

HOW TO USE AWK

Printing a column

awk default delimiter is a space. Meaning that every space is considered a new/different column. The delimiter can be changed with the flag **-F character/word**. To identify a column in syntax is with **\$ number**

Example Syntax: `awk '{print $1}' File`

File

```
Dog Plate 5
Cat Bowl 10
Turtle Fork 40
```

Output

```
Dog
Cat
Turtle
```

Example Syntax: `awk '{print $3}' File`

File

```
Dog      Plate      5
Cat      Bowl      10
Turtle   Fork      40
```

Output

```
5
10
40
```

Delimiter Example Syntax: `awk -F , '{print $2}' File`

File

```
Dog,Plate,5
Cat,Bowl,10
Turtle,Fork,40
```

Output

```
Plate
Bowl
Fork
```

Delimiter Example Syntax 2: `awk -F h '{print $1}' hawk`

hawk

I love hawks. The reason is because they are cool birds. Nothing else, but banana bread on the other hand

Output

I love

Delimiter Example Syntax 3: `awk -F h '{print $2}' hawk`

I love hawks. The reason is because they are cool birds. Nothing else, but banana bread on the other hand

Output

awks. T

Delimiter Example Syntax 4: `awk -F h '{print $3}' hawk`

I love hawks. The reason is because they are cool birds. Nothing else, but banana bread on the other hand

Output

e reason is because t

Last Column Example Syntax: `awk '{print $NF}' File`

File

Dog Plate 5 Banana High
Cat Bowl 10 Bread Middle
Turtle Fork 40 Cake Low

Output

High
Middle
Low

Printing Multiple Columns

To print multiple columns just add more columns to the print statement

Example Syntax: `awk '{print $1$2}' File`

File

Michael Tse 2024

```
Dog Plate 5  
Cat Bowl 10  
Turtle Fork 40
```

Output

```
DogPlate  
CatBowl  
TurtleFork
```

If you want to have a space when printing mutiple columns, between column identifier add “,”

Example Syntax: `awk '{print $3,$1}' File`

File

```
Dog Plate 5  
Cat Bowl 10  
Turtle Fork 40
```

Output

```
5 Plate  
10 Bowl  
40 Fork
```

Basic Functions

AWK is able to computer addition (+), subtraction (-), division (/), multiplication (*), etc. These functions can be chained together

Function Example Syntax: `awk '{print 1 + 1}'`

You may have to hit enter a few times to get results because there is no input file

Output

```
2
```

Functions and Columns

To mathematically add columns together just add column identifier to the function

Example Syntax: `awk '{print $1 + $2}' mathTest`

```
mathTest
```

```
1 10  
2 31
```

```
4 14
```

```
10 2
```

Output

```
11
```

```
33
```

```
18
```

```
12
```

Chaining Example Syntax: `awk '{print (($1 * $3) + $2)}' mathTest`

mathTest

```
6 3 8
```

```
10 4 32
```

```
7 2 1
```

```
3 1 3
```

Output

```
51
```

```
324
```

```
9
```

```
10
```

Example with Constant Syntax: `awk '{print $2 / 10}' mathTest`

Takes values from column 2 and then divides them by 10 and prints the results

mathTest

```
10 20
```

```
1 42
```

```
59 23
```

```
19 32
```

Output

```
2
```

```
4.2
```

```
2.3
```

```
3.2
```

+=

The function += is to add every single row in a column to a grand total

Example Syntax: `awk '{*variable*+= $2} END {print *variable*}' bankAccount`

Take the second column and add its rows to variable. **END** stops the += function

bankAccount

1 2 3

4 5 6

19 23 13

11 54 10

Output

84