

Awk Command

In this tutorial we will see examples of how to use the Awk Command which allows us to manipulate text. For example, using a file name AwkLab.data I will do the following...

```
efelix@192.168.68.70:~$ cat AwkLab.data
Samuel Vimes:(510) 548-1278:250:100:175
Ponder Stibbons:(408) 538-2358:155:90:201
Angua von Überwald:(206) 654-6279:250:60:50
Susan Sto Helit:(206) 548-1348:250:100:175
Tiffany Aching:(206) 548-1278:15:188:150
Sacharissa Cripslock:(916) 343-6410:250:100:175
Adora Belle Dearheart:(406) 298-7744:450:300:275
Frodo Baggins:(206) 548-1278:250:80:75
Tom Bombadil:(916) 348-4278:250:100:175
Peregrin Took:(510) 548-5258:50:95:135
Samwise Gamgee:(408) 926-3456:250:168:200
A.A. Milne:(916) 440-1763:175:75:300
Antoine de Saint-Exupery:(123) 978-6432:250:100:175
Adalgrim Took:(345) 978-7684:4673:100:467
Bandobras "Bullroarer" Took:(453) 978-3534:6753:368:4673
Belladonna Took:(123) 978-5754:356:247:175
Eglantine Took:(123) 978-3574:473:475:4367
Mirabella Took:(345) 978-2677:783:563:175
Ferumbras III Took:(563) 978-753:250:100:3457
Gerontius Took:(574) 978-8535:535:678:4562
```

1. Print all the First Names.

- a. Type **awk '{print \$1}' AwkLab.data** Since the first name is the first field in the file by using **\$1** it prints the first field. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk '{print $1}' AwkLab.data
Samuel
Ponder
Angua
Susan
Tiffany
Sacharissa
Adora
Frodo
Tom
Peregrin
Samwise
A.A.
Antoine
Adalgrim
Bandobras
Belladonna
Eglantine
Mirabella
Ferumbras
Gerontius
```

2. Print phone numbers for Tom and Frodo after their names

- a. Type **awk -F':' '/Tom|Frodo/ {print \$1, \$2}' AwkLab.data** Since the fields are using colons as delimiters we will use **-F':'** to separate the fields. The second field (\$2) is the phone number. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk -F':' '/Tom|Frodo/ {print $1, $2}' AwkLab.data
Frodo Baggins (206) 548-1278
Tom Bombadil (916) 348-4278
```

3. Print Peregrin's full name and phone number area code only.

- a. Type **awk -F'[:()]' '/Peregrin/ {print \$1, \$2, \$3}' AwkLab.data** Since the fields are using both colons and parentheses as delimiters we will use **-F'[:()]'** to separate the fields. By using the **\$** we are printing the first, last and area code. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk -F'[:()]' '/Peregrin/ {print $1, $2, $3}' AwkLab
.data
Peregrin Took 510
```

4. Print all phone numbers (full number) in the 123 area code along with the names
- a. Type **awk -F'[:()]' '/^(123\)/ {print \$1, \$3, \$4}' AwkLab.data** As before use the **-F** switch to let the command your delimiters. Use **/^(123\)/** to have awk just find 123 area codes. Since there is a space between the area code and the phone number we will also print the third field by using **\$3**. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk -F'[:()]' '/^(123\)/ {print $1, $3, $4}' AwkLab.data
Antoine de Saint-Exupery 123 978-6432
Belladonna Took 123 978-5754
Eglantine Took 123 978-3574
```

5. Print all Last names beginning with either a T or D (careful of middle names!)
- a.
6. Print all first names containing four or less characters.
- a. Type **awk 'length(\$1) <= 4 {print \$1}' AwkLab.data** This works as we are telling the awk command to use a length of **<=4** for the first field(**\$1**) <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource

```
efelix@192.168.68.70:~$ awk 'length($1) <= 4 {print $1}' AwkLab.data
Tom
A.A.
```

7. Print the first names and area codes of all those in the 916 area code.
- a. Type **awk '/^(916\)/ {print \$1}' AwkLab.data** Here using parameters on which to search for by using **/^1** and then telling it to print the first field which in this case is the first name. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk '/^(916\)/ {print $1}' AwkLab.data
Sacharissa
Tom
A.A.
```

8. Print Sacharissa's campaign contributions following her name. Each value should be printed with a leading dollar sign; e.g., \$250 \$100 \$175. 1
- a. Type **awk -F':' '/Sacharissa/ {printf "%s", \$1; for (i=3; i<=NF; i++) printf " %s", \$i; print ""}' AwkLab.data** Since we are using the **-F'** the first and last name become one field. Using **printf "%s", \$1** we are printing the first and last name. On the last part of the command, we are telling it to look at the last 3 fields and put a **\$** in front of it. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk -F':' '/Sacharissa/ {printf "%s", $1; for (i=3; i<=NF; i++) printf " %s", $i; print ""}' AwkLab.data
Sacharissa Cripslock $250 $100 $175
```

9. Print last names followed by a comma and the phone number. Be careful of the last names's format.
- a. Type `awk -F'[:()]' '{split($1, name, " "); print name[length(name)] ", "$3,$4}' AwkLab.data` There are 4 parts to the command. The **-F** as the delimiter. **split** to split the full name into parts using a space to determine it. **print name** so it only uses the last name. The last part of the command is the phone number fields. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk -F'[:()]' '{split($1, name, " "); print name[length(name)] ", "$3,$4}' AwkLab.data
Vimes, 510 548-1278
Stibbons, 408 538-2358
Überwald, 206 654-6279
Helit, 206 548-1348
Aching, 206 548-1278
Cripslock, 916 343-6410
Dearheart, 406 298-7744
Baggins, 206 548-1278
Bombadil, 916 348-4278
Took, 510 548-5258
Gamgee, 408 926-3456
Milne, 916 440-1763
Saint-Exupery, 123 978-6432
Took, 345 978-7684
Took, 453 978-3534
Took, 123 978-5754
Took, 123 978-3574
Took, 345 978-2677
Took, 563 978-753
Took, 574 978-8535
```

10. Print the first and last names of those who contributed more than \$110 in the last month.

Make sure to include their last month contribution amount after the name

- a. Type **awk -F:' '{if (\$5 > 110) print \$1, \$5}' AwkLab.data** We are telling the command to look in the fifth field which is the third contribution and telling it that if it is less than 110 **'{if (\$5 > 100) then print the name and the last contribution print \$1, \$5}'**

<https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk -F:' '{if ($5 > 110) print $1, $5}' AwkLab.data
Samuel Vimes 175
Ponder Stibbons 201
Susan Sto Helit 175
Tiffany Aching 150
Sacharissa Cripslock 175
Adora Belle Dearheart 275
Tom Bombadil 175
Peregrin Took 135
Samwise Gamgee 200
A.A. Milne 300
Antoine de Saint-Exupery 175
Adalgrim Took 467
Bandobras "Bullroarer" Took 4673
Belladonna Took 175
Eglantine Took 4367
Mirabella Took 175
Ferumbas III Took 3457
Gerontius Took 4562
```

11. Print the last names, phone numbers, and first month contribution of those who contributed less than \$150 in the first month.

- a. Type **awk -F:' '{if (\$3 < 150) {split(\$1, name, " "); print name[length(name)], \$2, \$3}}' AwkLab.data** By using the delimiter **-F:'** I know that second field is now the phone number and the third field is the first contribution. So, if the arguments are correct, for example if it's less than \$150 then we are telling the command to print the last name. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk -F:' '{if ($3 < 150) {split($1, name, " "); print name[length(name)], $2, $3}}' AwkLab.data
Aching (206) 548-1278 15
Took (510) 548-5258 50
```

12. Print the first names and contribution of those who contributed between \$10 and \$200 in the first month.

- a. Type **awk -F:' '{if (\$3 >= 10 && \$3 <= 200) {split(\$1, name, " "); print name[1], \$3}}' AwkLab.data** This time we are using the aperi signs to give two parameters for a search. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> as the resource.

```
efelix@192.168.68.70:~$ awk -F:' '{if ($3 >= 10 && $3 <= 200) {split($1, name, " "); print name[1], $3}}' AwkLab.data
Ponder 155
Tiffany 15
Peregrin 50
A.A. 175
```

13. Print the first name, last names and total contributions of those who contributed less than \$700 over the three-month period.

- a. Type **awk -F:' '{if (\$5 < 700) print \$1, \$5}' AwkLab.data** The fifth field is the last contribution, so we are telling it to print the ones less than 700. As you can see below it's quite a few. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource

```
efelix@192.168.68.70:~$ awk -F:' '{if ($5 < 700) print $1, $5}' AwkLab.data
Samuel Vimes 175
Ponder Stibbons 201
Angua von Überwald 50
Susan Sto Helit 175
Tiffany Aching 150
Sacharissa Cripslock 175
Adora Belle Dearheart 275
Frodo Baggins 75
Tom Bombadil 175
Peregrin Took 135
Samwise Gamgee 200
A.A. Milne 300
Antoine de Saint-Exupery 175
Adalgrim Took 467
Belladonna Took 175
Mirabella Took 175
```

14. Print the first names and first letter of the last name, and average contribution of those who had an average contribution of more then \$300

15. Print the last name and area code of those not in the 916 area code.

- a. Type **awk -F:' '{split(\$1, name, " "); if (\$2 != "916") print name[length(name)], \$2}' AwkLab.data** We are using the delimiter to separate the fields with **-F**. Then we are splitting the first field which is the full name to just get the first name. Then we are querying if the area code is not 916 in the second field which is the area code. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk -F:' '{split($1, name, " "); if ($2 != "916") print name[length(name)], $2}' AwkLab.data
Vimes (510) 548-1278
Stibbons (408) 538-2358
Überwald (206) 654-6279
Helit (206) 548-1348
Aching (206) 548-1278
Cripslock (916) 343-6410
Dearheart (406) 298-7744
Baggins (206) 548-1278
Bombadil (916) 348-4278
Took (510) 548-5258
Gamgee (408) 926-3456
Milne (916) 440-1763
Saint-Exupery (123) 978-6432
Took (345) 978-7684
Took (453) 978-3534
Took (123) 978-5754
Took (123) 978-3574
Took (345) 978-2677
Took (563) 978-753
Took (574) 978-8535
```

16. Print each record preceded by the number of the record.

a. Type `awk '{print NR ": " $0}' AwkLab.data` NR is the variable for the current record. We are telling the command to print the rest of the record.

<https://opensource.com/article/19/11/fields-records-variables-awk> was the resource.

```
efelix@192.168.68.70:~$ awk '{print NR ": " $0}' AwkLab.data
1: Samuel Vimes:(510) 548-1278:250:100:175
2: Ponder Stibbons:(408) 538-2358:155:90:201
3: Angua von Überwald:(206) 654-6279:250:60:50
4: Susan Sto Helit:(206) 548-1348:250:100:175
5: Tiffany Aching:(206) 548-1278:15:188:150
6: Sacharissa Cripslock:(916) 343-6410:250:100:175
7: Adora Belle Dearheart:(406) 298-7744:450:300:275
8: Frodo Baggins:(206) 548-1278:250:80:75
9: Tom Bombadil:(916) 348-4278:250:100:175
10: Peregrin Took:(510) 548-5258:50:95:135
11: Samwise Gamgee:(408) 926-3456:250:168:200
12: A.A. Milne:(916) 440-1763:175:75:300
13: Antoine de Saint-Exupery:(123) 978-6432:250:100:175
14: Adalgrim Took:(345) 978-7684:4673:100:467
15: Bandobras "Bullroarer" Took:(453) 978-3534:6753:368:4673
16: Belladonna Took:(123) 978-5754:356:247:175
17: Eglantine Took:(123) 978-3574:473:475:4367
18: Mirabella Took:(345) 978-2677:783:563:175
19: Ferumbas III Took:(563) 978-753:250:100:3457
20: Gerontius Took:(574) 978-8535:535:678:4562
```

17. Print the name and total contribution of each person.

a. Type `awk -F:' '{ split($1, name, " "); total = 0; for (i=3; i<=NF; i++) total += $i; print name[1], name[2], name[length(name)], total}' AwkLab.data` We are telling it to start in the third field (`i=3`) to the end (`i<=NF`) and then give us a total and print it.

<https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk -F:' '{ split($1, name, " "); total = 0; for (i=3; i<=NF; i++) total += $i; print name[1], name[2], name[length(name)], total}' AwkLab.data
Samuel Vimes Vimes 525
Ponder Stibbons Stibbons 446
Angua von Überwald 360
Susan Sto Helit 525
Tiffany Aching Aching 353
Sacharissa Cripslock Cripslock 525
Adora Belle Dearheart 1025
Frodo Baggins Baggins 405
Tom Bombadil Bombadil 525
Peregrin Took Took 280
Samwise Gamgee Gamgee 618
A.A. Milne Milne 550
Antoine de Saint-Exupery 525
Adalgrim Took Took 5240
Bandobras "Bullroarer" Took 11794
Belladonna Took Took 778
Eglantine Took Took 5315
Mirabella Took Took 1521
Ferumbas III Took 3807
Gerontius Took Took 5775
```

18. Add \$10 to Tiffany Aching's first contribution and print her full name and first contribution.

a. Type **awk -F:' '/Tiffany/ {split(\$1, name, " "); first_contribution = \$3 + 10; print name[1], name[2], name[length(name)], first_contribution}' AwkLab.data** We are combining commands from previous examples. Since we know that the first contribution **\$3** field we add \$10 to it. <https://www.geeksforgeeks.org/awk-command-unixlinux-examples/> was the resource.

```
efelix@192.168.68.70:~$ awk -F:' '/Tiffany/ {split($1, name, " "); first_contribution = $3 + 10; print name[1], name[2], name[length(name)], first_contribution}' AwkLab.data
Tiffany Aching Aching 25
```

19. Change Samwise Gamgee's name to Sean Astin

a. Type **awk '{gsub("Samwise Gamgee", "Sean Astin"); print}' AwkLab.data** By using **gsub** we are telling the command for all instances of Samwise Gamgee to change it. [https://www.gnu.org/software/gawk/manual/html_node/String-Functions.html#:~:text=The%20gsub\(\)%20function%20returns%20the%20number%20of%20substitutions%20made.%20If](https://www.gnu.org/software/gawk/manual/html_node/String-Functions.html#:~:text=The%20gsub()%20function%20returns%20the%20number%20of%20substitutions%20made.%20If) was the resource.

```
efelix@192.168.68.70:~$ awk '{gsub("Samwise Gamgee", "Sean Astin"); print}' AwkLab.data
Samuel Vimes: (510) 548-1278:250:100:175
Ponder Stibbons: (408) 538-2358:155:90:201
Angua von Überwald: (206) 654-6279:250:60:50
Susan Sto Helit: (206) 548-1348:250:100:175
Tiffany Aching: (206) 548-1278:15:188:150
Sacharissa Cripslock: (916) 343-6410:250:100:175
Adora Belle Dearheart: (406) 298-7744:450:300:275
Frodo Baggins: (206) 548-1278:250:80:75
Tom Bombadil: (916) 348-4278:250:100:175
Peregrin Took: (510) 548-5258:50:95:135
Sean Astin: (408) 926-3456:250:168:200
A.A. Milne: (916) 440-1763:175:75:300
Antoine de Saint-Exupery: (123) 978-6432:250:100:175
Adalgrim Took: (345) 978-7684:4673:100:467
Bandobras "Bullroarer" Took: (453) 978-3534:6753:368:4673
Belladonna Took: (123) 978-5754:356:247:175
Eglantine Took: (123) 978-3574:473:475:4367
Mirabella Took: (345) 978-2677:783:563:175
Ferumbras III Took: (563) 978-753:250:100:3457
Gerontius Took: (574) 978-8535:535:678:4562
```


20. Write an awk script to do the following (MUST be an awk script not just a bash script or commands on the command line)

(a) Prints first name of the all the Took's followed by their total campaign contributions.

awk -F:' '/Took/{ split(\$1, name, " "); total = 0; for (i=3; i<=NF; i++) total += \$i; print name[1], total}' AwkLab.data

```
efelix@192.168.68.70:~$ awk -F:' '/Took/{ split($1, name, " "); total = 0; for (i=3; i<=NF; i++) total += $i; print name[1], total}' AwkLab.data
Peregrin 280
Adalgrim 5240
Bandobras 11794
Belladonna 778
Eglantine 5315
Mirabella 1521
Ferumbras 3807
Gerontius 5775
```

(b) Print the full names and contributions of anyone who contributed between \$10 and \$200 in the last contribution

awk -F:' '{if (\$NF >= 10 && \$NF <= 200) {split(\$1, name, " "); print \$1, \$NF}}}' AwkLab.data

```
efelix@192.168.68.70:~$ awk -F:' '{if ($NF >= 10 && $NF <= 200) {split($1, name, " "); print $1, $NF}}}' AwkLab.data
Samuel Vimes 175
Angua von Überwald 50
Susan Sto Helit 175
Tiffany Aching 150
Sacharissa Cripslock 175
Frodo Baggins 75
Tom Bombadil 175
Peregrin Took 135
Samwise Gamgee 200
Antoine de Saint-Exupery 175
Belladonna Took 175
Mirabella Took 175
```

```
awk -F':' '{ split($1, name, " "); total = 0; for (i=3; i<=NF; i++) total += $i; avg = total /  
(NF-3); if (avg < 300) print $1, avg}' AwkLab.data
```

```
efelix@192.168.68.70:~$ awk -F':' '{ split($1, name, " "); total = 0;  
    for (i=3; i<=NF; i++) total += $i; avg = total / (NF-3); if (avg <  
300) print $1, avg}' AwkLab.data  
Samuel Vimes 262.5  
Ponder Stibbons 223  
Angua von Überwald 180  
Susan Sto Helit 262.5  
Tiffany Aching 176.5  
Sacharissa Cripslock 262.5  
Frodo Baggins 202.5  
Tom Bombadil 262.5  
Peregrin Took 140  
A.A. Milne 275  
Antoine de Saint-Exupery 262.5
```

Below is the script created. I couldn't group the answers in the way the questions are asked so I had answers identifying who they are associated with.

```
# AwkLab Script
# Written by Eliud Felix
#
# (a) Prints first name of all the Took's followed by their total
# campaign contributions
{
    if ($0 ~ /Took/) {
        split($1, name, " ");
        total = 0;
        for (i = 3; i <= NF; i++) {
            total += $i;
        }
        print "Part (a):", name[1], total;
    }
}
#
# (b) Print the full names and contributions of anyone who
# contributed of anyone who contributed between $10 and $200
# in the last contribution.
{
    if ($NF >= 10 && $NF <= 200) {
        split($1, name, " ");
        print "Part (b):", $1, $NF;
    }
}
#
# (c) Print the full names and average contributions of those
# who contributed less than $300 on average.
{
    total = 0;
    for (i = 3; i <= NF; i++) {
        total += $i;
    }
    avg = total / (NF - 3);
    if (avg < 300) {
        print "Part (c):", $1, avg;
    }
}
#
```

Typed `awk -F':' -f AwkScript.awk AwkLab.data` to execute the script. Below are the results.

```
efelix@192.168.68.59:~$ awk -F':' -f AwkScript.awk AwkLab.data
Part (b): Samuel Vimes 175
Part (c): Samuel Vimes 262.5
Part (c): Ponder Stibbons 223
Part (b): Angua von Überwald 50
Part (c): Angua von Überwald 180
Part (b): Susan Sto Helit 175
Part (c): Susan Sto Helit 262.5
Part (b): Tiffany Aching 150
Part (c): Tiffany Aching 176.5
Part (b): Sacharissa Cripslock 175
Part (c): Sacharissa Cripslock 262.5
Part (b): Frodo Baggins 75
Part (c): Frodo Baggins 202.5
Part (b): Tom Bombadil 175
Part (c): Tom Bombadil 262.5
Part (a): Peregrin 280
Part (b): Peregrin Took 135
Part (c): Peregrin Took 140
Part (b): Samwise Gamgee 200
Part (c): A.A. Milne 275
Part (b): Antoine de Saint-Exupery 175
Part (c): Antoine de Saint-Exupery 262.5
Part (a): Adalgrim 5240
Part (a): Bandobras 11794
Part (a): Belladonna 778
Part (b): Belladonna Took 175
Part (a): Eglantine 5315
Part (a): Mirabella 1521
Part (b): Mirabella Took 175
Part (a): Ferumbras 3807
Part (a): Gerontius 5775
```