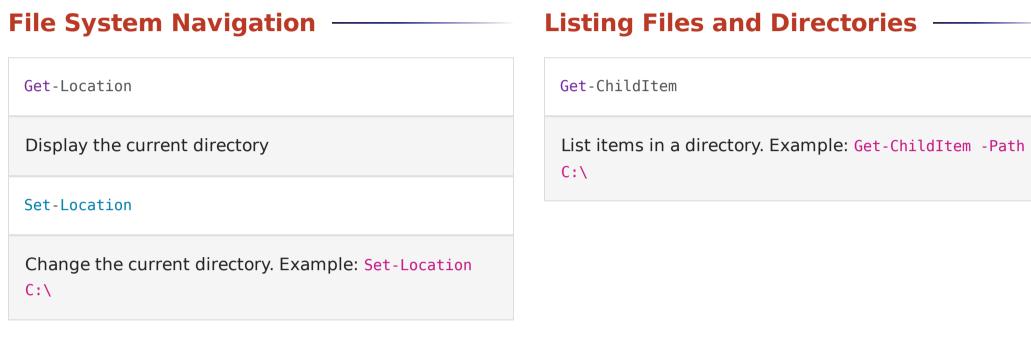
DEV < CODE > HINTS **Categories** Home » Cheatsheet Blogs **PowerShell Cheat Sheet** 22 January 2024 by Huzaif Sayyed Cheatsheet PowerShell is a powerful scripting language and command-line shell developed by Microsoft for task automation and configuration management. It has become an essential tool for Windows administrators, allowing them to efficiently **Recent Articles** manage and automate various tasks. Whether you're a seasoned PowerShell user or just getting started, having a cheatsheet can be incredibly helpful. In this blog post, we'll provide a comprehensive PowerShell cheat sheet to assist you **TypeScript Cheat Sheet** in your day-to-day tasks. **PowerShell Cheat Sheet** Links — Hello World! in Powershell ———— **Shell Scripting Cheat Sheet** Write-Output 'Hello, World!' **PowerShell Documentation** 'Hello, World!' | Write-Output **Bash Cheat Sheet Download Powershell Cheatsheet PDF** This will print Hello, World! 2 times **JavaScript Cheat Sheet ★ Want More Cheatsheet Getting Started with PowerShell Basic Commands** Win + XGet-Help Opening PowerShell: Press Win + X and choose Display help for cmdlets and functions. Example: Get-"Windows PowerShell" or "PowerShell" from the menu. Help Get-Process Run as Administrator: Right-click on the PowerShell Get-Command icon and select "Run as Administrator." List all available commands **Working with Modules** — Import-Module ModuleName Load a PowerShell module Get-Module List loaded modules **Navigating the File System** File System Navigation -Get-Location Get-ChildItem Display the current directory C:\ Set-Location Change the current directory. Example: Set-Location C:\ **Working with Processes**

Type cheatsheet, code Search **Related Posts TypeScript Cheat Sheet Bash Cheat Sheet JavaScript Cheat Sheet Java Cheat Sheet Shell Scripting Cheat Sheet Zig Cheat Sheet Python Cheat Sheet** More Cheatsheet Tutorials >>



```
Get-Process
Display a list of running processes
Stop-Process -Name ProcessName
Terminate a running process
```

Managing Services

Managing Processes -

```
Service Management –
 Get-Service
 List all services
 Start-Service -Name ServiceName
 Start a service
 Stop-Service -Name ServiceName
 Stop a service
```

Working with PowerShell Variables

```
Variables ———
                                                        Arrays —
                                                         $colors = @('Red', 'Green', 'Blue')
 $variable_name
  Create a variable. Example: $name = "John"
                                                         @(): Create an array
 Write-Host $name
 Display the value of a variable.
```

Conditional Statements

```
If-Else Statements —
                                                  Switch Statement —
 if (condition) {
                                                   switch (expression) {
    # Code to execute if condition is true
                                                       'Value1' { # Code for Value1 }
                                                       'Value2' { # Code for Value2 }
 } else {
    # Code to execute if condition is false
                                                      Default { # Code for other values }
```

Looping Constructs

```
ForEach Loop ————
                                           While Loop —
 foreach ($item in $collection) {
                                            while (condition) {
    # Code to execute for each item
                                               # Code to execute while condition is true
PowerShell Functions
```

Shell Scripting Cheat Sheet

```
Functions —
                                                       Function Multiply Example ———
 function MyFunction {
                                                         function Multiply-Numbers {
     param (
                                                            param (
         [parameter_type]$parameter1,
                                                                [int]$num1,
         [parameter_type]$parameter2
                                                                [int]$num2
     # Function code here
                                                            result = num1 * num2
                                                            return $result
                                                         $result = Multiply-Numbers -num1 5 -num2 8
                                                         Write-Host "Result: $result"
Pipeline -
 Get-Process | Stop-Process
 | (Pipe): Pass the output of one command as input to
 another.
Congratulations on reaching the end of this PowerShell Scripting Language Cheatsheet! This resource is
designed to make your coding experience easy and efficient. Feel free to bookmark this page or download
the PDF for future reference. Happy Powershell Scripting!
Cheatsheet
Bash Programming Language, Powershell Scripting, Programming Language Cheatsheet, Shell Scripting
```

TypeScript Cheat Sheet >