

# How to Pause in PowerShell

---

 lazyadmin.nl/powershell/pause-powershell

October 26, 2023

PowerShell is a great way to automate tasks or processes, like creating users. But sometimes during such a task or process, you need to pause the script. For example to wait until a user account is synced or a file is downloaded. In those cases, we need the Pause PowerShell.

There are a couple of methods to pause a PowerShell script. Some use the built-in features in PowerShell, while others may be system-specific.

In this article, we will look at the different options to pause PowerShell or to let it wait for a command to continue.

## Pause PowerShell with Start-Sleep

---

The most common way to pause a PowerShell script for a given period of time is to use the **Start-Sleep** command. It allows you to define how long to pause your script in **seconds** or **milliseconds**. So to pause your script for 5 seconds, you can simply do:

```
# Pause 5 seconds
```

```
Start-Sleep -Seconds 5
```

```
# Or shorter options are:
```

```
Start-Sleep -s 5
```

```
Start-Sleep 5 # Default is in seconds
```

To use milliseconds, simply use the **-milliseconds** parameter. Other options, like hours or minutes, are not available. So if you want to pause your script for 5 minutes, simply specify 300 seconds instead.

If you are using PowerShell 7 then you can also use the **-Duration** parameter, which accepts a TimeSpan object. The advantage of this is that also can specify the time to wait in minutes or hours even:

```
Start-Sleep -Duration (New-TimeSpan -Minutes 5)
```

If you want to know more about the Start-Sleep command, and for example how to use it with a progress bar, then make sure you read [this article](#) where I further explain the options.

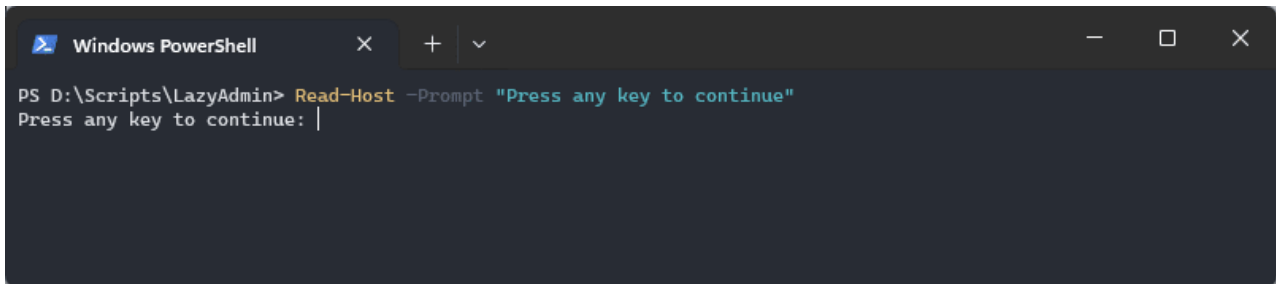
## Wait for Command with Read-Host

---

Another common option is to wait for a command in PowerShell before the script continues. To pause your scripts and wait for a command, we can use the **Read-Host** cmdlet. This will pause your script and wait for any key to press to continue.

For example, to simply wait for any key to be pressed before continuing your script, we can use the following code:

```
Read-Host -Prompt "Press any key to continue"
```



We can also be more specific with Read-Host. We could for example ask if the user wants to continue or not:

```
$continue = Read-Host -Prompt "Continue? [y/n]"  
if ( $continue -eq 'n' ) {  
Exit  
}
```

If you really want to limit the options to only Y or N, then we could use `PromptForChoice()` method for this. This allows us to specify the options that the user can choose:

```
$title = 'Question'  
$question = 'Are you sure you want to continue?'  
$choices = '&Yes', '&No'  
$decision = $Host.UI.PromptForChoice($title, $question, $choices, 1)  
if ($decision -eq 0) {  
Write-Host 'confirmed'  
} else {  
Write-Host 'cancelled'  
Exit  
}
```

## Using the Pause Command

---

Another easy way to your Pause PowerShell script and wait for a command to continue is to use the Windows method Pause. This is not a PowerShell command, but a command prompt method that we can also use in PowerShell:

```
cmd /c 'pause'
```



```
Windows PowerShell
PS D:\Scripts\LazyAdmin> cmd /c 'pause'
Press any key to continue . . . |
```

There is really no advantage of this method compared to using the Read-Host function. Personally, I would use the native PowerShell function instead of this. The problem with the command function is that it only works on Windows.

## Wrapping Up

---

When you need to pause your PowerShell script, try to use the PowerShell methods first. This will make sure that your script will also run on other systems.

I hope you found this article helpful if you have any questions, just let me know!

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.