* **What is powershell?**

<https://docs.microsoft.com/en-us/powershell/scripting/overview?view=powershell-5.1>

PowerShell is a cross-platform task automation solution made up of a command-line shell, a scripting language, and a configuration management framework. PowerShell runs on Windows, Linux, and macOS.

* **CmdletBinding :**

<https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.core/about/about_functions_cmdletbindingattribute?view=powershell-5.1>

<https://stackoverflow.com/questions/14671051/what-is-cmdletbinding-and-how-does-it-work>

CmdletBinding, Parameter etc. are special attribute classes that scripters can use to define PowerShell's behavior, e.g. make a function an Advanced function with Cmdlet capabilites.

When you call them via e.g. [CmdletBinding()] you initialize a new instance of the class.

**In order to turn our function into an advanced function, simply add the CmdletBinding declaration. This attribute makes the function operate similar to compiled cmdlets written in C#**

* There are methods of input processing that can help control your function’s workflow.
  + Begin: Contains all the code that is needed to execute at the beginning of the function.
  + Process: Contains the main functionality of the function.
  + End: Contains all the code that is needed to execute at the end of the function.

For examples: <https://www.sapien.com/blog/2019/05/13/advanced-powershell-functions-begin-to-process-to-end/>

**Begin:**

The Begin block is an optional, preprocessing of the function that will only run once per call of the function. Use this block to setup the function by initializing objects such as variables, database connections, or arrays that will be used throughout the function. Any variables that are created in the Begin block will be accessible elsewhere in the function. Again, the Begin block is optional and is not required if you just want to use either the Process or End blocks.

**Process:**

The Process block is used to specify the code that will continually execute on every object that might be passed to the function. A function can have a Process block without the other blocks, and a Process block is mandatory if a parameter is set to accept pipeline input.

The Process block behaves differently depending on how the input is passed:

* + Values passed by parameter will only be processed once in the process block, and the loop will run through all of the objects passed.
  + When passing by pipeline, the foreach loop is redundant as it will only run once, but the process block will execute once for each item on the pipeline.

**END:**

After all objects have been sent through the pipeline, the End block is called. Like the Begin block, the End block is called once per function call. It is optional; one-time post-processing. Think of this as a place to finalize the function. It is a good practice to have an End block even if it is left empty.