By OnlineInterviewQuestions.com

PowerShell Interview Questions

Q1. What is PowerShell?

PowerShell is a command-line tool built upon the .NET framework to help the windows system administrator. With PowerShell, you can control your system operations such as accessing data storage, registry, and file system. It also has a good expression parser and a scripting language to control and automate the administration of the windows system. This open-sourced tool was developed by Microsoft and released in 2006.

The operations in the command line are executed by cmdlets. Cmdlets are .NET classes to implement operations in the PowerShell.

Q2. How to run a PowerShell script?

To run a script in the **PowerShell**, go to the directory where your script is stored on your PowerShell command. Then, run the script by typing the following command.

```
PS C:\script_folder> ./Script.ps1
```

Here, script_folder is the **path** of the folder where your script is stored, and **Script.ps1** is the name of your script.

```
PS C:\script_folder> .\Script.ps1
```

Alternatively, you can use the above command to run the script.

Q3. How to check powershell version in Windows?

Open your PowerShell **command** and type the following command to find the version,

```
PS C:\> $PSVersionTable.PSVersion
```

The major column value of the result is the version of your PowerShell.

Q4. How to start powershell in windows 10?

To Start the PowerShell, go to the search option and type PowerShell. The PowerShell will be listed in the

option. Click on it open the PowerShell window. Another method is to type PowerShell on your cmd to open the PowerShell window.

Q5. What is $null in PowerShell?

The $null is a variable in the PowerShell that is used to **represent NULL** as the name suggests. It can be assigned to a variable, use it for comparisons, etc. In PowerShell, $null is an object that holds the value of NULL. To assign $null, use the following command

```
PS> $null –eq $variable
```

Here, the **$variable** holds the value of $null.

Q6. How to start and stop a service in PowerShell?

To start a service in the PowerShell, use the start-service command.

```
PS > start-service ServiceName
```

// here ServiceName is the name of the service that you wish to start. To stop a service in the PowerShell, use the stop-service command.

```
PS> stop-service ServiceName
```

// here ServiceName is the name of the service that you wish to stop.

Q7. What is String Interpolation in PowerShell?

The String Interpolation is the way of displaying the value of the variable by surrounding the variable in the double-quotes.

```
//example

$var = “cool” Echo “the value is $var”
```

The above statement produces the **‘value is cool’**. Here the **$var** is surrounded by double quotes when used in the echo statement. So, it prints the value of the variable instead of the variable name itself. It is called string interpolation.

Q8. What are cmdlet's in PowerShell?

The PowerShell **cmdlet (Command let)** is just a group of commands that are used in the PowerShell to perform a function. It is a lightweight .NET framework class object that is invoked by the PowerShell runtime. Programmers can create and invoke it manually too. You can construct your cmdlet by grouping a few lines of PowerShell code. The cmdlet has a **.ps1** extension.

Q9. What is use of Loop in PowerShell?

The loop in the PowerShell is used to execute **a single** or **group** of statements for a repeated number of times or until the conditions become false.

The cmdlet has the following types of loop such as:-

for loop,

forEach loop,

while loop, and **do-while loop**.

Q10. How to call a function in powershell?

To call a **function** in PowerShell, **first**, you have to declare the function using the function keyword. Then to call the function, just type and enter the function name.

//example

```
function functionName {
$args[0] - $args[1]
}
PSC:\>functionName 10 5 5
```

You can also pass arguments to the called function as you see in the example above.

Q11. What is $PSScriptRoot in PowerShell?

The **$PSScriptRoot** is an automatic variable in the PowerShell. It is used to define the filesystem path to the script when it is used inside a module. When it is used outside a module, it just becomes a **$null variable**. It can be used to load additional resources from the script location by pointing to it.

Q12. What does $_ mean in PowerShell?

$_** in the PowerShell is the **‘THIS’** toke. It refers to the current item in the pipeline. It can be considered as the alias for the automatic variable **$PSItem.

```
//example
PS> 1, 2 | ForEach-Object {Write-Host $-}
```

The above code prints **1 2**. It represents the current value and prints it.

Q13. What are Filters in PowerShell?

A lot of commands in the PowerShell return values that are not always useful. To get useful return objects from the commands we can **use filters**. Filters can be created using the Filter parameter or even using the Where-Object. We can use the Filter parameter with the **Get-ChildItem** to get the items that we specified in the filter.

```
//example PS> Get-ChildItem -Path C:\folder\ -Filter '*1*.txt'
```

The above command filters and returns the files with a **prefix as ‘1’**. The same result can be achieved using the Where-Object, but generally, **the filter** is much faster than the Where-Object.

Q14. Are PowerShell commands are case sensitive?

The PowerShell **commands** are **not case sensitive**. So, the capitalization in the commands doesn’t matter to the PowerShell.

Q15. What are Automatic Variables? Enlist some commonly used automatic variables?

Automatic variables in the PowerShell are used to **store** the state information. It is created and maintained by the PowerShell. These are mainly **read-only** variables.

Some examples of automatic variables are,

- **$$ -** it contains the last token received by the session.
- **$/ -** it contains the execution status of the last executed operation in the PowerShell.
- **$^ -** it contains the first token in the last line.
- **$null –** it contains the NULL value.
- **$_ –** it contains the current object in the pipeline.
- **$IsWindows –** it contains
- **$TRUE** if you are running on windows or false.
- **$Error –** it contains an array of error objects in the order of the most recent error.
- **$false –** it contains the value FALSE.
- **$HOME –** it contains the path of the user’s home directory.
- **$PID –** it contains the process ID.
- **$IsLinus –** it returns true if you are running on Linux or false.

- **$IsMacOS –** it returns true if you are running on Mac OS or false.
- **$PSItem –** it contains the current value in the pipeline.

Q16. What are $Home and $PID in PowerShell?

The $HOME is an automatic variable in the PowerShell. It contains the path of the user's home directory. It is equivalent to `C:\Users\.` The **$PID** is also an automatic variable that contains the process identifier of the process which is hosting the PowerShell's current session.