AWK Cheat Sheet

AWK is usually used to scan the input file line by line.

AWK calls each scanned line a "*record*" and each column of the record a "*field*."

The operation below will print the first field/column of every line/record of file:

```
awk '{print $1}' file
```

Usage: `awk '{/pattern/ {action}' file`

Basic examples:

awk '{print $1}' file	Print the first field of each record of the input file
awk '{print $1;print $2}' file	Print both first and second fields (the char ; separates two actions)
awk '{print $0}' file	Prints the current record ($0 references the entire current record)
awk '/regex/' file	Print only the lines that match the /regex/ in file
awk '!/regex/' file	Print only the lines that do not match the /regex/ in file
awk '$2 == "x"' file	Print only lines where second column is equal to "x"
awk '$2 != "x"' file	Print only lines where second column is different than "x"
awk '$1 ~ /regex/' file	Print lines where first column matches the /regex/ in file
awk '$1 !~ /regex/' file	Print lines where first column does not match the /regex/ in file

Variables:

$N	Reference Nth field
FS	Field separator of input file (default whitespace)
RS	Record separator of input file (default newline)
NF	Number of fields/columns in current record
NR	Row/line number of current record
FILENAME	References the current input file

Functions:

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 argument is passed)
split(s,a,fs)	Splits string *s* into array *a* by field separator *fs* (returns length of a)
match(s,regex)	Position in string *s* where *regex* match occurs (0 if not found)
sub(regex,t,s)	Substitute *t* for first occurrence of *regex* in string *s* ($0 when no *s*)
gsub(regex,t,s)	Substitute *t* for all occurrences of *regex* in string *s*
tolower(s)	String *s* to lowercase
toupper(s)	String *s* to uppercase
substr(s,index,n)	Returns *n*-char substring of *s* that begins at *index* (counted from 1)