Trim characters from strings in PowerShell

https://4sysops.com/archives/trim-characters-from-strings-in-powershell/

Most programming languages provide string functions that can be used to selectively truncate leading or trailing characters. String objects in PowerShell have three such trim() methods that remove not only spaces but also any characters at the beginning and end.

Contents

1. Remove a list of characters from a string
2. Trim substrings
3. Summary

Those who have experience with Visual Basic know that LTrim, Trim, and RTrim provide a convenient way to strip strings of unnecessary whitespace at the beginning and end. For all other characters, you have to resort to a method for replacing substrings.

In PowerShell, you can replace substrings using either a method or an operator, but you don't need them to truncate leading or trailing characters. This can be done using the trim methods.

If you want to delete the characters only at the beginning, use *TrimStart()*, and for trimming the string at the end, use *TrimEnd()*. *Trim()* removes them from both sides.

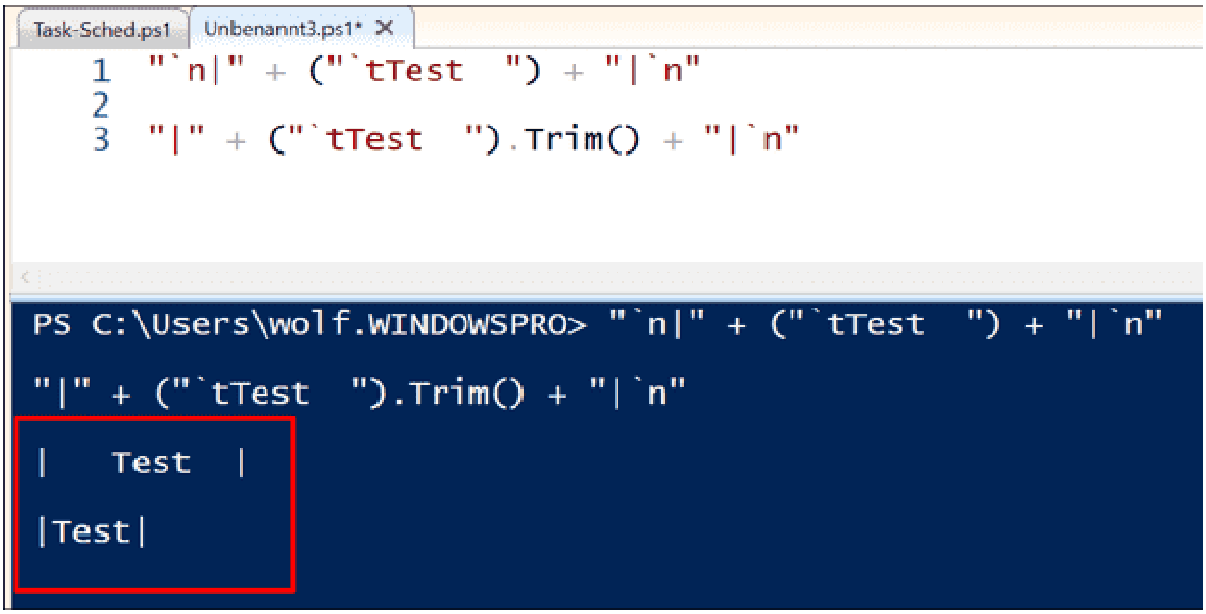
If you invoke them without parameters, then they will only trim whitespace at the beginning, at the end, or on both sides, as in Basic. The string

```
"`tTest "
```

starts with a tab (`t) and ends with two spaces. The command

```
("`tTest ").Trim()
```

removes the whitespace at both ends.



Remove a list of characters from a string

The option to truncate not only whitespace but arbitrary characters is even more interesting. You pass these as an array to one of the Trim() methods. They then remove all characters until they hit one that is not on the list:

```
("<-- This is an HTML comment -->").Trim(" ", "<", "-", ">")
```

In this example, *Trim()* clips all characters from the beginning and the end until it encounters one that is not a space, a lesser-than sign, a greater-than sign, or a hyphen. The result is the plain comment without the comment markers.

Extract the text of an HTML comment with Trim

When you use these methods with an array as an argument, keep in mind that they remove character by character and not a substring like "<--". The above example would therefore not only extract comments when run over an HTML file but also distort HTML elements if they are at the beginning or end of a line. Hence, the result for

```
("<p>This is a paragraph</p>").Trim(" ", "<", "-", ">")
```

would be

```
p> This is a paragraph</p
```

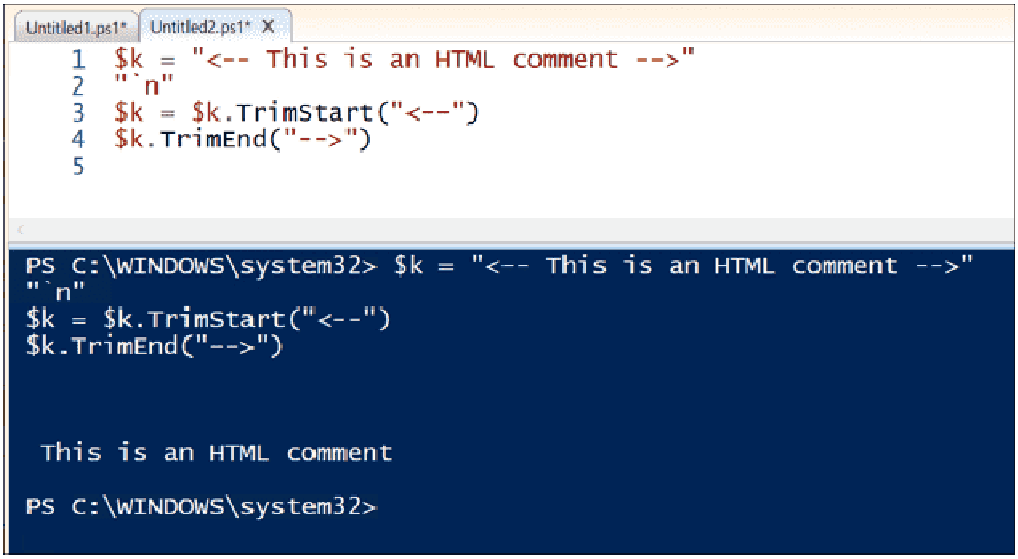
Trim substrings

In such situations, however, the *Trim()* methods enable passing whole strings as arguments:

```
("<-- This is an HTML comment -->").TrimStart("<--")
```

The above command removes the opening marker of an HTML comment. However, you are limited to a single string, and it is not possible to specify an array of strings. Therefore, you would have to pass the result to a *Trim()* method to remove more characters:

```
$k = "<-- This is an HTML comment -->"
$k = $k.TrimStart("<--")
$k.TrimEnd("-->")
```



Summary

The trim methods in PowerShell are more powerful than in, for example, Basic or JavaScript. If you invoke them without arguments, they remove only whitespace at the ends of a string.

When given an array of characters, *Trim(), TrimStart(), and TrimEnd()* will remove all characters from this list and only stop when they encounter one that isn't part of it.