Cmdlets are the basic, lightweight commands in the Windows PowerShell environment. They are designed to work in conjunction with the pipeline, processing objects as input and outputting objects. Cmdlets are named using a "Verb-Noun" format, where the verb represents the action and the noun represents the object of that action. [1, 2, 3, 4, 5]

Here's a more detailed explanation:

Key Characteristics of Cmdlets:

- **Verb-Noun Naming:** Cmdlet names are structured as "Verb-Noun" pairs, clearly indicating the action and the target object (e.g., Get-Process, Start-Service, New-Item). [1, 1, 5, 5]
- **Pipeline-Oriented:** Cmdlets are designed to work with the PowerShell pipeline, meaning they can be chained together to perform complex tasks. [1, 1, 2, 2]
- **Object-Oriented:** Cmdlets process and output objects, not just text streams. This allows for more flexible and powerful scripting. [2, 2, 4, 4, 6, 7]
- **Built-in and Custom:** PowerShell includes a vast library of cmdlets for various tasks, and you can also create your own custom cmdlets. [1, 1, 5, 5]
- **Not Executables:** Cmdlets are not standalone executable files but are part of PowerShell's core functionality. [1, 1]

How Cmdlets Work:

1. **Input:** Cmdlets can receive input from the pipeline (other cmdlets or data streams) or can be provided with input directly as parameters. [2, 3]
2. **Processing:** They perform a specific action on the input, based on their verb and noun. [1, 5]
3. **Output:** Cmdlets return objects as output, which can then be piped to subsequent cmdlets in the pipeline. [1, 2]

Example:

The Get-Process cmdlet retrieves information about running processes. You can use it to get a list of all processes, filter them based on specific criteria, and then pipe the results to another cmdlet like Sort-Object to sort the list. [1, 5, 8, 9, 10, 11, 12]