awk - A Pattern scanning and Processing Language
by Aho, Kernighan and Weinberger
1978, Bell Labs

command syntax:

awk [-F delim] [ -f pgmfile or 'command str'] inputfile

awk processes its input one line at a time, applying commands to
the line when it matches the pattern associated with the command.
Each input line is broken up into fields called $1, $2, ..., based
on the current value of the FS variable. (field separator-see below)
The awk program consists of one or more patterns on separate lines
or separated by semicolons. Each pattern has the form:

    pattern { action }

awk Patterns (like grep, sed, and vi)

In awk, a pattern is an arbitrary Boolean expression involving
regular expressions and relational expressions. Patterns can be any
of the following:

BEGIN
END
/expr/
relational expression
pattern1 & pattern2
(pattern1 || pattern2)
(pattern)
!pattern
pattern1, pattern2

BEGIN and END match before the first line is read, and after the last line
has been read, respectively. All other patterns can contain extended
regular expressions.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;</td>
<td>less than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal to</td>
</tr>
<tr>
<td>==</td>
<td>equal to</td>
</tr>
<tr>
<td>!=</td>
<td>not equal to</td>
</tr>
<tr>
<td>~</td>
<td>means &quot;contains&quot;</td>
</tr>
<tr>
<td>!~</td>
<td>does not contain</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>length</td>
<td>return length of line</td>
</tr>
<tr>
<td>length(x)</td>
<td>return length of x</td>
</tr>
<tr>
<td>int(x)</td>
<td>return integer value of x</td>
</tr>
<tr>
<td>split(str,arr,ch)</td>
<td>split string str into array arr using ch as delimiter, returns number of fields found.</td>
</tr>
<tr>
<td>system(&quot;cmd&quot;)</td>
<td>Runs Unix command cmd and returns exit status.</td>
</tr>
</tbody>
</table>
BUILTIN awk VARIABLES

$0  the entire current input line.
$1,$2,$3,... the fields of the current input line.
NF  total number of fields on current line.
NR  sequence number of current line.
FS  input field separator character (default=blanks)
RS  input record separator (default=NEWLINE)
OFS output field separator character (default=blanks)
FILENAME current input file
ARGV number of arguments in command line.
ARGV list of command line arguments

awk uses the C language syntax for:

printf function

Numerical expression operators: +,-,*,/,%;++--,+-,=-,*=,/=%

Logical Control:

The If Construct                          The While Construct

    if ( expr )                          while ( expr )
        {                                 {
            command                          command
            command                          command
        }...                              }
    else (OR else if )                 else (OR else if )
        { command                          { command
            command                          command
        }

The For construct
for (init step; cond; postbody step)
{ command
  command
  command...
}

Note: individual lines are not terminated by semicolons.
Use semicolon if you want two or more commands on the same line