PowerShell is a scripting language and shell developed by Microsoft for task automation and configuration management. It is built on the .NET framework and uses objects to represent data. These objects have properties and methods, allowing for more complex data manipulation than simple text-based scripting. Cmdlets are the basic commands used in PowerShell, designed to perform specific actions.

Objects

PowerShell works with objects, which are structured collections of information. Instead of dealing with raw text, PowerShell commands return objects that have properties (attributes) and methods (actions). For example, the Get-Process cmdlet returns process objects, and you can access their properties like Name, ID, and CPU.

Arrays

Arrays in PowerShell are used to store collections of items. An array can hold multiple objects of the same or different types. To create an array, you can assign multiple values to a variable, separated by commas.

```
$myArray = 10, 20, 30, "apple", "banana"
```

You can access elements in an array using their index, starting from 0.

```
$myArray[0]  # Accesses the first element (10)
$myArray[-1] # Accesses the last element ("banana")
```

Arrays in PowerShell are dynamic, meaning their size can be adjusted. You can add elements using the += operator.

```
$myArray += "orange"
```

Variables

Variables in PowerShell are used to store values. They are denoted by a $ symbol followed by the variable name. Variable names are not case-sensitive and can include letters, numbers, and underscores.

```
$name = "John Doe"
$age = 30
$isValid = $true
```

PowerShell is dynamically typed, meaning you don't need to explicitly declare the data type of a variable. However, you can use type casting if needed.

```
[int]$number = "123"
```

Variables can have different scopes, such as global, script, and local, which determine their visibility and lifetime.