AWK ::

Awk is a scripting language used for manipulating data and generating reports.The awk command programming language requires no compiling, and allows the user to use variables, numeric functions, string functions, and logical operators.

Awk is a utility that enables a programmer to write tiny but effective programs in the form of statements that define text patterns that are to be searched for in each line of a document and the action that is to be taken when a match is found within a line. Awk is mostly used for pattern scanning and processing. It searches one or more files to see if they contain lines that matches with the specified patterns and then performs the associated actions.

Awk is abbreviated from the names of the developers – Aho, Weinberger, and Kernighan.

**WHAT CAN WE DO WITH AWK ?**

**1. AWK Operations:**  
(a) Scans a file line by line  
(b) Splits each input line into fields  
(c) Compares input line/fields to pattern  
(d) Performs action(s) on matched lines

**2. Useful For:**  
(a) Transform data files  
(b) Produce formatted reports

**3. Programming Constructs:**  
(a) Format output lines  
(b) Arithmetic and string operations  
(c) Conditionals and loops

**awk options 'selection \_criteria {action }' input-file > output-file**

**Default behavior of Awk :**By default Awk prints every line of data from the specified file.

$ awk '{print}' employee.txt

**Print the lines which matches with the given pattern.**

$ awk '/manager/ {print}' employee.txt

**Splitting a Line Into Fields :**For each record i.e line, the awk command splits the record delimited by whitespace character by default and stores it in the $n variables. If the line has 4 words, it will be stored in $1, $2, $3 and $4 respectively. Also, $0 represents the whole line.

$ awk '{print $1,$4}' employee.txt

**Built in Variables in Awk**

Awk’s built-in variables include the field variables—$1, $2, $3, and so on ($0 is the entire line) — that break a line of text into individual words or pieces called fields.

**NR:** NR command keeps a current count of the number of input records. Remember that records are usually lines. Awk command performs the pattern/action statements once for each record in a file.

**NF:** NF command keeps a count of the number of fields within the current input record.

**NF:** NF command keeps a count of the number of fields within the current input record.

**FS:** FS command contains the field separator character which is used to divide fields on the input line. The default is “white space”, meaning space and tab characters. FS can be reassigned to another character (typically in BEGIN) to change the field separator.

**RS:** RS command stores the current record separator character. Since, by default, an input line is the input record, the default record separator character is a newline.

**OFS:** OFS command stores the output field separator, which separates the fields when Awk prints them. The default is a blank space. Whenever print has several parameters separated with commas, it will print the value of OFS in between each parameter.

**ORS:** ORS command stores the output record separator, which separates the output lines when Awk prints them. The default is a newline character. print automatically outputs the contents of ORS at the end of whatever it is given to print.

**Use of NR built-in variables (Display Line Number)**

$ awk '{print NR,$0}' employee.txt

**Use of NF built-in variables (Display Last Field)**

$ awk '{print $1,$NF}' employee.txt

**Another use of NR built-in variables (Display Line From 3 to 6)**

$ awk 'NR==3, NR==6 {print NR,$0}' employee.txt

**To print the first item along with the row number(NR) separated with ” – “ from each line in geeksforgeeks.txt:**

$ awk '{print NR "- " $1 }' geeksforgeeks.txt

**To return the second row/item from geeksforgeeks.txt:**

awk '{print $2}' employee.txt

**To print any non empty line if present**

awk 'NF > 0' geeksforgeeks.txt

**To find the length of the longest line present in the file:**

$ awk '{ if (length($0) > max) max = length($0) } END { print max } ' employee.txt

**5) To count the lines in a file:**

$ awk 'END { print NR }' geeksforgeeks.txt

**Printing lines with more than 10 characters:**

$ awk 'length($0) > 10' geeksforgeeks.txt

**To find/check for any string in any column:**

$ awk '{ if($3 == "B6") print $0;}' geeksforgeeks.txt

**To find/check for any string in any column:**

$ awk '{ if($3 == "B6") print $0;}' geeksforgeeks.txt

**) To print the squares of first numbers from 1 to n say 6:**

$ awk 'BEGIN { for(i=1;i<=6;i++) print "square of", i, "is",i\*i; }'