Awk Lab

Jayviar Laporte

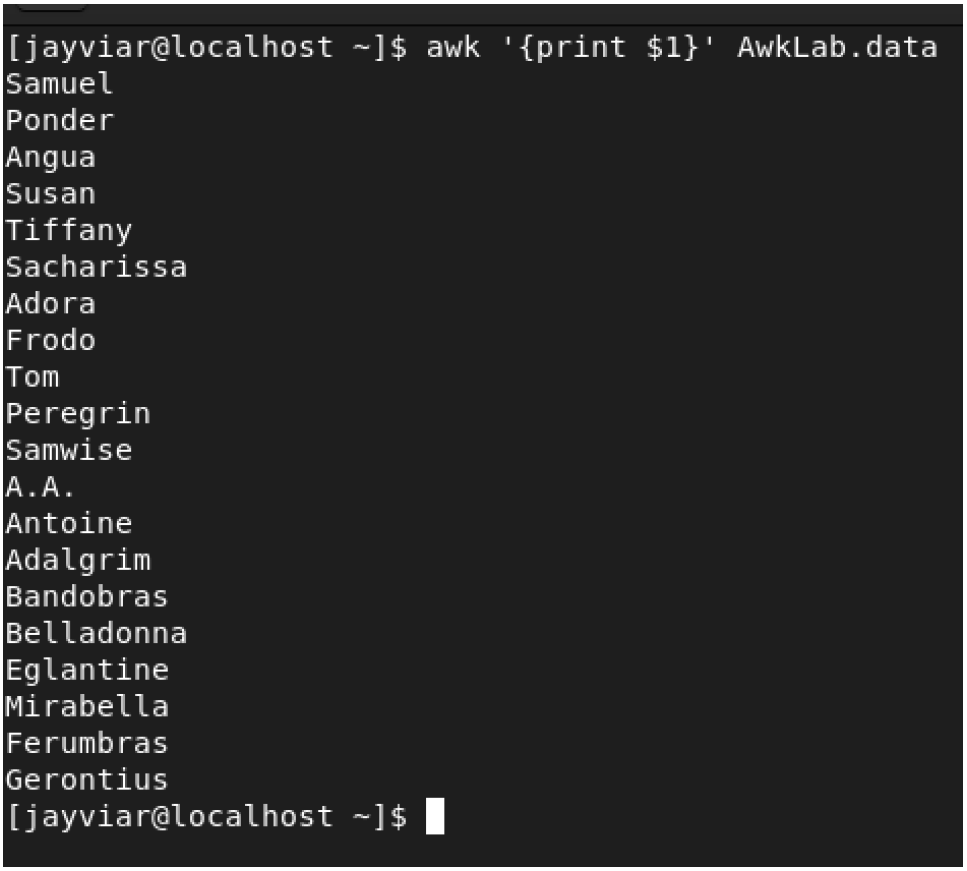
Hello! This is a Doc that will assist anyone using awk with a very wide and large variety of examples of awk being used.

1) Print all the First Names.

Command: awk '{print $1}' AwkLab.data

Explanation: This command gives us all first names because we know the very first field Of Every record is going to be the first name so $1 will give us the first name

Screenshot:



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

Command: awk –F: '/Tom|Frodo/ {print$1$2}' AwkLab.data

Explanation: This command works as we used the literal words Tom and Frodo to limit

Our search to just those two things. We then printed the full name and phone number

Using : as the delimiter and printing the 1st and 2nd field.

Screenshot:

```
[jayviar@localhost ~]$ 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.

Command: awk –F"[:]" '/Peregrin/ {print $1,$2,$3}' AwkLab.data

Explanation: This command works by changing our delimiter to [:] which is a space

AND a colon. This will then be used to our advantage as we can then print fields, 1, 2, and 3 consecutively which will yield us the first and last name as well as the area code

Only.

Screenshot:

```
[jayviar@localhost ~]$ 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

Command: awk –F '/123/ {print $1 $2}' AwkLab.data

Explanation: This command works by setting our delimiter to : and then specifying the area code 123 within the two forward slashes. It then only grabs lines with records containing the area code and then prints out the first and second fields for our full name and phone number

Screenshot:

```
[jayviar@localhost ~]$ awk -F: '/123/ {print $1 $2 }' 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

Command: awk –F":" '{print $1}' AwkLab.data | awk '{print $NF}' | awk '/^T|^D/ {print $1}'

Explanation: So this command may seem long but that is okay! There are multiple ways

To write something that will do the exact same task. For this example, we are starting
off by printing the Full name of every record in the file. We used ":" as our delimiter to do this. Reason being, the full name of every record in the file occurs before the first instance of a ":". With this we take that as our first field and then we pipe that into the second part of our command. The second part of the command takes the full name and since it we didn't specify a delimiter it assumes a space as the delimiter. With this we the $NF command to grab the last field in every record which will always be the last name since we piped the full name into this section. We then end off with piping the
last names into the last section of our command which will use ^T and ^D to print out the records that start with a T or a D.

Screenshot:

```
[jayviar@localhost ~]$ awk -F":" '{print $1}' AwkLab.data | awk '{print $NF}'| awk '/^T|^D/ {print$1}'
Dearheart
Took
Took
Took
Took
Took
Took
Took
Took
```

6) Print all first names containing four or less characters.

Command: awk '{if (length($1) <= 4) print $1}' AwkLab.data

Explanation: For this command. We simply use the length command. Then we specify

What field we want to check the length of. In this case we use the first one with the default delimiter and then use <= 4 to declare we want names with 4 or less characters. then we just print the first field

Screenshot:

```
[jayviar@localhost ~]$ awk '{if (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.

Command: awk '/916/ {print $1,$2}' AwkLab.data | awk –F'[:]' '{print $1, $3}'

Explanation: For this command. We start off by specifying our search to things just containing 916. We then print the first and second field of every record containing that.

We then pipe that into our second part with the delimiter of " :" Which accounts for a

Space and a colon as our two delimiters and out puts the first and third field which gives us the first name and area codes of 916

Screenshot:

```
[jayviar@localhost ~]$ awk '/916/ {print $1,$2}' AwkLab.data | awk -F'[ :]' '{print $1,$3}'
Sacharissa (916)
Tom (916)
A.A. (916)
```

8) Print Sacharissa's campaign contributions following her name. Each value should be printed with a leading dollar sign; e.g., $250 $100 $175.

Command: awk –F'[:] '/Sacharissa/ {print $1,"$"$3,"$"$4,"$"$5"}' AwkLab.data

Explanation: The following code starts with using : as the delimiter. Then specifies within the forwards slashes that we want to look for records with the instance of "Sacharissa". We then print the first field then fields 3-5 with a "$" before each field so we can print a dollar sign before each contribution and also print the full name

Screenshot:

```
[jayviar@localhost ~]$ awk -F'[:]' '/Sacharissa/ {print $1,"$"$3,"$"$4, "$"$5}' AwkLab.data
Sacharissa Cripslock $250 $100 $175
```

9) Print last names followed by a comma and the phone number.

Command: awk –F":" '{print $2,$1}' Awklab.data | awk '{print $NF"," $1,$2}'

Explanation: So for this example. We start off by using the : as our field separator. We then use print the *second and *first fields which we then pipe into the second part of our code and firstly print the $NF as we want to start the line with the last name and then print the first and second field to give us area code and then the rest of our phone number. We also included a comma contained like this "," in between the $NF and $1 as that is what was asked of us.

Screenshot:

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

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

Command: awk –F: '$5 > 110 {print $1,"$"$5}' AwkLab.data

Explanation: For this solution. We are starting off by swapping our field separator for a ":" . Then we immediately ask the server to check for every record where the 5th field is greater than 110. This is due to the fact that we know the 5th field is the final months contridution

For every person in the record. We then just print every instance of this along with the first

And last name and ending with a "$" added to the end right before the final months

Contribution

Screenshot:

```
[jayviar@localhost ~]$ awk -F: '$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
Ferumbras III Took $3457
Gerontius Took $4562
[jayviar@localhost ~]$
```

11) Print the last names, phone numbers, and first month contribution of those who contributed less than $150 in the first month.
Command: awk -F":" '$3 <=150 {print $3,$2,$1}' AwkLab.data | awk '{print $NF, $2,$3,$1}'

Explanation: To break this solution down. We start off by changing our field separator to a ":" . We then immediately check if the the 3rd field is less than or equal to 150 as we know that is our first months contribution. We then print fields 3,2,1 as we want to get the full name and number, as well as the contribution prepared for our pipe. We then pipe this with the default field seperator and print the NF as we want the last name to be at the top of the line and then we print the 2nd, 3rd, and 1st field to print out everything in the correct order.

Screenshot:

```
[jayviar@localhost ~]$ awk -F":" '$3 <=150 {print $3,$2,$1}' AwkLab.data | awk '{print $NF, $2,$3,$1}'
Aching (206) 548-1278 15
Took (510) 548-5258 50
```

12) Print the first names and contribution of those who contributed be- tween $10 and $200 in the first month

Command: awk –F":" '$3 > 10 && $3 <=200 {print $3,$2,$1}' AwkLab.data | awk ' {print$1,$4}'

Explanation: The code above does two things at once. It first checks field 3 and identifies if it is greater than 10. It then checks if field 3 is ALSO less than or equal to 200

at the same time using our and operator also seen as &&. We want both things to be true at once so this Is why we are using && in this situation.

Screenshot:

```
[jayviar@localhost ~]$ awk -F":" '$3 > 10 && $3 <=200 {print $3,$2,$1}' AwkLab.data | awk '{print$1,$4}'
155 Ponder
15 Tiffany
50 Peregrin
175 A.A.
```

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

Command:

Explanation: Like the previous exercise. We are going to check if something is less than a specific number. However, For this solution. We know that fields; 3, 4 and 5 are every months contribution. So, we want to add these together and check if it is less than 700 to make sure

We follow the question being asked. We then print the first and last name along with

The printing fields 3,4 and 5 added together to display what the overall contribution was for each record. To print last only first and last name we also needed to pipe this into awk again and then specify the last field to grab the last name and then print everything again.

Screenshot:

```
[jayviar@localhost ~]$ awk -F":" '$3+$4+$5 < 700 {print $3+$4+$5,$1}' AwkLab.data | awk '{print$2,$NF,$1}'
Samuel Vimes 525
Ponder Stibbons 446
Angua Überwald 360
Susan Helit 525
Tiffany Aching 353
Sacharissa Cripslock 525
Frodo Baggins 405
Tom Bombadil 525
Peregrin Took 280
Samwise Gamgee 618
A.A. Milne 550
Antoine Saint-Exupery 525
[jayviar@localhost ~]$
```

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

Command: awk –F '[:]' '{$3+$4+$5/3)>300 {print ($3+$4+$5/3), $1} AwkLab.data | awk '{print $2, $NF, $1}'

Explanation:For this solution we simply start by doing all of our math. We firstly change the field operator to ": ". We then add all of the contribution fields and then divide them by 3. Next, we print the exact same thing however we specify the math first so that way our name f field purposely will be

the last field. We then pipe that again and then print the first name, Then since our the last name is now the last field, we print that using $NF.

Note: This program does everything except the first letter of the last name. I understand

That what is most likley needed here is some sort of field that only prints the first character

Of the last name field but I was not sure how to get that.

Screenshot:

```
[jayviar@localhost ~]$ awk -F'[:]' '($3+$4+$5/3)>300 {print($3+$4+$5/3), $1}' AwkLab.data | awk '{print $2,$NF,$1}'
Samuel Vimes 408.333
Ponder Stibbons 312
Angua Überwald 326.667
Susan Helit 408.333
Sacharissa Cripslock 408.333
Adora Dearheart 841.667
Frodo Baggins 355
Tom Bombadil 408.333
Samwise Gamgee 484.667
A.A. Milne 350
Antoine Saint-Exupery 408.333
Adalgrim Took 4928.67
Bandobras Took 8678.67
Belladonna Took 661.333
Eglantine Took 2403.67
Mirabella Took 1404.33
Ferumbras Took 1502.33
Gerontius Took 2733.67
```

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

Command: awk –F":" '!/916/ {print $2,$1}' AwkLab.data | awk'{print$NF, $1}'

Explanation: To start with this solution. We change our field separator to ":". We then specify within our forward slashes that we want things that are NOT 916. We then print every second and first field with that information. Then we pipe it with the default field separator using awk again, and this time printing the last field which will be the last name. Then the first field

Which will be the area code.

Screenshot:

```
[jayviar@localhost ~]$ awk -F":" '!/916/ {print $2,$1}' AwkLab.data | awk '{print$NF,$1}'
Vimes (510)
Stibbons (408)
Überwald (206)
Helit (206)
Aching (206)
Dearheart (406)
Baggins (206)
Took (510)
Gamgee (408)
Saint-Exupery (123)
Took (345)
Took (453)
Took (123)
Took (123)
Took (345)
Took (563)
Took (574)
```

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

Command:awk ‘{print NR, $0}’ AwkLab.data

Explanation:This solution is relatively simple. We simply print the NR command which will keep track of the number of records. We then print the entire record. By using $0

Screenshot:

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

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

Command: awk-F":” ‘{print$1, $3+$4+$5}’ AwkLab.data

Explanation: This solution is solved by simply changing our field operator to “:” and then printing the first field and adding the fields 3-5 as those contain each contribution

Screenshot:

```
[jayviar@localhost ~]$ awk -F":" '{print$1, $3+$4+$5}' AwkLab.data
Samuel Vimes 525
Ponder Stibbons 446
Angua von Überwald 360
Susan Sto Helit 525
Tiffany Aching 353
Sacharissa Cripslock 525
Adora Belle Dearheart 1025
Frodo Baggins 405
Tom Bombadil 525
Peregrin Took 280
Samwise Gamgee 618
A.A. Milne 550
Antoine de Saint-Exupery 525
Adalgrim Took 5240
Bandobras "Bullroarer" Took 11794
Belladonna Took 778
Eglantine Took 5315
Mirabella Took 1521
Ferumbras III Took 3807
Gerontius Took 5775
[jayviar@localhost ~]$
```

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

Command: awk –F":" '/Tiffany/ {print $`, "$"$3+10}' AwkLab.data

Explanation: For this Solution. We simply change our field operator to ":", then we specify our search within our forward slashes with the name Tiffany. We then simply print out the first field operator which would be her name and then print out the 3rd field operator which would be her first contribution followed by +10 to add 10 to her first contribution

Screenshot:

```
[jayviar@localhost ~]$ awk -F":" '/Tiffany/ {print$1,"$"$3+10}' AwkLab.data
Tiffany Aching $25
[jayviar@localhost ~]$
```

19) Change Samwise Gamgee's name to Sean Astin

Command: awk '/Samwise Gamgee/ {print $1,$2}' AwkLab.data

Explanation: To further explain this command. We start off by searching specifically for

The name Samwise Gamgee within our forward slashes. We then print the first field, but we also change it by setting its new value to the name Sam Astin. Then we simply print the

First field again.

Screenshot:

```
[jayviar@localhost ~]$ awk '/Samwise Gamgee/ {print $1,$2}' AwkLab.data
Samwise Gamgee:(408)
[jayviar@localhost ~]$ awk '/Samwise Gamgee/{print $1="Sam Astin";$1;}' AwkLab.data
Sam Astin
[jayviar@localhost ~]$
```

20) Write an Awk Script

Command: Code is long so attached is an image

```
  GNU nano 5.6.1
#!/bin/awk -f

BEGIN {FS="[: ]"} {
        if($2 == "Took"){
                {print $1,$5+$6+$7};

        if($7 >= 10 && $7 <= 200)
                {print $1,$2,$7};
        if(($5+$6+$7/3) < 300)
                {print $1,$2,($5+$6+$7/3)};
}
        else{
                if($3 == "Took")
                        {print $1, $6,$7,$8};
                if($8 >= 10 && $8 <=200)
                        {print $1,$2,$3,$8};
                if(($6+$7+$8/3) < 300)
                        {print$1,$2,($6+$7+$8/3)};
}
}
```

Explanation: This solution is a little bit more complex. For starters, We basically want to

Being by setting our field Seperator to space and ":" . We know that since there

Are two being used here we know the fields for both wont be identical. So with this

Knowledge. We Start by catching the records with a shorter amount of

Fields. Which in this case, will be identifying records with the field Seperator We then start off by telling the machine that if we see a second field with the last

Name "Took". We want to print the total contributions, as well as their name . We then ask

The machine to check if their last contribution is between 10 and 200. We then finalize

It by seeing if the average contribution is less than 300. This process is repeated again after our else statement however we are now catching records with the longer amount of fields which forces us to change the values 1 placement higher.

Note(I am not sure Why this is not printing every single thing I assumed it would write

I had a bit of trouble with this one. However I am not sure where my issue lies)

Screenshot: