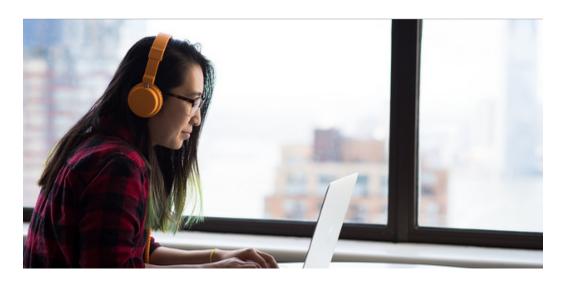10 year anniversary

LOG IN

Main menu

Articles   Resources   Downloads   About

Open Organization

Getting started with awk, a powerful text-parsing tool

Let's jump in and start using it.

30 Oct 2019 | Seth Kenlon (Red Hat) (/users/seth-morriss) | Dave Morriss (/users/dave-morriss) | Robert Young (/users/ryoung29) | 75 | 4 comments

Awk is a powerful text-parsing tool for Unix and Unix-like systems, but because it has programmed functions that you can use to perform common parsing tasks, it's also considered a programming language. You probably won't be developing your next GUI application with awk, and it likely won't take the place of your default scripting language, but it's a powerful utility for specific tasks.

What those tasks may be is surprisingly diverse. The best way to discover which of your problems might be best solved by awk is to learn awk; you'll be surprised at how awk can help you get more done but with a lot less effort.

Awk's basic syntax is:

```
awk [options] 'pattern {action}' file
```

To get started, create this sample file and save it as **colours.txt**

```
name        color  amount
apple       red    4
banana      yellow 6
strawberry  red    3
grape       purple 10
apple       green  8
plum        purple 2
kiwi        brown  4
potato      brown  9
pineapple   yellow 5
```

This data is separated into columns by one or more spaces. It's common for data that you are analyzing to be organized in some way. It may not always

We use cookies on our websites to deliver our online services. Details about how we use cookies and how you may disable them are set out in our Privacy Statement. By using this website you agree to our use of cookies.   ✕

pattern. You can use patterns of data to help awk extract and process the data that you want to focus on.

Printing a column

In awk, the **print** function displays whatever you specify. There are many predefined variables you can use, but some of the most common are integers designating columns in a text file. Try it out:

```
$ awk '{print $2;}' colours.txt
color
red
yellow
red
purple
green
purple
brown
brown
yellow
```

In this case, awk displays the second column, denoted by **$2**. This is relatively intuitive, so you can probably guess that **print $1** displays the first column, and **print $3** displays the third, and so on.

To display *all* columns, use **$0**.

The number after the dollar sign (**$**) is an *expression*, so **$2** and **$(1+1)** mean the same thing.

Conditionally selecting columns

a header, and the columns relate directly to one another. By defining *conditional* requirements, you can qualify what you want awk to return when looking at this data. For instance, to view items in column 2 that match "yellow" and print the contents of column 1:

```
awk '$2=="yellow"{print $1}' colours.txt
banana
pineapple
```

Regular expressions work as well. This conditional looks at **$2** for approximate matches to the letter **p** followed by any number of (one or more) characters, which are in turn followed by the letter **p**:

```
$ awk '$2 ~ /p.+p/ {print $0}' colours.txt
grape    purple  10
plum     purple  2
```

Numbers are interpreted naturally by awk. For instance, to print any row with a third column containing an integer greater than 5:

```
awk '$3>5 {print $1, $2}' colours.txt
name     color
banana   yellow
grape    purple
apple    green
potato   brown
```

Field separator

By default, awk uses whitespace as the field separator. Not all text files use

```
name,color,amount
apple,red,4
banana,yellow,6
strawberry,red,3
grape,purple,10
apple,green,8
plum,purple,2
kiwi,brown,4
potato,brown,9
pineapple,yellow,5
```

Awk can treat the data in exactly the same way, as long as you specify which character it should use as the field separator in your command. Use the **--field-separator** (or just **-F** for short) option to define the delimiter:

```
$ awk -F"," '$2=="yellow" {print $1}' file1.csv
banana
pineapple
```

Saving output

Using output redirection, you can write your results to a file. For example:

```
$ awk -F, '$3>5 {print $1, $2} colours.csv > output.txt
```

This creates a file with the contents of your awk query.

More Linux resources

- Linux commands cheat sheet (https://developers.redhat.com/cheat-

/cheat-sheets/advanced-linux-commands
/?intcmp=70160000000h1jYAAQ&utm_source=intcallout&
utm_campaign=linuxcontent)

- Free online course: RHEL Technical Overview (https://www.redhat.com
  /en/services/training/rh024-red-hat-linux-technical-
  overview?intcmp=70160000000h1jYAAQ&utm_source=intcallout&
  utm_campaign=linuxcontent)

- Linux networking cheat sheet (https://opensource.com/downloads/cheat-
  sheet-networking?intcmp=70160000000h1jYAAQ&
  utm_source=intcallout&utm_campaign=linuxcontent)

- SELinux cheat sheet (https://opensource.com/downloads/cheat-sheet-
  selinux?intcmp=70160000000h1jYAAQ&utm_source=intcallout&
  utm_campaign=linuxcontent)

- Linux common commands cheat sheet (https://opensource.com
  /downloads/linux-common-commands-cheat-
  sheet?intcmp=70160000000h1jYAAQ&utm_source=intcallout&
  utm_campaign=linuxcontent)

- What are Linux containers? (https://opensource.com/resources/what-are-
  linux-containers?intcmp=70160000000h1jYAAQ&utm_source=intcallout&
  utm_campaign=linuxcontent)

- Our latest Linux articles (https://opensource.com
  /tags/linux?intcmp=70160000000h1jYAAQ&utm_source=intcallout&
  utm_campaign=linuxcontent)

You can also split a file into multiple files grouped by column data. For example, if you want to split colours.txt into multiple files according to what color appears in each row, you can cause awk to redirect *per query* by including the redirection in your awk statement:

This produces files named **yellow.txt**, **red.txt**, and so on.

In the next article, you'll learn more about fields, records, and some powerful awk variables.

This article is adapted from an episode of Hacker Public Radio (http://hackerpublicradio.org/eps.php?id=2114), a community technology podcast.

 (/article/18/7/cheat-sheet-awk)

Extracting and displaying data with awk (/article/18/7/cheat-sheet-awk)

Get our awk cheat sheet.

Jim Hall (Correspondent) (/users/jim-hall)

Topics :   **Programming** (/tags/programming)   **Linux** (/tags/linux)

About the author

Seth Kenlon - Seth Kenlon is an independent multimedia artist, free culture advocate, and UNIX geek. He has worked in the film

/seth)   He is one of the maintainers of the Slackware-based multimedia production
                project, http://slackermedia.info (http://slackermedia.info)

                • More about me (/users/seth)

About the author

Robert Young -

• More about me (/users/ryoung29)
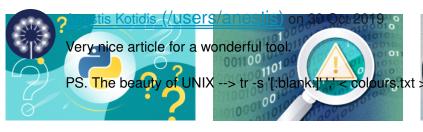
(/users
/ryoung29)

About the author

Dave Morriss - Dave Morriss is a retired IT Manager based in Edinburgh,
Scotland. He worked in the UK higher education sector providing IT
services to students and staff. He has a BSc degree in Zoology and spent
some time working towards a PhD in Animal Behaviour. However, the
prospect of working in IT proved to be more attractive and he moved to
Lancaster University (in Lancashire, UK) in the mid 1970's where he worked
as a Systems Programmer for five years. At Lancaster he worked with the
University's ICL...

(/users
/dave-
morriss)

• More about me (/users/dave-morriss)

Recommended reading

Anestis Kotidis (/users/anestis) on 30 Oct 2019

Very nice article for a wonderful tool.

PS. The beauty of UNIX --> tr -s '[:blank:]' ',' < colours.txt > colours.csv

Use Python to solve a charity's business problem (/article /20/9/solve-problem-python?utm_campaign=intrel)

Seth Kenlon (/users/seth) on 30 Oct 2019

Thanks! And yeah, not just the beauty of UNIX - the beauty of UNIX *one-liners*. It's amazing!

How this open source project tests your .NET code, by evolves with .NET (/article/20/9/testing-net-fixie?utm_campaign=intrel)

Manage your SSH connections with this open source tool (/article/20/9/ssh-connection-manager?utm_campaig

Stefano Figura on 14 Nov 2019

Thanks for the nice article!

One minor correction:

`awk '$2=="yellow"{print $1}' file1.txt`

should be:

`awk '$2=="yellow"{print $1}' colours.txt`

A beginner's guide to SSH for remote connection on Linux (/article /20/9/ssh?utm_campaign=intrel)

Program hardware from the Linux command line (/article /20/9/hardware-command-line?utm_campaign=intrel)

Managing a non-profit organization's supply chain with Groovy (/article /20/9/groovy?utm_camp

Seth Kenlon (/users/seth) on 15 Nov 2019

Good catch! Fixed it, thanks.

We use cookies on our websites to deliver our online services. Details about how we use cookies and how you may disable them are set out in our Privacy Statement. By using this website you agree to our use of cookies.

✕

Enter your email address...

Select your country or region

Subscribe

[Privacy Statement](#)

Get the highlights in your inbox every week.

Find us:

[Privacy Policy](#) | [Terms of Use](#) | [Contact](#) | [Meet the Team](#) | [Visit opensource.org](#)