

# Beginning Programming with Powershell

## Cheat Sheet

### Variables:

\$a	Variable
\$a=5	Variable with value
[Int]\$a	Typed variable
\$a   gm	Determine variable type
\$a + \$b	Add int variables, or concatenate strings
'\$a'	Returns the literal string "\$a"
"\$a"	Returns the value held in \$a

### Pipes and operators:

Dir   gm	Passes the actual directory objects to the "get-member" function
Where-Object{}	Filtering loop operator
?{}	Filtering loop operator (same as Where-Object{})
ForEach-Object{}	Loop operator
%{}	Loop operator (same as ForEach-Object{})
\$_	Current object in the pipeline
\$_ .Name	The name of the current object in the pipeline
-eq	"Equals" comparison operator
-gt	"Greater than" comparison operator
-lt	"Less than" comparison operator
If ( ) {} ELSE {}	If/then operator.
Try {} Catch {}	

### Cmdlets:

Get-Help	Alias <b>help</b>
Where-Object	Alias <b>?</b>
ForEach-Object	Alias <b>%</b>
Get-Member	Alias <b>gm</b>
Get-Service	
Get-Process	
Get-ChildItem	Alias <b>dir</b> , also <b>gci</b>
Out-File	
Format-Table	Alias <b>ft</b>
Format-List	Alias <b>fl</b>
Test-Path	

More:

-ErrorAction	Determine how errors are handled, e.g. silentlycontinue, Stop, Continue, Ignore, Inquire
-ErrorVariable	Create an error variable for use later, e.g. -errorvariable +err \$err
Param()	Create parameters, so you can pass values into the script. Example: param ([string]\$FilePath)
function fname(\$var) {}	Create functions to componetize your code.
fname \$var	Function call.
.. \IncludeFile.ps1;	Dot-source include a file, so it runs as part of your script.

## References

- Us:
  - [Twitter.com\KenpoDBA](#)
  - [Twitter.com\MidnightDBA](#)
  - [MidnightDBA.com](#)
  - [MidnightSQL.com](#)
  - [MinionWare.net](#)
- Powershell documentation on Microsoft.com: <https://msdn.microsoft.com/en-us/powershell/scripting/powershell-scripting>