Rm 206
http://rc.fas.harvard.edu
rchelp@fas.harvard.edu
odybot.rc.fas.harvard.edu
Office hours: M-F, 12-3p



**Syntax:**

using stdin: cat file | awk ***'command'***

using files: awk ***'command'*** file

- Ex: awk '{print $1}' 123.txt

  Prints the first column of 123.txt

**Internal Environmental Variables:**

NF – number of fields in the current record

NR – ordinal number of the current record

FS – regular expression used to separate fields; also settable by option -Ffs (default whitespace)

RS – input record separator (default newline)

OFS – output field separator (default blank)

ORS – output record separator (default newline)

**Assigning Variables**:

You can use the stock $1, $2, $3, … fields and set them to variables in the *action* block.

- Ex: {a=$1,b=$2,c=$3; print a, b, c}

**String Functions**

split(s, a, fs) splits the string s into array elements a[1], a[2], ..., a[n], and returns n. The separation is done with the regular expression fs or with the field separator FS if fs is not given. An empty string as field separator splits the string into one array element per character.

Ex: split($5,t,":");exec_time+=t[1]+t[2]/60+t[3]/3600

sub(r, t, s) substitutes t for the first occurrence of the regular expression r in the string s. If s is not given, $0 is used.

sort(s [, d]) Returns the number of elements in the source array s. The contents of s are sorted using awk's normal rules for comparing values, and the indexes of the sorted values of s are replaced with sequential integers starting with 1. If the optional destination array d is specified, then s is first duplicated into d, and then d is sorted, leaving the indexes of the source array s unchanged.