

How-to use Substring in PowerShell

 lazyadmin.nl/powershell/substring

March 9, 2022

PowerShell SubString method

How to extract a part of string



When you want to extract a part of a string in PowerShell we can use the Substring() method. This method allows us to specify the start and length of the substring that we want to extract from a string.

In this article, I will explain how the Substring method works, and show you a couple of examples of how to get parts of a string using the startIndex and/or length function.

Using the Substring Method in PowerShell

The Substring method can be used on any string object in PowerShell. This can be a literal string or any variable of the type string.

To use the method we will need to specify the starting point of the string that we want to extract. Optionally we can specify the length (number of characters), that we want to extract.

Substring(startIndex, length)

Tip

Keep in mind that a string starts at position 0 and not 1

This method is best explained with the help of examples, so let's start with the following famous string "hello world". To extract only the word *world* from the string we can use the following command:

```
"Hello World".Substring(6)
```

```
# Result
```

```
World
```

We have only specified the start point of the substring, 6. This way the method will extract all characters starting from position 6 of the string.

To explain the position of each character a bit better, I have numbered the position below each letter of the string. As you can see, position 6 is the letter **W**.

```
Hello World
```

```
012345678910
```

If we only want to extract the word *Hello* from the string, then we need to specify the `startIndex` of 0 and a length of 5 characters:

```
"Hello World".Substring(0,5)
```

```
# Result
```

```
Hello
```

Working with Dynamic Strings

When working with dynamic strings you don't always know the length of the string. For this, we can use the string length function inside the substring method.

Let's say we have a folder with log files from which we want to extract the filenames without the extension. The filename itself can have different lengths, but the extension, in this case, is 3 letters and a dot.

So to keep only the filenames we can get the length of the filename and subtract 4 characters from it:

```
$filename = "la-ams-file02-log-1.log"
```

```
$filename.Substring(0, $filename.Length - 4)
```

```
# Result
```

```
la-ams-file02-log-1
```

We can also use this the other way around to get only the extensions from the filenames. We determine the start position based on the length of the filename, and specify a length of 3 characters:

```
$filename = "la-ams-file02-log-1.log"
```

```
$filename.Substring($filename.Length - 3, 3)
```

```
# Result
```

```
log
```

In this case, we actually don't need to specify the length of the substring to return (3), because we can simply return the remaining characters:

```
$filename.Substring($filename.Length - 3)
```

```
# Result
```

log

Using IndexOf in Substring

When working with dynamics string lengths you sometimes want to extract a part in the middle of the string. The challenge then is to find/calculate the exact starting position for the substring method.

One option to do this is to use IndexOf. This method allows you to find the position of a character or string inside a string.

Let's go back to our log files example. I have added a totally useless date format to it, with inconsistent length. To get only the date from the name of the log files, we can use the `IndexOf()` method.

We are going to search in the string for the unique part before the date `-log`. The IndexOf method will return the starting position of the string or character that you searched for, so in this case, we will need to add 4 characters to it:

```
$files = @(
'la-ams-ad01-log-202223.log',
'la-ams-ad01-log-2022113.log',
'la-osl-ad01-log-2022113.log',
'la-osl-file01-log-202214.log'
)
$files | Foreach {
# Get the position of the substring 'log-' in each filename and add 4 to this position.
$part = $_.Substring($_.IndexOf('log-') + 4)
# Returns 202223.log for example
# Remove the extension from the substring
$date = $part.Substring(0, $part.Length - 4)
write-host $date
# Returns 202223 for example
}
```

In this particular case, you could also use the method `LastIndexOf()` in the hyphens in the names of the files:

```
$files | Foreach {
$part = $_.Substring($_.LastIndexOf('-') + 1)
write-host $part
$date = $part.Substring(0, $part.Length - 4)
write-host $date
}
```

Wrapping Up

When working with strings in PowerShell you will probably be using the substring method quite a lot. Keep in mind that you can always combine the substring method with other string operators like split or replace.

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.