

# Adding a New Line in PowerShell

---

When writing script output to the console, or a text file, you often want to format it for readability. A common way for this in PowerShell, is to use line breaks, which forces the next part to the output on a new line.

Line breaks are always a bit tricky to get right in any scripting or programming language, but by using ``n` and ``r` we can force the content easily on a new line.

In this article, I will show you how to use the new line characters in PowerShell.

## New Line in PowerShell

---

To continue the output in PowerShell on a new line, we can use the special characters ``n` and ``r`. The first is used for a line break, and the latter is used for the carriage return. Now this difference is important to understand, when it comes to using these characters.

The ``n` (Line Feed) character is the newline character (ASCII 10, LF), which moves the cursor down to the next line.

Write-Host "String that we want to show `non two lines"

# Result

String that we want to show  
on two lines

Now important to note in the example above, is that we don't add a space after the new line character. If you would add a space after the special character then the new line will start with a space, something that you don't want.

The ``r` is the carriage return character (ASCII 13, CR), which moves the cursor back to the beginning of the line. If you would use this without the new line character, then you would overwrite your existing output, as you can see in the example below:

Write-Host "String that we want to show `ron a new line"

# Result

on a new line we want to show

PowerShell will first write the output "String what we want to show" and then the cursor is placed back to the beginning of the line, and the rest of the output "on an new line) will overwrite the first characters of the existing string.

## Combining ``r` and ``n`

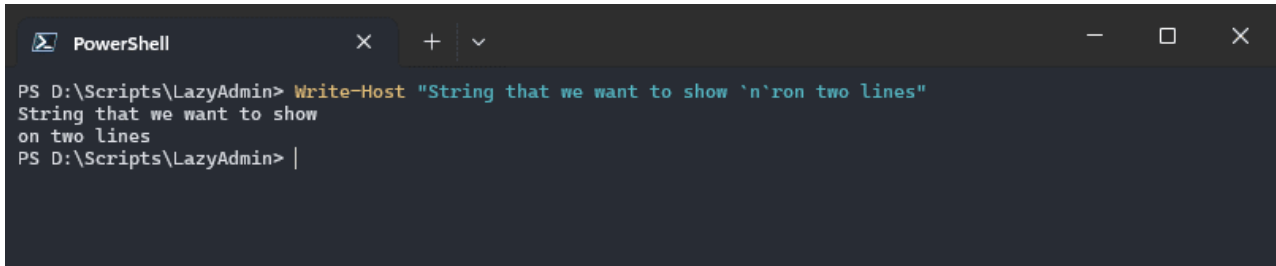
---

In Windows we commonly use both special characters to create a new line. This isn't really necessary anymore these days, but this way you can be sure that it works with all applications on Windows.

If you are using the PowerShell scripts only on Unix-based systems, then you can do with only using the new line character.

So continue the output on a new line in Windows, it's recommend to use the following:

Write-Host "String that we want to show `n`ron two lines"



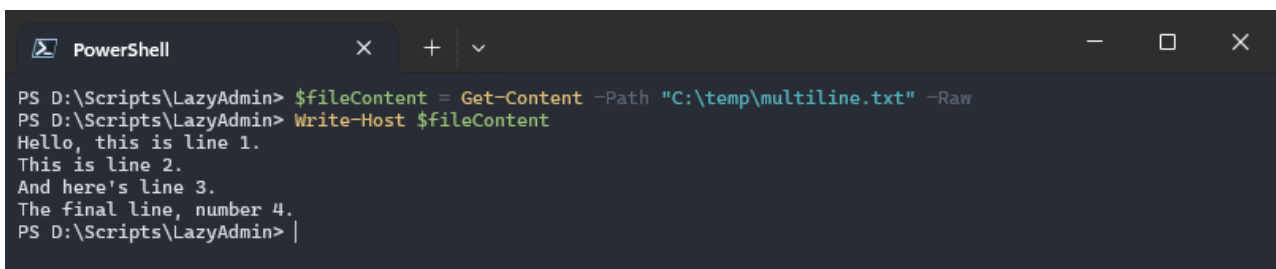
```
PS D:\Scripts\LazyAdmin> Write-Host "String that we want to show `n`ron two lines"
String that we want to show
on two lines
PS D:\Scripts\LazyAdmin> |
```

## Reading a File while keeping Line Breaks

When you read a text file with PowerShell, all lines will be formatted into one long string by default. We can however keep the new line information by using the `-raw` parameter with the `Get-Content` cmdlet.

The `-raw` parameter will read the entire content of the file, while preserving the original line breaks. This way, when you output the script, the line breaks will shown:

```
$fileContent = Get-Content -Path "C:\temp\multiline.txt" -Raw
Write-Host $fileContent
```



```
PS D:\Scripts\LazyAdmin> $fileContent = Get-Content -Path "C:\temp\multiline.txt" -Raw
PS D:\Scripts\LazyAdmin> Write-Host $fileContent
Hello, this is line 1.
This is line 2.
And here's line 3.
The final line, number 4.
PS D:\Scripts\LazyAdmin> |
```

## Writing Commands on a New Line

Another common question with PowerShell is how to write a part of a command on a new line. Long commands on a single line are harder to read, so sometimes you want to continue your command on a new line.

The best way to do this is by using `Splatting`, this is a method where you add parameters into a hashtable. But this isn't always possible or practical. In these cases, we can use the ``` character to break the command into new line.

The example below is maybe a bit extreme, but it gives you a good idea of how breaking up your command into multiple lines improves the readability in PowerShell:

```
Get-ChildItem -Path C:\Users `
-Recurse `
```

```
-Filter *.txt `
-File `
| Where-Object { $_.CreationTime -gt (Get-Date).AddDays(-7) } `
| Select-Object Name, CreationTime, Length `
| Sort-Object CreationTime -Descending `
| Format-Table -AutoSize
```

## Wrapping Up

---

Using new lines in PowerShell not only improves the readability of your output, but also for long commands. When it comes to line breaks in the output, I recommend to use both special characters for maximum compatibility, but keep in mind that using only ``n` is also possible.

Hope you liked this article, if you have any questions, just drop a comment below.

Did you **Liked** this **Article**?

Get the latest articles like this **in your mailbox**  
or share this article

Не удается связаться с сервисом reCAPTCHA. Проверьте подключение к Интернету и перезагрузите страницу.

I hate spam to, so you can unsubscribe at any time.