

10 year anniversary

[LOG IN](#)

Main menu

[Articles](#)[Resources](#)[Downloads](#)[About](#)[Open Organization](#)

Fields, records, and variables in awk

In the second article in this intro to awk series, learn about fields, records, and some powerful awk variables.

04 Nov 2019 | [Seth Kenlon \(Red Hat\) \(/users/seth\)](#) | 88 | [1 comment](#)



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.



Image credits :

[Pixabay](https://pixabay.com/en/guy-man-working-laptop-computer-2618928/) (<https://pixabay.com/en/guy-man-working-laptop-computer-2618928/>), [CC0](https://pixabay.com/en/service/terms/#usage) (<https://pixabay.com/en/service/terms/#usage>).

Awk comes in several varieties: There is the original **awk**, written in 1977 at AT&T Bell Laboratories, and several reimplementations, such as **mawk**, **nawk**, and the one that ships with most Linux distributions, GNU awk, or **gawk**. On most Linux distributions, awk and gawk are synonyms referring to GNU awk, and typing either invokes the same awk command. See the [GNU awk user's guide](https://www.gnu.org/software/gawk/manual/html_node/History.html#History) (https://www.gnu.org/software/gawk/manual/html_node/History.html#History) for the full history of awk and gawk.

The [first article](https://opensource.com/article/19/10/intro-awk) (<https://opensource.com/article/19/10/intro-awk>) in this series showed that awk is invoked on the command line with this syntax:

```
$ awk [options] 'pattern {action}' inputfile
```

Awk is the command, and it can take options (such as **-F** to define the field separator). The action you want awk to perform is contained in single quotes, at least when it's issued in a terminal. To further emphasize which part of the awk command is the action you want it to take, you can precede your program with the **-e** option (but it's not required):

```
$ awk -F, -e '{print $2;}' colours.txt
yellow
blue
green
[...]
```

— - - - -

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.



Programming and development

- [Red Hat Developers Blog \(https://developers.redhat.com/?intcmp=7016000000127cYAAQ&src=programming_resource_menu4\)](https://developers.redhat.com/?intcmp=7016000000127cYAAQ&src=programming_resource_menu4)
- [Programming cheat sheets \(https://opensource.com/downloads/cheat-sheets?intcmp=7016000000127cYAAQ\)](https://opensource.com/downloads/cheat-sheets?intcmp=7016000000127cYAAQ)
- [Try for free: Red Hat Learning Subscription \(https://www.redhat.com/en/services/training/learning-subscription?src=programming_resource_menu3\)](https://www.redhat.com/en/services/training/learning-subscription?src=programming_resource_menu3)
- [New Python content \(https://opensource.com/tags/python?src=programming_resource_menu1\)](https://opensource.com/tags/python?src=programming_resource_menu1)
- [Our latest JavaScript articles \(https://opensource.com/tags/javascript?src=programming_resource_menu2\)](https://opensource.com/tags/javascript?src=programming_resource_menu2)

Awk views its input data as a series of *records*, which are usually newline-delimited lines. In other words, awk generally sees each line in a text file as a new record. Each record contains a series of *fields*. A field is a component of a record delimited by a *field separator*.

By default, awk sees whitespace, such as spaces, tabs, and newlines, as indicators of a new field. Specifically, awk treats multiple *space* separators as one, so this line contains two fields:

raspberry red

As does this one:

.

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.



is a comma, the following example record contains three fields, with one probably being zero characters long (assuming a non-printable character isn't hiding in that field):

```
a, ,b
```

The awk program

The *program* part of an awk command consists of a series of rules. Normally, each rule begins on a new line in the program (although this is not mandatory). Each rule consists of a pattern and one or more actions:

```
pattern { action }
```

In a rule, you can define a pattern as a condition to control whether the action will run on a record. Patterns can be simple comparisons, regular expressions, combinations of the two, and more.

For instance, this will print a record *only* if it contains the word "raspberry":

```
$ awk '/raspberry/ { print $0 }' colours.txt  
raspberry red 99
```

If there is no qualifying pattern, the action is applied to every record.

Also, a rule can consist of only a pattern, in which case the entire record is written as if the action was **{ print }**.

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.



The NF variable

Each field has a variable as a designation, but there are special variables for fields and records, too. The variable **NF** stores the number of fields awk finds in the current record. This can be printed or used in tests. Here is an example using the [text file \(https://opensource.com/article/19/10/intro-awk\)](https://opensource.com/article/19/10/intro-awk) from the previous article:

```
$ awk '{ print $0 " (" NF ")" }' colours.txt
name      color  amount (3)
apple     red    4 (3)
banana    yellow 6 (3)
[...]
```

Awk's **print** function takes a series of arguments (which may be variables or strings) and concatenates them together. This is why, at the end of each line in this example, awk prints the number of fields as an integer enclosed by parentheses.

The NR variable

In addition to counting the fields in each record, awk also counts input records. The record number is held in the variable **NR**, and it can be used in the same way as any other variable. For example, to print the record number before each line:

```
$ awk '{ print NR ": " $0 }' colours.txt
1: name      color  amount
2: apple     red    4
3: banana    yellow 6
```

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.



Note that it's acceptable to write this command with no spaces other than the one after **print**, although it's more difficult for a human to parse:

```
$ awk '{print NR": "$0}' colours.txt
```

The printf() function

For greater flexibility in how the output is formatted, you can use the awk **printf()** function. This is similar to **printf** in C, Lua, Bash, and other languages. It takes a *format* argument followed by a comma-separated list of items. The argument list may be enclosed in parentheses.

```
$ printf format, item1, item2, ...
```

The format argument (or *format string*) defines how each of the other arguments will be output. It uses *format specifiers* to do this, including **%s** to output a string and **%d** to output a decimal number. The following **printf** statement outputs the record followed by the number of fields in parentheses:

```
$ awk 'printf "%s (%d)\n", $0, NF}' colours.txt
name      color  amount (3)
raspberry red    4 (3)
banana    yellow 6 (3)
[...]
```

In this example, **%s (%d)** provides the structure for each line, while **\$0,NF** defines the data to be inserted into the **%s** and **%d** positions. Note that, unlike with the **print** function, no newline is generated without explicit

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.



Awk scripting

All of the awk code in this article has been written and executed in an interactive Bash prompt. For more complex programs, it's often easier to place your commands into a file or *script*. The option **-f FILE** (not to be confused with **-F**, which denotes the field separator) may be used to invoke a file containing a program.

For example, here is a simple awk script. Create a file called **example1.awk** with this content:

```
/^a/ {print "A: " $0}  
/^b/ {print "B: " $0}
```

It's conventional to give such files the extension **.awk** to make it clear that they hold an awk program. This naming is not mandatory, but it gives file managers and editors (and you) a useful clue about what the file is.

Run the script:

```
$ awk -f example1.awk colours.txt  
A: raspberry red 4  
B: banana yellow 6  
A: apple green 8
```

A file containing awk instructions can be made into a script by adding a **#!** line at the top and making it executable. Create a file called **example2.awk** with these contents:

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.



```
NR > 1 {  
    printf "%d: %s\n", NR, $0  
}
```

Arguably, there's no advantage to having just one line in a script, but sometimes it's easier to execute a script than to remember and type even a single line. A script file also provides a good opportunity to document what a command does. Lines starting with the `#` symbol are comments, which `awk` ignores.

Grant the file executable permission:

```
$ chmod u+x example2.awk
```

Run the script:

```
$ ./example2.awk colours.txt  
2: apple      red      4  
2: banana     yellow  6  
4: raspberry  red      3  
5: grape      purple  10  
[...]
```

An advantage of placing your `awk` instructions in a script file is that it's easier to format and edit. While you can write `awk` on a single line in your terminal, it can get overwhelming when it spans several lines.

Try it

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.



functions than just **print** and **printf**, refer to [the gawk manual](https://www.gnu.org/software/gawk/manual/) (<https://www.gnu.org/software/gawk/manual/>) online.

Here's an idea to get you started:

```
#!/usr/bin/awk -f
#
# Print each record EXCEPT
# IF the first record contains "raspberry",
# THEN replace "red" with "pi"

$1 == "raspberry" {
    gsub(/red/, "pi")
}

{ print }
```

Try this script to see what it does, and then try to write your own.

The next article in this series will introduce more functions for even more complex (and useful!) scripts.

This article is adapted from an episode of [Hacker Public Radio](http://hackerpublicradio.org/eps.php?id=2129) (<http://hackerpublicradio.org/eps.php?id=2129>), a community technology podcast.



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.



[sheet-awk](#))

Get our awk cheat sheet.

[Jim Hall \(Correspondent\)](#) [\(/users/jim-hall\)](#)



[\(/article/19/10/intro-awk\)](#)

Getting started with awk, a powerful text-parsing tool

[\(/article/19/10/intro-awk\)](#)

Let's jump in and start using it.

[Seth Kenlon \(Red Hat\)](#) [\(/users/seth\)](#) | [Dave Morriss](#) [\(/users/dave-morriss\)](#) | [Robert Young](#) [\(/users/ryoung29\)](#)

Topics : **[Programming](#)** [\(/tags/programming\)](#)



[\(/users/seth\)](#)

About the author

Seth Kenlon - Seth Kenlon is an independent multimedia artist, free culture advocate, and UNIX geek. He has worked in the [film](#) <http://www.imdb.com/name/nm1244992>) and [computing](#) <http://people.redhat.com/skenlon>) industry, often at the same time. He is one of the maintainers of the Slackware-based multimedia production project, <http://slackermmedia.info> (<http://slackermmedia.info>).

• [More about me](#) [\(/users/seth\)](#)

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.



Recommended reading



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



[How this open source test framework evolves with .NET](#) ([/article/20/9/testing-net-fixie?utm_campaign=intrel](#))



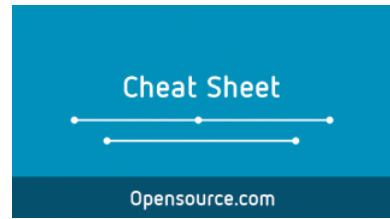
[Managing a non-profit organization's supply chain with Groovy](#) ([/article/20/9/groovy?utm_campaign=intrel](#))



[A practical guide to learning awk](#) ([/article/20/9/awk-ebook?utm_campaign=intrel](#))



[Optimize runtime performance with C++'s move semantics](#) ([/article/20/9/optimizing-runtime-performance](#))



[Learn the basics of programming with C](#) ([/article/20/8/c-programming-cheat-sheet?utm_campaign=intrel](#))

1 Comments



[Joseph cafe](#) ([/users/joseph567](#)) on 20 Nov 2019

0

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.



Its excelent for XML or JSON but NOT plain text! Awk can zero in on what I'm looking for, extract what I need, and give it to me back in no time. Nicely job!



[_ \(http://creativecommons.org/licenses/by-sa/4.0/\)](http://creativecommons.org/licenses/by-sa/4.0/)

Subscribe to our weekly newsletter

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](#)

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.

