AWK useful examples

WE'LL COVER THE FOLLOWING

- File content: data.txt
- Example 1. AWK `print` function
- Example 2. AWK `print` specific field
- Example 3. AWK's `BEGIN` and `END` Actions
- Example 4. AWK fields variable (`$1` , `$2` and so on)
- Example 5. AWK built-in variables
- Example 6. AWK fields comparison `>`
- Self-contained AWK scripts example

The best way to learn AWK is probably looking at some examples in a given context, let us create file `data.txt` file which has the following content:

File content: data.txt

Months	Water	Gas	Elec	Phone
Jan	647	2851	3310	1200
Feb	287	3544	25500	1000
Mar	238	3212	21900	1710
Apr	468	6986	35000	2000
May	124	1097	96300	13000

Thanks to educative.io team for helping me to apriori upload this file (data.txt) to the Educative storage space. So, you should be good to go, just press the Run button!

Example 1. AWK `print` function

By default Awk prints every line from the file.

```
awk -F, '{print;}' data.txt | csvlook
```

Action `print` with out any argument prints the whole line by default. So it prints all the lines of the file with out fail. Note that the actions need to be enclosed with in the braces.

Example 2. AWK `print` specific field

```
awk -F, '{print $1,$2}' data.txt
```

Example 3. AWK's `BEGIN` and `END` Actions

```
awk -F, 'BEGIN {print "Period\tBill1\tBill2";}
{print $1,"\t",$2,"\t",$3,"\t",$NF;}
END{print "END\n--------------"; }' data.txt
```

Here, the `actions` specified in the `BEGIN` section executed before AWK starts reading the lines from the input and `END` actions will be performed after completing the reading and processing the lines from the input.

Example 4. AWK fields variable (`$1`, `$2` and so on)

Let's consider we want to find a total of al bills in all months in the data. We then create the following script:

```
#!/bin/bash


echo -n "Field sum: Water + Gas + Electtricty + Phones = "
awk -F "," '{
        if(FNR == 1){
                next;
        }

        Water=$2;
        Gas=$3;
        Electricity=$4;
        Phones=$5;

        fields_sum=Water + Gas + Electtricty + Phones;

        total +=fields_sum;

} END { print total; }' data.txt
```

Note that it's a bash script that calls `awk` from inside and we have used `FNR` to detect the first row which we want to avoid in the sum calculation.

Example 5. AWK built-in variables

As mentioned earlier, the built-in variable `$NF` represents number of field, in this case last field (5):

```
awk -F, '{print $1,$NF;}' data.txt
```

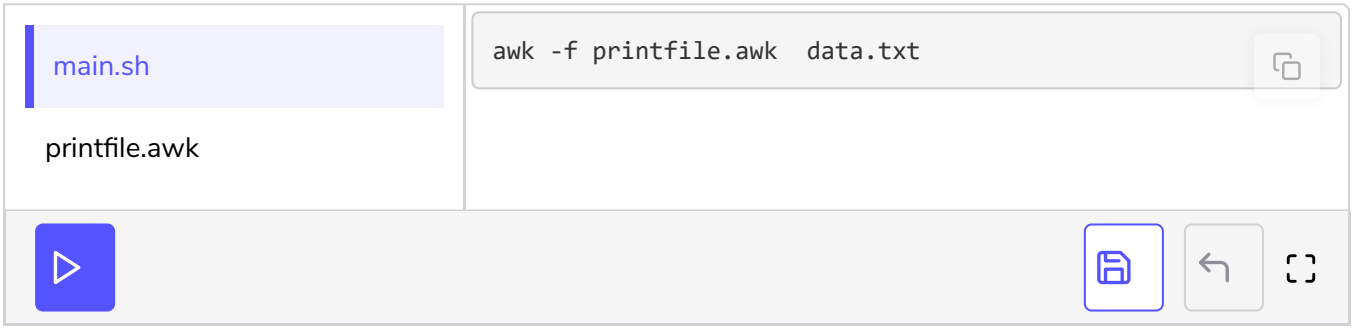
Example 6. AWK fields comparison `>`

Let's now find the months with water `bills > 500`:

```
awk -F, '$2 > 500' data.txt
```

Self-contained AWK scripts example

In Linux systems self-contained AWK scripts can be constructed using. For example, a script that prints the content of a given file may be built by creating a file named `printfile.awk` with the following content:



The `-f` flag tells AWK that the argument that follows is the file to read the AWK program from.