Built-in Functions in AWK

Last Updated : 28 Apr, 2025

Built-in functions in AWK are **predefined operations** that simplify common tasks like calculations, text manipulation, and data processing. These functions are included directly in AWK, so you don't need to write code from scratch for basic operations. They save time and make scripts shorter and more efficient. AWK has two categories of high-level built-in functions:

- Built-in functions for numeric operations
- Built-in functions for string operations

1. Built-in Functions for Numeric Operations

This `awk` code shows the basic mathematical functions `int()`, `log()`, `sqrt()`, `sin()`, and `cos()`:

1. `int(n)` function

Returns integer part of given argument (lowest integer part).

```
awk 'BEGIN{print int(3.534);print int(4);print
int(-5.223);print int(-5);}'
```

Output:

```
3
4
-5
-5
```

2. `log(n)` function

Returns natural logarithm (base e) of positive numbers.

```
awk 'BEGIN{print log(3.534);print log(4);print log(0);print
log(-5);print log(-1);}
```

Output:

```
1.26243
1.38629
```

```
-inf
awk: cmd. line:1: warning: log: received negative argument
-5
nan
awk: cmd. line:1: warning: log: received negative argument
-1
nan
```

- Returns `-inf` for zero and `nan` for negative numbers.

3. `sqrt(n)` function

Returns positive root of given positive number.

```
awk 'BEGIN{print sqrt(16);print sqrt(0);print sqrt(-12);}'
```

Output:

```
4
0
awk: cmd. line:1: warning: sqrt: called with negative
argument -12
-nan
```

4. `sin(n)` function

Returns sine value of n (n in radians).

```
awk 'BEGIN{print sin(-60);print sin(90);print sin(45);}'
```

Output:

```
0.304811
```

```
0.893997
0.850904
```

5. `cos(n)` function

Returns cosine value of n (n in radians).

```
awk 'BEGIN{print cos(-60);print cos(90);print cos(45);}'
```

Output:

```
-0.952413
-0.448074
0.525322
```

2. Built-in Functions for String Operations

These AWK functions manipulate strings by finding positions, determining lengths, extracting substrings, changing case, and splitting them into arrays based on delimiters.

1. `index(str1, str2)` function

Returns position of first occurrence of `str2` in `str1` (indices start at 1).

```
awk 'BEGIN{print index("Graphic", "ph"); print
index("University", "abc")}'
```

Output:

```
4
0
```

- Returns `0` if `str2` is not found.

2. `length(string)` function

Returns length of string (includes spaces).

```
$ awk 'BEGIN{print length("Graphic Era University")}'
```

Output:

```
22
```

3. `substr(s, p, n)` function

Extracts substring from `s` starting at position `p` up to length `n`.

```
awk 'BEGIN{print substr("Graphic Era University", 9)}'
```

Output:

```
Era University
```

```
$ awk 'BEGIN{print substr("Graphic Era University", 9, 8)}'
```

Output:

```
Era Univ
```

4. `tolower(s)` function

Converts all uppercase characters to lowercase.

```
awk 'BEGIN{print tolower("GEEKSFORGEEKS")}'
```

Output:

```
geeksforgeeks
```

5. `toupper(s)` function

Converts all lowercase characters to uppercase.

```
awk 'BEGIN{print toupper("geeksforgeeks")}'
```

Output:

```
GEEKSFORGEEKS
```

6. `split(string, array, fieldsep)` function

Splits `string` into `array` using `fieldsep` as separator.

```
awk 'BEGIN{string="My Nationality Is Indian"; fieldsep=" ";
n=split(string, array, fieldsep); for(i=1; i<=n; i++)
{printf("%s\n", array[i]);}}'
```

Output:

```
My
Nationality
Is
Indian
```

- Splits sentence into words using space as separator.

Conclusion

Whether you're a system admin, data analyst, or just starting your journey in shell scripting, **AWK's built-in functions** can save you hours of manual work. These ready-to-use operations—covering both **numeric** and **string processing tasks**—eliminate the need to write complex logic from scratch.

With just a few lines of AWK code, you can do:

- **Calculate** roots, logs, and trigonometric values without external tools.
- **Process text**—find words, change cases, extract parts of strings—faster and more efficiently.
- **Split and parse data** from logs, CSVs, and input streams using powerful string functions.

Next Article

A aka...

Similar Reads

Bash Scripting - Functions

A Bash script is a plain text file. This file contains different commands for step-by-step execution. These commands can be written directly into the command line but from a reusability perceptive it is useful to st…

6 min read

PHP create_function() Function

The create_function() is an inbuilt function in PHP which is used to create an anonymous (lambda-style) function in the PHP. Syntax: string create_function ($args, $code) Parameters: This function accepts two…

2 min read

function command in Linux with examples

The function is a command in Linux that is used to create functions or methods. It is used to perform a specific task or a set of instructions. It allows users to create shortcuts for lengthy tasks making the…

2 min read

Lua Functions

Functions are fundamental building blocks in Lua programming. They help you organize code into reusable, manageable chunks, making your programs more readable, maintainable, and efficient.In this article, you'll…

5 min read

PHP | Functions

A function in PHP is a self-contained block of code that performs a specific task. It can accept inputs (parameters), execute a set of statements, and optionally return a value. PHP functions allow code reusabili…

8 min read

SQL | Advanced Functions

SQL (Structured Query Language) offers a wide range of advanced functions that allow you to perform complex calculations, transformations, and aggregations on your data. Aggregate Functions In database…

2 min read

Article Tags : Misc Linux-Unix linux-command

Practice Tags : Misc

Corporate & Communications Address:
A-143, 7th Floor, Sovereign Corporate

Registered Address:
K 061, Tower K, Gulshan Vivante Apartment, Sector 137, Noida, Gautam Buddh Nagar, Uttar Pradesh, 201305

Company

Languages

DSA

Basic DSA Problems
DSA Roadmap
Top 100 DSA Interview Problems
DSA Roadmap by Sandeep Jain
All Cheat Sheets

Data Science & ML

Data Science With Python
Data Science For Beginner
Machine Learning
ML Maths
Data Visualisation
Pandas
NumPy
NLP
Deep Learning

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Python Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering

Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar

Commerce

World GK

GeeksforGeeks Videos

DSA

Python

Java

C++

Web Development

Data Science

CS Subjects