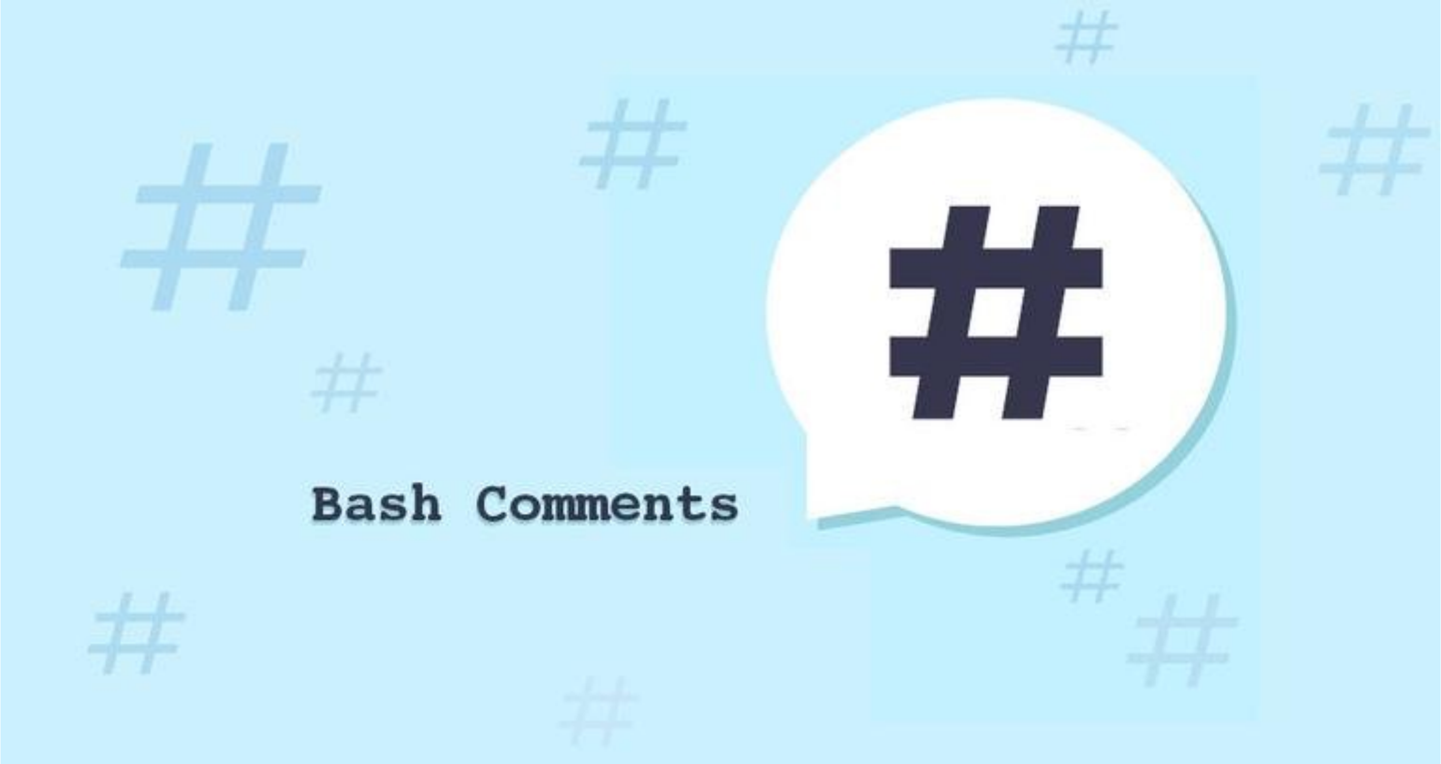
Writing Comments in Bash Scripts

Updated Feb 26, 2020 • 3 min read



When writing Bash scripts, it is always a good practice to make your code clean and easily understandable. Organizing your code in blocks, indenting, giving variables and functions descriptive names are several ways to do this.

Another way to improve the readability of your code is by using comments. A comment is a human-readable explanation or annotation that is written in the shell script.

Adding comments to your Bash scripts will save you a lot of time and effort when you look at your code in the future. Let’s say you want to change a script that you wrote a few months or years ago. The chances are that you will not remember why you wrote some complicated piece of code unless you added a comment.

The comments also help other developers and system administrators who may need to maintain the script to understand your code and its purpose.

or parameter substitution inside your Bash script, you should write a comment describing what the code does. Comments should be short and to the point. Do not explain something which is simple and obvious to the reader.

In this article, we'll cover the basics of writing comments in Bash.

Writing Comments in Bash

Bash ignores everything written on the line after the hash mark (`#`). The only exception to this rule is when the first line on the script starts with the `#!` characters. This sequence of characters is called Shebang and is used to tell the operating system which interpreter to use to parse the rest of the file.

Comments can be added at the beginning on the line or inline with other code:

```
# This is a Bash comment.
echo "This is Code" # This is an inline Bash comment.
```

The blank space after the hash mark is not mandatory, but it will improve the comment's readability.

If your text editor supports syntax highlighting, comments are usually represented in green.

Comments are also useful when testing a script. Instead of deleting some lines or blocks, you can comment them out:

```
# if [[ $VAR -gt 10 ]]; then
#  echo "Variable is greater than 10."
# fi
```

Multiline Comments in Bash

The simplest way to write multiline comments in Bash is to add single comments one after another:

```
# This is the first line.
# This is the second line.
```

Another option is to use HereDoc . It is a type of redirection that allows you to pass multiple lines of input to a command. If the HereDoc block is not redirected to a command, it can serve as a multiline comments placeholder:

```
<< 'MULTILINE-COMMENT'
    Everything inside the
    HereDoc body is
    a multiline comment
MULTILINE-COMMENT
```

Using HereDoc is a hack, not a real builtin way to write multiline Bash comments. To avoid any issues, you should prefer using single-line comments.

Conclusion

Writing comments is a good practice and helps other developers, including future self, to understand the shell script. In Bash, everything after the hash mark (#) and until the end of the line is considered to be a comment.

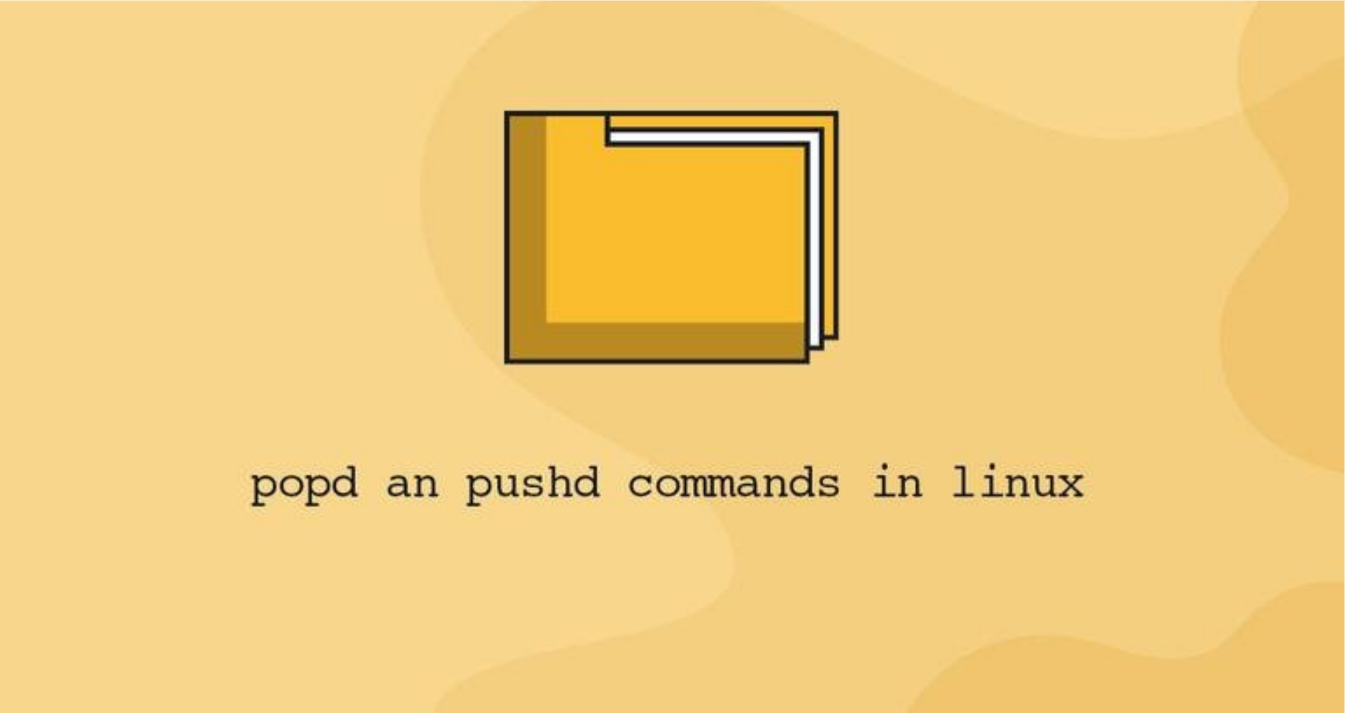
If you have any questions or feedback, feel free to leave a comment.

bash terminal

Related Articles

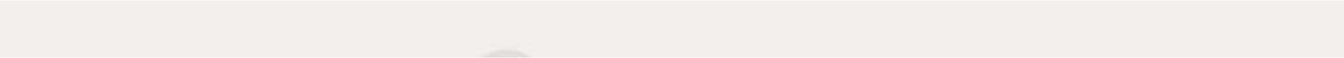
SEP 21, 2019

Pushd and Popd Commands in Linux



AUG 11, 2019

How to Increment and Decrement Variable in Bash (Counter)



Show comments (7)