PowerShell Get-Help on command 'Clear-Variable'

PS C:\Users\wahid> Get-Help Clear-Variable

NAME

    Clear-Variable

SYNOPSIS

    Deletes the value of a variable.

SYNTAX

    Clear-Variable [-Name] <System.String[]> [-Exclude <System.String[]>] [-Force]

    [-Include <System.String[]>] [-PassThru] [-Scope <System.String>] [-Confirm]

    [-WhatIf] [<CommonParameters>]

DESCRIPTION

    The `Clear-Variable` cmdlet deletes the data stored in a variable, but it does

    not delete the variable. As a result, the value of the variable is NULL

    (empty). If the variable has a specified data or object type, this cmdlet

    preserves the type of the object stored in the variable.

PARAMETERS

-Exclude <System.String[]>

    Specifies an array of items that this cmdlet omits in the operation. The

    value of this parameter qualifies the Name parameter. Enter a name element

    or pattern, such as "s*". Wildcards are permitted.

-Force <System.Management.Automation.SwitchParameter>

    Allows the cmdlet to clear a variable even if it is read-only. Even using

    the Force parameter, the cmdlet cannot clear constants.

-Include <System.String[]>

    Specifies an array of items that this cmdlet includes in the operation.

    The value of this parameter qualifies the Name parameter. Enter a name

    element or pattern, such as "s*". Wildcards are permitted.

-Name <System.String[]>

    Specifies the name of the variable to be cleared. Wildcards are permitted.

    This parameter is required, but the parameter name Name is optional.

-PassThru <System.Management.Automation.SwitchParameter>

    Returns an object representing the item with which you are working. By

    default, this cmdlet does not generate any output.

-Scope <System.String>

    Specifies the scope in which this alias is valid.

    The acceptable values for this parameter are:

    - `Global`

    - `Local`

    - `Script`

You can also use a number relative to the current scope (0 through the number of scopes, where 0 is the current scope and 1 is its parent). Local is the default. For more information, see about_Scopes (../Microsoft.PowerShell.Core/About/about_Scopes.md).

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

<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).

Example 1: Remove the value of global variables that begin with a search string

```
Clear-Variable my* -Scope Global
```

This command removes the value of global variables that have names that begin with my.

Example 2: Clear a variable in a child scope but not the parent scope

```
$a=3
&{ Clear-Variable a }
$a
```

```
3
```

These commands demonstrate that clearing a variable in a child scope does not clear the value in the parent scope. The first command sets the value of the variable `$a` to 3. The second command uses the invoke operator (`&`) to run the `Clear-Variable` command in a new scope. The variable is cleared in the child scope (although it did not exist), but it is not cleared in the local scope. The third command, which gets the value of `$a`, shows that the value 3 is unaffected.

---- Example 3: Delete the value of the specified variable ----

Clear-Variable -Name "Processes"

This command deletes the value of the variable named Processes. After the cmdlet completes the operation, the variable named Processes still exists, but the value is null.

REMARKS

To see the examples, type: "get-help Clear-Variable -examples".

For more information, type: "get-help Clear-Variable -detailed".

For technical information, type: "get-help Clear-Variable -full".

For online help, type: "get-help Clear-Variable -online"