

PowerShell Interview Questions



PowerShell

Interview
Questions

A list of frequently asked **PowerShell Interview Questions** and Answers are given below.

1) What is PowerShell?

PowerShell is an extendable command-line shell which is designed especially for the system administrators and power-users for managing the operating system and processors.

It is a scripting environment which is built on the .NET Framework.

It is platform-independent and open-source.

2) What are the features of PowerShell?

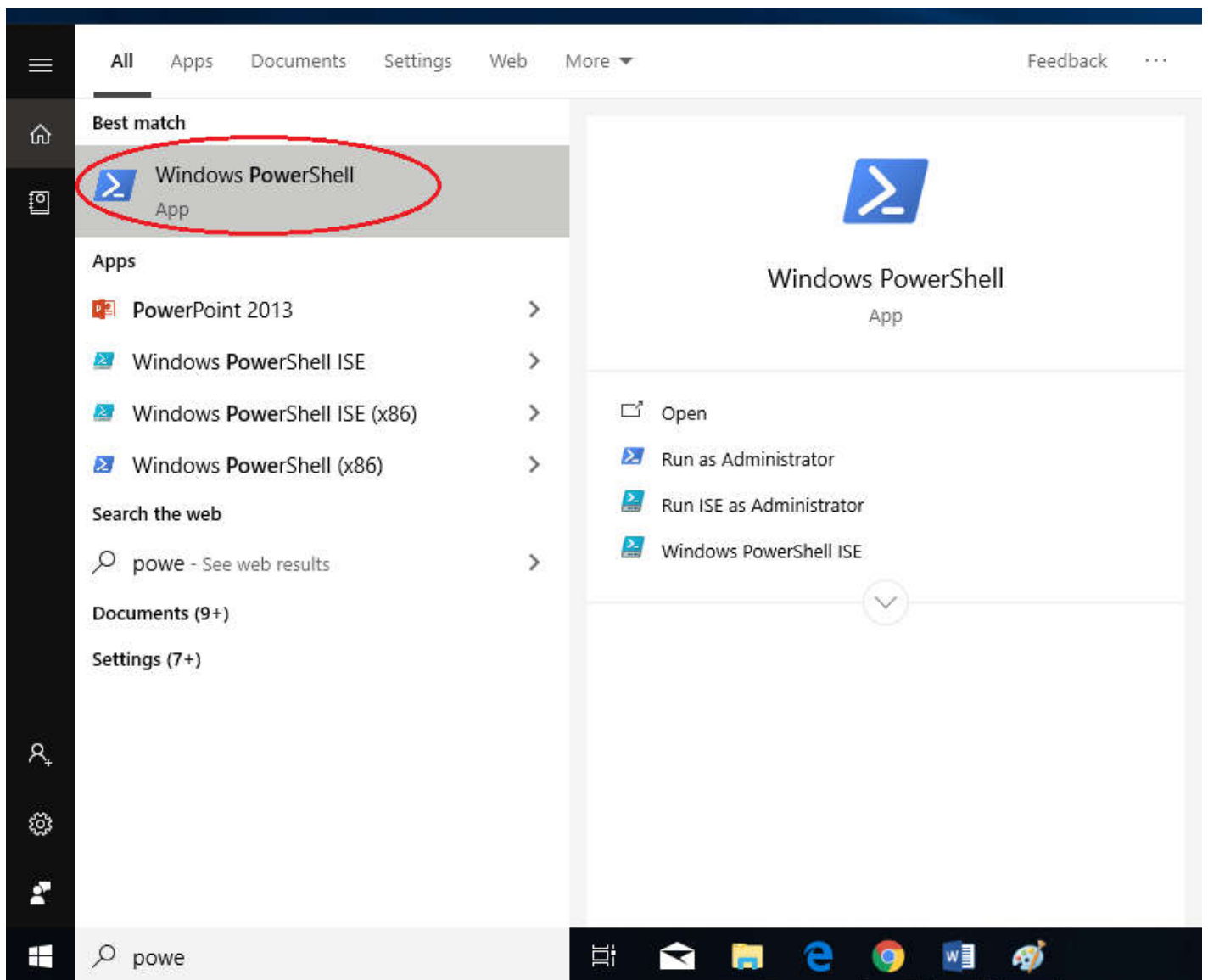
Following are the different features or characteristics of PowerShell:

1. The main feature of PowerShell is that it is based on an object, not on text.
2. The PowerShell commands are customizable.
3. PowerShell remoting is the feature which enables the execution of cmdlets on a remote system that helps the developers for managing the set of remote computers from one single machine.
4. Scripting debugging is also an essential feature of PowerShell, which is used to examine the commands, scripts, functions, and expressions at the time of PowerShell execution.

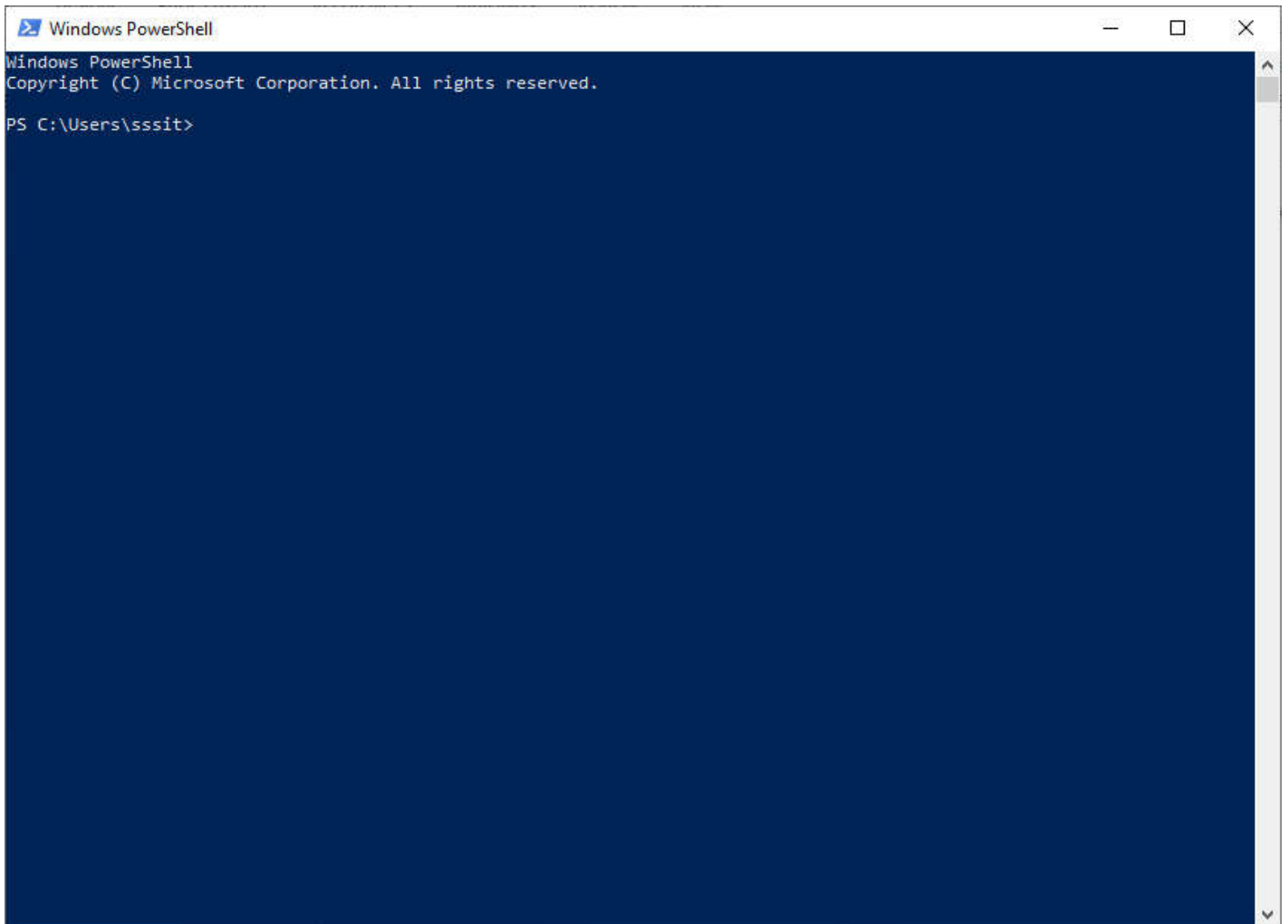
3) How to launch the PowerShell in the Windows operating system?

PowerShell is already installed in all the latest versions of Windows. We need to start PowerShell by the following given steps:

1. Search for the Windows PowerShell. Select and Click.



2. PowerShell Window will open.



4) What is a pipeline in PowerShell?

A PowerShell pipeline is the series of statements which are connected by the pipe '|' operator. In a command, each pipeline operator is used to send the output or result of the preceding command to the next command.

Following box describes the syntax of a pipeline:

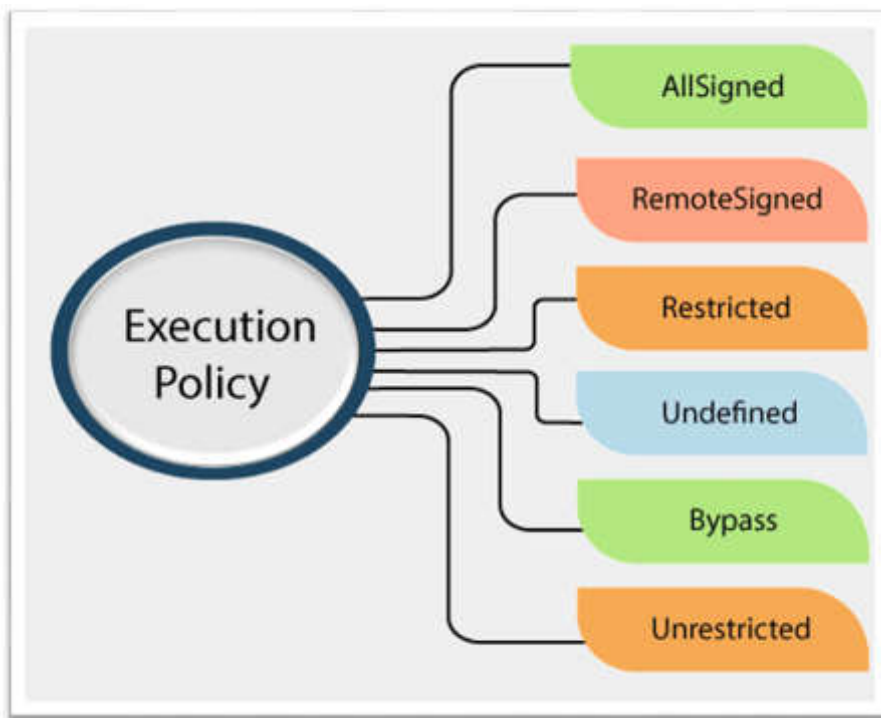
```
Command-1 | Command-2 | Command-3
```

5) What is the Execution policy in PowerShell?

Execution policy in PowerShell is a concept which controls the conditions under which PowerShell loads the configuration files and execute scripts.

6) Explain the types of Execution policy?

Followings are the six types of execution policy in PowerShell:



1. AllSigned
2. RemoteSigned
3. Restricted
4. Undefined
5. Bypass
6. Unrestricted

AllSigned

It is an execution policy which executes only those scripts that are signed by a trusted publisher with a digital signature.

RemoteSigned

PowerShell will execute any script which is written on a local computer. But the script which is downloaded from the internet must require the digital signature from the trusted publisher.

Restricted

This type of execution policy does not allow to execute any script but permits the individual commands. This policy is available for the Windows client computer by default.

Undefined

No execution policy is set or defined in the current scope.

Bypass

It is mainly designed for those configurations in which a PowerShell script is built into a larger application. In this type of policy, no warnings and no prompts are provided.

Unrestricted

This policy is the default for Non-Windows computers. This policy executes those scripts which are unsigned.

7) What is Try, Catch, and finally in PowerShell?

Try: It is a part of a script where we want the PowerShell to monitor the errors. If an error occurs in this block, the automatic variable `$Error` stores the error. And then, the PowerShell searches the Catch block to handle the error.

Catch: In a PowerShell script, it is a part which handles the errors generated by the Try block.

Finally: In a PowerShell script, it is a part which releases the resource that no longer needed by a script.

8) What are the differences between CIM and WMI?

Following table describes the difference between CIM and WMI:

CMI	WMI
CIM stands for Common Information Model.	WMI stands for Windows Management Instrumentation.
It was developed by a company Called DMTF.	It was developed by Microsoft as well.
It can work on any platform.	It works only for windows.

9) What is the function of the \$input variable?

The function of the **\$input** variable enables the PowerShell functions for accessing the data which is coming from the pipeline.

10) What is the difference between PowerShell and CMD?

Following are the differences between PowerShell and CMD:

PowerShell	Command Prompt (CMD)
1. It is a task-based command-line shell and associated scripting language which is based on the .NET framework.	1. It is the default command-line interpreter for the Microsoft Windows Operating System.
2. It can interpret both the commands of a batch and PowerShell.	2. It can interpret only the commands of a batch.
3. It controls and automates the Windows operating system and applications on a Windows server.	3. It executes the commands which are specified on the console to debug the problem.
4. The output generated by the PowerShell is not just a stream of characters (text) but a collection of objects.	4. The output generated by the command prompt is just a stream of characters (text).

5. It is both a shell and scripting environment which supports the creation of large files for managing the windows operating system.

5. It is just a shell environment, which allows a user to do only easy and basic scripts for the execution of the batch file.

11) How to use comments in PowerShell?

Like other scripting languages, PowerShell also supports single or multi-line comments.

Single-Line comments: "Those comments which use the (hash) # symbol at the beginning of each line".

Syntax of single-line comment:

```
<Any Command or statement> # <Any comment>
```

Multi-line comments: "Those comments which use the # (hash) symbol at the end of last line and beginning of the first line".

Syntax of Multi-line comment:

```
<# Multiple-line Comment.....  
.....  
.....#>  
Statement-1  
Statement-2  
Statement-N
```

12) Explain the significance of PowerShell brackets?

Braces Brackets {}: These types of brackets are used in the blocked statements.

Parenthesis Brackets (): These brackets are used for required arguments.

Square Brackets []: These types of brackets used to define the optional items.

13) What is the variable in PowerShell?

A PowerShell variable is a unit of memory which is used to store the data. You can easily store all the types of values in the variable of PowerShell. In the variables, PowerShell contains integers, strings, and objects.

14) How to declare and create a variable in PowerShell?

Declaration: In PowerShell, you can declare a variable by using the \$ (dollar) sign at the beginning of the variable name. Following syntax describes how to declare the variable:

```
$ <variable_name>
```

For example: \$var

Creation or Initialization: In PowerShell, you can create a variable by assigning the value to a variable using the assignment operator. Following syntax describes how to declare the variable:

```
$ <variable_name> = <value>
```

15) Enlist the two ways of extending PowerShell?

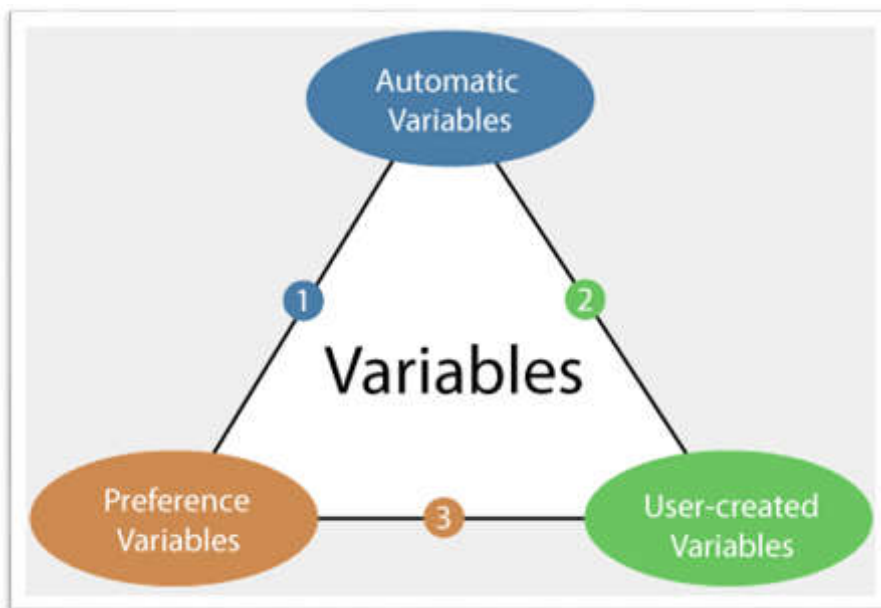
1. PSSnapins
2. Modules

16) Enlist the two methods by which input of pipeline works?

1. ByValue
2. ByPropertyName

17) Name the Different types of variables in PowerShell?

Following are the three types of variables in PowerShell:



1. User-Created variables
2. Automatic variables
3. Preference variables

18) What is the Automatic variable in PowerShell and enlist the common automatic variables?

There are so many predefined variables in PowerShell, which are known as the automatic variables. These variables mainly store the information about the PowerShell, and created and maintained by the PowerShell. Any user can't change or update the value of these variables.

Following are some common automatic variables:

- \$\$
- \$?
- \$^
- \$_
- \$args
- \$Error
- \$foreach
- \$Home
- \$input
- \$null
- \$PSHome
- \$PWD

19) Explain the array in PowerShell?

An array in a PowerShell is a type of data structure which stores the collection of values in the single variable.

20) What is a hash table in PowerShell?

The hash table in PowerShell is also called as a dictionary or an associative array. It is a compact data structure which is used to store each value by using a key. For every hash table, there is an object of a hash table in the PowerShell.

Following box describes the syntax of hash table:

```
$variable_name = @{ <key1> = <value1> ; < key2> = <value2> ; ..... ; < keyN> = <valueN>;}
```

21) Name the different type of PowerShell operators?

Following are the different types of operators in PowerShell:

1. Arithmetic Operator
2. Assignment Operator
3. Comparison Operators
4. Logical Operators
5. Redirection Operators
6. Split and Join Operators

22) Explain the Comparison operator in PowerShell?

Comparison Operators are used in PowerShell for comparing the values. Following are four types of comparison operators:

1. Equality Comparison Operator
2. Match Comparison Operator
3. Containment Comparison Operator
4. Replace Comparison Operator

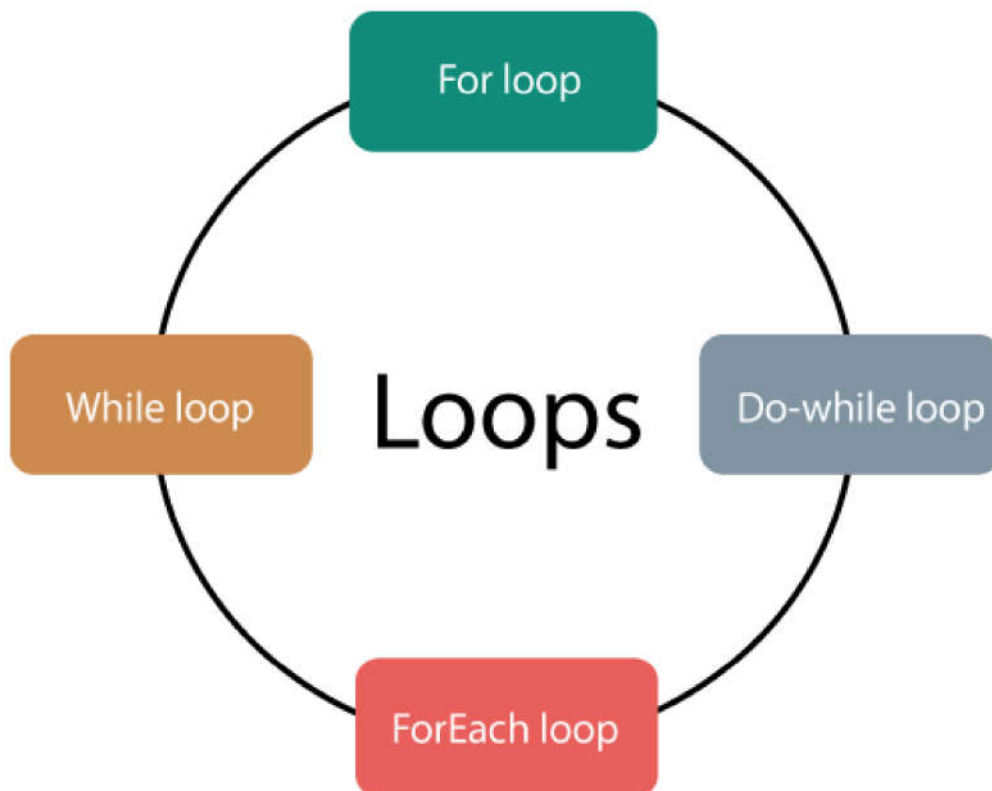
23) What are cmdlets in PowerShell?

The cmdlets are the commands which are written in a .net language like VB or C# introduced by the Windows PowerShell. The cmdlets return the object of Microsoft .NET framework to the next commands in the pipeline. The PowerShell cmdlets follow the **verb-noun** pattern, such as **get-help**, **set-Item**.

24) What is a loop in PowerShell?

When you need to execute the block of commands or statements several times in PowerShell, use the concept of loops.

Following are the several types of Loops in PowerShell:



1. While loop
2. Do-While
3. For loop
4. ForEach loop

25) Explain the different types of loops in PowerShell?

1. While loop: This loop is also called an entry-controlled loop. It executes the commands in a block sequentially, when the given condition evaluates to True. The following box describes the syntax of while loop:

```
while(test_expression)
{
    Statement-1
    Statement-2
    Statement-N
}
```

2. For loop: In this loop, the commands or statements in a block are executed one by one, if the condition returns a True value, otherwise the loop terminates. Following box describes the syntax of For loop:

```
for (<Initialization>; <Condition or Test_expression>; <Repeat>)
{
    Statement-1
    Statement-2
    Statement-N
}
```

3. Do-While and Do-Until loop:

- **Do-while:** This loop is called an exit-controlled loop. This loop executes a block of commands at least once because the condition is checked after the block. In this loop, if the condition is True, then the commands in the block are executed. Otherwise, the loop terminates. The following box describes the syntax of Do-while loop:

```
Do
{
    Statement-1
    Statement-2
    Statement-N
} while( test_expression)
```

- **Do-Until:** This loop is also called an exit-controlled loop. This loop also executes a block of commands at least once because the condition is checked after the block. In this loop, if the condition is False, then the commands in the block are executed. Otherwise, the loop terminates. The following box describes the syntax of Do-Until loop:

```
Do
{
    Statement-1
```

```
Statement-2
Statement-N
} until( test_expression)
```

4. ForEach loop: This loop is used in PowerShell to access the array or a collection of objects, numbers, strings, etc. Following box describes the syntax of ForEach loop:

```
Foreach($<item> in $<collection>)
{
    Statement-1
    Statement-2
    Statement-N
}
```

26) Name the command which is used to copy a file, registry key, or folder?

Copy-Item is a command which is used to copy the files or folders in a file system drive and the registry keys in the registry drive.

27) Name the type of format commands which are used to format the data.

1. Format-List
2. Format-Table
3. Format-Wide
4. Format-Custom

28) How to rename a variable?

Following example describes how to rename a variable:

```
PS C:\> $x
PS C:\> $x=50
PS C:\> $x
50
PS C:\> Rename-Item -Path variable:x -NewName y
PS C:\> $y
50
PS C:\>
```

29) Explain the PowerShell get-command?

The **get-command** is a cmdlet which displays all the installed commands on the computer system. It displays all the commands, filters, aliases, scripts, and applications.

30) How to execute a PowerShell script?

To run a PowerShell script, you need to follow the given steps:

Step 1: Type the commands in a text editor, such as Notepad.

Step 2: Save the file with the .ps1 extension.

Step 3: Open the PowerShell command-shell, and execute the file by typing the name of the PowerShell file.