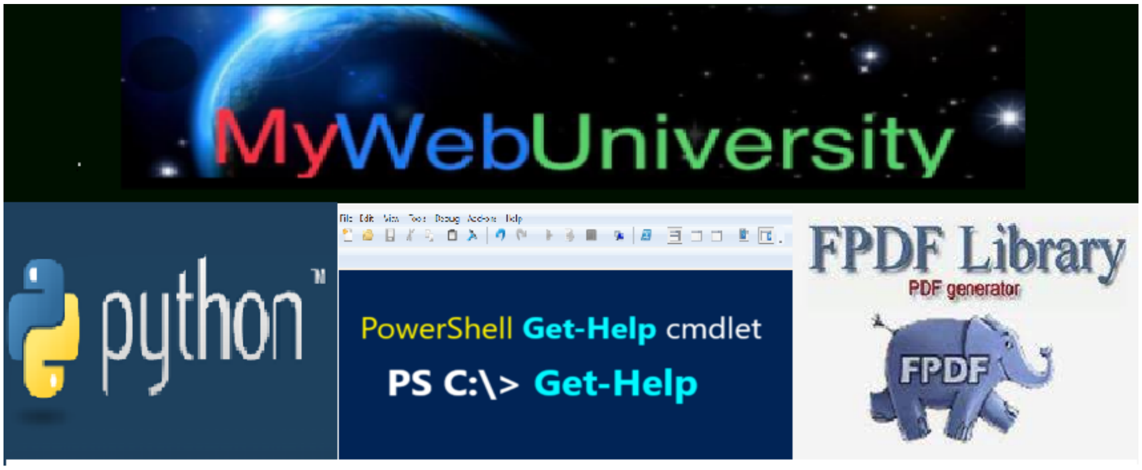

Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Write-Color'

PS:\>Get-HELP Write-Color -Full

NAME

Write-Color

SYNOPSIS

Write-Color is a wrapper around Write-Host delivering a lot of additional features for easier color options.

SYNTAX

Write-Color [[-Text] <String[]>] [[-Color] {Black | DarkBlue | DarkGreen | DarkCyan | DarkRed | DarkMagenta | DarkYellow | Gray | DarkGray | Blue | Green | Cyan | Red

| Magenta | Yellow | White}] [[-BackGroundColor] {Black | DarkBlue | DarkGreen | DarkCyan | DarkRed | DarkMagenta | DarkYellow | Gray | DarkGray | Blue | Green | Cyan

| Red | Magenta | Yellow | White}] [[-StartTab] <Int32>] [[-LinesBefore] <Int32>] [[-LinesAfter] <Int32>] [[-StartSpaces] <Int32>] [[-LogFile] <String>]

[[-DateTimeFormat] <String>] [[-LogTime] <Boolean>] [[-LogRetry] <Int32>] [[-Encoding] <String>] [-ShowTime] [-NoNewLine] [-NoConsoleOutput] [<CommonParameters>]

DESCRIPTION

Write-Color is a wrapper around Write-Host delivering a lot of additional features for easier color options.

It provides:

- Easy manipulation of colors,

- Logging output to file (log)

- Nice formatting options out of the box.

- Ability to use aliases for parameters

PARAMETERS

-Text <String[]>

Text to display on screen and write to log file if specified.

Accepts an array of strings.

```
Required?                    false
Position?                    1
Default value
Accept pipeline input?       false
Accept wildcard characters?  false
```

-Color

Color of the text. Accepts an array of colors. If more than one color is specified it will loop through colors for each string.

If there are more strings than colors it will start from the beginning.

Available colors are: Black, DarkBlue, DarkGreen, DarkCyan, DarkRed, DarkMagenta, DarkYellow, Gray, DarkGray, Blue, Green, Cyan, Red, Magenta, Yellow, White

```
Required?                    false
Position?                    2
Default value                White
Accept pipeline input?       false
Accept wildcard characters?  false
```

-BackGroundColor

Color of the background. Accepts an array of colors. If more than one color is specified it will loop through colors for each string.

If there are more strings than colors it will start from the beginning.

Available colors are: Black, DarkBlue, DarkGreen, DarkCyan, DarkRed, DarkMagenta, DarkYellow, Gray, DarkGray, Blue, Green, Cyan, Red, Magenta, Yellow, White

Required?   false

Position?   3

Default value

Accept pipeline input?   false

Accept wildcard characters? false

-StartTab <Int32>

Number of tabs to add before text. Default is 0.

Required?   false

Position?   4

Default value   0

Accept pipeline input?   false

Accept wildcard characters? false

-LinesBefore <Int32>

Number of empty lines before text. Default is 0.

Required?   false

Position?   5

Default value   0

Accept pipeline input?   false

Accept wildcard characters? false

-LinesAfter <Int32>

Number of empty lines after text. Default is 0.

Required?   false

Position?   6

Default value   0

Accept pipeline input?   false

Accept wildcard characters? false

-StartSpaces <Int32>

Number of spaces to add before text. Default is 0.

Required?   false

Position?   7

Default value   0

Accept pipeline input?   false

Accept wildcard characters? false

-LogFile <String>

Path to log file. If not specified no log file will be created.

Required?   false

Position?   8

Default value

Accept pipeline input?   false

Accept wildcard characters? false

-DateTimeFormat <String>

Custom date and time format string. Default is yyyy-MM-dd HH:mm:ss

Required?   false

Position?   9

Default value   yyyy-MM-dd HH:mm:ss

Accept pipeline input?   false

Accept wildcard characters? false

-LogTime <Boolean>

If set to $true it will add time to log file. Default is $true.

Required?	false
Position?	10
Default value	True
Accept pipeline input?	false
Accept wildcard characters?	false

-LogRetry <Int32>

Number of retries to write to log file, in case it can't write to it for some reason, before skipping. Default is 2.

Required?	false
Position?	11
Default value	2
Accept pipeline input?	false
Accept wildcard characters?	false

-Encoding <String>

Encoding of the log file. Default is Unicode.

Required?	false
Position?	12
Default value	Unicode
Accept pipeline input?	false
Accept wildcard characters?	false

-ShowTime [<SwitchParameter>]

Switch to add time to console output. Default is not set.

Required?	false
Position?	named
Default value	False

```
        Accept pipeline input?       false
        Accept wildcard characters?  false


    -NoNewLine [<SwitchParameter>]
        Switch to not add new line at the end of the output. Default is not set.


        Required?                    false
        Position?                    named
        Default value                False
        Accept pipeline input?       false
        Accept wildcard characters?  false


    -NoConsoleOutput [<SwitchParameter>]
        Switch to not output to console. Default all output goes to console.


        Required?                    false
        Position?                    named
        Default value                False
        Accept pipeline input?       false
        Accept wildcard characters?  false


    <CommonParameters>
        This cmdlet supports the common parameters: Verbose, Debug,
        ErrorAction, ErrorVariable, WarningAction, WarningVariable,
        OutBuffer, PipelineVariable, and OutVariable. For more information, see
        about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

INPUTS


OUTPUTS


NOTES
```

Understanding Custom date and time format strings: https://learn.microsoft.com/en-us/dotnet/standard/base-types/custom-date-and-time-format-strings

Project support: https://github.com/EvotecIT/PSWriteColor

Original idea: Josh (https://stackoverflow.com/users/81769/josh)

-------------------------- EXAMPLE 1 --------------------------

```
PS C:\>Write-Color -Text "Red ", "Green ", "Yellow " -Color Red,Green,Yellow
```

-------------------------- EXAMPLE 2 --------------------------

```
PS C:\>Write-Color -Text "This is text in Green ",

"followed by red ",
                "and then we have Magenta... ",
                "isn't it fun? ",
                "Here goes DarkCyan" -Color Green,Red,Magenta,White,DarkCyan
```

-------------------------- EXAMPLE 3 --------------------------

```
PS C:\>Write-Color -Text "This is text in Green ",

"followed by red ",
                "and then we have Magenta... ",
            "isn't it fun? ",

            "Here goes DarkCyan" -Color Green,Red,Magenta,White,DarkCyan -StartTab 3 -LinesBefore 1 -LinesAfter 1
```

-------------------------- EXAMPLE 4 --------------------------

```
PS C:\>Write-Color "1. ", "Option 1" -Color Yellow, Green


Write-Color "2. ", "Option 2" -Color Yellow, Green

Write-Color "3. ", "Option 3" -Color Yellow, Green

Write-Color "4. ", "Option 4" -Color Yellow, Green

Write-Color "9. ", "Press 9 to exit" -Color Yellow, Gray -LinesBefore 1
```

-------------------------- EXAMPLE 5 --------------------------

```
PS C:\>Write-Color -LinesBefore 2 -Text "This little ","message is ", "written to log ", "file as well." `


-Color Yellow, White, Green, Red, Red -LogFile "C:\testing.txt" -TimeFormat "yyyy-MM-dd HH:mm:ss"

Write-Color -Text "This can get ","handy if ", "want to display things, and log actions to file ", "at the same time." `

        -Color Yellow, White, Green, Red, Red -LogFile "C:\testing.txt"
```

-------------------------- EXAMPLE 6 --------------------------

```
PS C:\>Write-Color -T "My text", " is ", "all colorful" -C Yellow, Red, Green -B Green, Green, Yellow
```

```
Write-Color -t "my text" -c yellow -b green

Write-Color -text "my text" -c red
```

-------------------------- EXAMPLE 7 --------------------------

```
PS C:\>Write-Color -Text "Testuje czy sie ladnie zapisze, czy beda problemy" -Encoding unicode -LogFile 'C:\temp\testinggg.txt' -Color Red -NoConsoleOutput
```

RELATED LINKS