PowerShell Cheatsheet

Overview

PowerShell is a cross-platform, object-oriented scripting language and command-line shell created by Microsoft. It is designed for system administrators and developers to automate tasks and manage configurations across Windows, Linux, and macOS.

Variables

PowerShell variables are prefixed with a `$` symbol and can store a variety of data types, including strings, integers, arrays, and objects.

```
# String variable
$name = "Alice"

# Integer variable
$age = 30

# Array variable
$numbers = @(1, 2, 3, 4, 5)

# Object variable
$person = @{
    Name = "Bob"
    Age = 40
}
```

Functions

PowerShell has a large number of built-in cmdlets (pronounced "command-lets") for common tasks such as string manipulation, file I/O, and process management. Cmdlets are called using a verb-noun syntax.

```
# String manipulation
$name = "Alice"
$length = $name.Length  # Returns 5

# File I/O
Get-Content -Path "input.txt"

# Process management
Get-Process | Where-Object {$_.Name -eq "notepad"}
```

Loops

PowerShell has several types of loops, including `for`, `foreach`, `while`, and `do-while`. The `for` and `foreach` loops are used to iterate over arrays or lists, while the `while` and `do-while` loops are used to repeat a block of code while a condition is true or false, respectively.

```
# For loop
for ($i = 0; $i -lt 10; $i++) {
    Write-Output $i
}

# Foreach loop
$numbers = @(1, 2, 3, 4, 5)
foreach ($num in $numbers) {
    Write-Output $num
}

# While loop
$i = 0
while ($i -lt 10) {
    Write-Output $i
    $i++
}

# Do-while loop
$i = 0
do {
    Write-Output $i
    $i++
} while ($i -lt 10)
```

Conditionals

PowerShell has several conditional statements, including `if`, `elseif`, `else`, and `switch`. These statements are used to control the flow of a program based on certain conditions.

```
# If statement
$age = 30
if ($age -ge 18) {
    Write-Output "You are an adult"
}

# If-else statement
$age = 15
if ($age -ge 18) {
    Write-Output "You are an adult"
} else {
    Write-Output "You are a minor"
}

# If-elseif-else statement
$age = 25
if ($age -lt 18) {
    Write-Output "You are a minor"
} elseif ($age -lt 65) {
    Write-Output "You are an adult"
} else {
    Write-Output "You are a senior"
}
```

```
# Switch statement
$fruit = "apple"
switch ($fruit) {
    "apple" {
        Write-Output "It's an apple"
    }
    "banana" {
        Write-Output "It's a banana"
    }
    default {
        Write-Output "It's something else"
    }
}
```

File Manipulation

PowerShell provides several cmdlets for manipulating files, including `Get-Content`, `Set-Content`, `Rename-Item`, and `Remove-Item`.

```
# Read file contents
Get-Content -Path "input.txt"

# Write to file
Set-Content -Path "output.txt" -Value "Hello, world!"

# Rename file
Rename-Item -Path "input.txt" -NewName "input.old"

# Remove file
Remove-Item -Path "output.txt"
```

Resources

- PowerShell documentation
- PowerShell on GitHub (source code)
- PowerShell Gallery (community repository)
- Learn PowerShell in Y minutes (quick reference guide)