

How-to: Reference Variables

A Reference variable is used to pass values into a **function**. By default, PowerShell **variables** are created with a "Local" scope, so a variable definition like `$myvar = 'Hello World'` will be visible only to the current script or the current function.

A reference variable (defined with type: `[ref]`) is able to change the value of another variable that is passed to it. This means that you can write a **function** that directly modifies one, (or more likely several) existing variables, rather than just returning a value.

Examples

```
PS C:\> function add5([ref]$num)
>> {
>> $num.value = $num.value + 5
>> }
```

```
PS C:> $testing = 2
PS C:> $testing
2
PS C> add5 ([ref]$testing)
PS C> $testing
7
```

```
# Extending this to modify two variables:
PS> function add5minus5([ref]$add,[ref]$minus)
>> {
>> $add.value = $add.value + 5
>> $minus.value = $minus.value - 5
>> }
```

If the parameter passed is not a reference variable, an *InvalidArgument* exception will be thrown. The parameters passed must also match the *type* required, integer, string etc.

“To take the measure of oneself by reference to one’s colleagues leads to envy or complacency rather than constructive self-examination” ~ Benno C. Schmidt, Jr.

Related PowerShell Cmdlets:

Variables - Create/read variables.

Set-PSBreakpoint - Set a breakpoint on a line, command, or variable.

New-Variable - allows a choice of "Global", "Local", "Private" or "Script"

Get-Variable - Get a PowerShell variable.

Set-Variable - Set a variable and a value.