



python



PowerShell

FPDF Library  
PDF generator

*Full credit is given to the above companies including the OS that this PDF file was generated!*

### **PowerShell Get-Help on command 'Read-Host'**

**PS C:\Users\wahid> Get-Help Read-Host**

#### NAME

Read-Host

#### SYNOPSIS

Reads a line of input from the console.

#### SYNTAX

Read-Host [[-Prompt] <System.Object>] [-AsSecureString] [<CommonParameters>]

#### DESCRIPTION

The `Read-Host` cmdlet reads a line of input from the console (stdin). You can use it to prompt a user for input. Because you can save the input as a secure string, you can use this cmdlet to prompt users for secure data, such as passwords.

> [!NOTE] > `Read-Host` has a limit of 8190 characters it can accept as input from a user.

## PARAMETERS

`-AsSecureString <System.Management.Automation.SwitchParameter>`

Indicates that the cmdlet displays asterisks (`\*`) in place of the characters that the user types as input. When you use this parameter, the output of the `Read-Host` cmdlet is a SecureString object ( `System.Security.SecureString` ).

`-Prompt <System.Object>`

Specifies the text of the prompt. Type a string. If the string includes spaces, enclose it in quotation marks. PowerShell appends a colon ( `:` ) to the text that you enter.

`<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: Save console input to a variable -----

```
$Age = Read-Host "Please enter your age"
```

----- Example 2: Save console input as a secure string -----

```
$pwd_secure_string = Read-Host "Enter a Password" -AsSecureString
```

----- Example 3: Normalizing input -----

```
$prompt = @(
```

```
'List the cities you want weather information for.'
```

```
'When specifying multiple cities, separate them with a semi-colon, like:'
```

```

    "New York; Osan; Koforidua"
) -join ' '

$cities = Read-Host $prompt

"Input cities string: `n`t$cities"

$splitCities = $cities -split ';'

"Split cities array:"
$splitCities | ForEach-Object -Process { "`t$_" }

$normalizedCities = $splitCities | ForEach-Object -Process { $_.Trim() }

"Normalized split cities array:"
$normalizedCities | ForEach-Object -Process { "`t$_" }

Input cities string:
    ' New York; Osan ;Koforidua '

Split cities array:
    ' New York'
    ' Osan '
    'Koforidua '

Normalized split cities array:
    'New York'
    'Osan'
    'Koforidua'

```

The example uses the ``-split`` operator to convert the input string into an array of strings. Each string in the array includes the name of a different city. However, the split strings include extra spaces. The ``Trim()`` method removes the leading and trailing spaces from each string.

To see the examples, type: "get-help Read-Host -examples".

For more information, type: "get-help Read-Host -detailed".

For technical information, type: "get-help Read-Host -full".

For online help, type: "get-help Read-Host -online"