How to Split a String in PowerShell

lazyadmin.nl/powershell/split-string

October 3, 2023

In PowerShell, a pretty common task is to split a string into one or more substrings. Splitting a string is always done based on a delimiter. This can be pretty much anything in the string, like a space, or comma, or a specific character.

To split a string in PowerShell we will use the `-split` operator or the Split method. This allows you to split a string into an array of substrings based on a specified delimiter, how many substrings you want to return, and some other options.

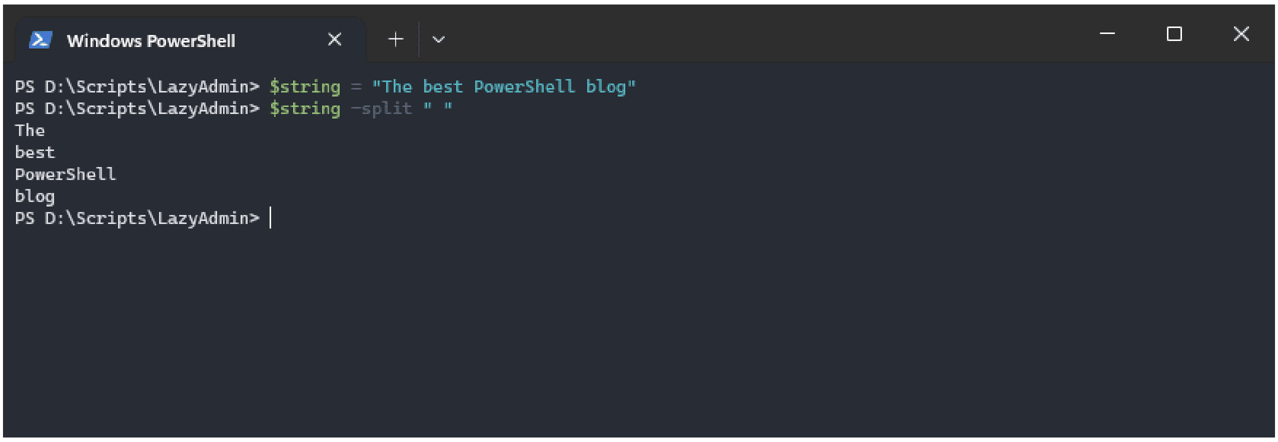
In this article, we will look at how to simply split a string in Powershell and at some of the advanced options.

PowerShell Split String

The `-split` operator or `Split()` method is a built-in way to split a string into an array of substrings. You provide a delimiter (or a regular expression pattern) to determine where the string should be split.

For example, to split a string into each word, we can use the space as a delimiter. This will return each individual word of the string:

```
$string = "The best PowerShell blog"
$string -split " "
# Result
The
best
PowerShell
blog
```



```
PS D:\Scripts\LazyAdmin> $string = "The best PowerShell blog"
PS D:\Scripts\LazyAdmin> $string -split " "
The
best
PowerShell
blog
PS D:\Scripts\LazyAdmin>
```

In the example above we outputted the results directly to the console, but normally you want to use the results in your script. When you store the results in a variable, you will get an array with each substring.

Taking the example, below, if we only want to get the date from the string, we can split the string, using the space as a delimiter, and only return the last item from the array:

```
$string = "Delivery-contract-20230401"
$splitArray = $string.Split("-")
# Select the last item from the array
$splitArray[-1]
# Result
20230401
```

Good to know is that if you don’t specify a delimiter, then it will split on space as well. The split operator is case-insensitive by default, if you need to be case-sensitive, then use the `-csplit` operator.

In some case you might want to split a string on a comma, or other delimiter, but each substrings starts with a space. To remove the spaces from each substring you can best use the Trim() method:

```
$string = "apple,banana,cherry,date"
$string.split(',')
# Result
apple
banana
cherry
date
# Remove space by including trim:
$string.split(',').trim()
```

Using Multiple Delimiters

In some cases, you might need to split a string based on different delimiters. We can do this by using a regular expression. This allows us to specify multiple delimiters. In the example below, we are going to split the string on `,` and `|`

```
$string = "apple,banana,cherry|date"
$string -split "[,|]"
# Result
apple
banana
cherry
date
```

Regular expressions can be used to define custom patterns on which you want to split a string. I won’t go into detail on how regular expressions work, but I will show you a few examples to get you started:

```
# Split on multiple spaces or tabs:
$string = "This is a string with multiple spaces and tabs"
$string -split "\s+"
```

```
# Split on decimals:
$string = "Hello123World456Test789"
$string -split "\d+"
# Split on sentences:
$string = "This is a sentence. And here is another one! Finally, a third sentence."
$splitArray = $string -split "(?<=[.!?])\s+"
```

Limit the Results

By default, the split method will return all the substrings, but what if you only need the first part? For this, we can use the `Maxsubstrings` parameter, which will limit the number of substrings that are returned.

You can use a positive number to start counting from the beginning of the string and a negative number to start counting from the end of the string. For example, we only want to separate the date from the file name:

```
$string = "20230401-La-srv-dc01 Event Log.log"
$string -split "-", 2
# Result
20230401
La-srv-dc01 Event Log.log
```

The advantage of limiting the results, is that it can speed up your script. With a single string you won't notice it. But if you are processing thousands of records, then getting only what you need can save a couple of seconds of your script run time.

Wrapping Up

Splitting string in PowerShell is pretty straight forward with the spit method or operator. If you only want to have to first n characters of a string, then you should use the substring method in PowerShell.

I hope you found this article useful, 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

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