

# POWERSHELL 快速参考

Cmdlet、管道、对象、脚本与模块

## 基础

### 命令与帮助

```
Get-Help Get-Process # 查看 cmdlet 帮助
Get-Help Get-Process -Online # 打开在线文档
Get-Command *service* # 按名称查找命令
Get-Alias ls # 查看别名目标
```

### 常用别名

```
ls → Get-ChildItem 列出文件和目录
cd → Get-Location 切换目录
cp → Copy-Item 复制文件或目录
mv → Move-Item 移动或重命名
rm → Remove-Item 删除文件或目录
cat → Get-Content 读取文件内容
echo → Write-Output 输出到管道
cls → Clear-Host 清屏
```

## 变量

### 变量基础

```
$name = "Alice" # 字符串
$count = 42 # 整数
$pi = 3.14 # 浮点数
$list = @(1, 2, 3) # 数组
$hash = @{a=1; b=2} # 哈希表
```

### 自动变量

```
$ # 当前管道对象
$PSVersionTable # PowerShell 版本信息
$HOME # 用户主目录
$PWD # 当前目录
$null # 空值
$true / $false # 布尔常量
$error # 最近错误的数组
$LASTEXITCODE # 上一个原生命令的退出码
```

### 环境变量

```
$env:PATH # 读取环境变量
$env:MY_VAR = "value" # 设置环境变量
Get-ChildItem Env: # 列出所有环境变量
```

## 运算符

### 比较运算符

```
-eq -ne # 等于 / 不等于
-gt -lt # 大于 / 小于
-ge -le # 大于等于 / 小于等于
-like -notlike # 通配符匹配 (*, ?, ~)
-match -notmatch # 正则匹配
-contains # 集合中包含该值
-in -notin # 值在集合中
```

### 逻辑与其他运算符

```
-and -or -not # 逻辑运算符
! # 逻辑非 (别名)
-replace # 正则替换: 'hi' -replace 'h','b'
-split -join # 分割 / 合并字符串
.. # 范围: 1..5 → 1,2,3,4,5
? # 三目运算符 (v7+): '$x? yes: 'no''
```

## 控制流

### if / elseif / else

```
if ($age -ge 18) {
    "Adult"
} elseif ($age -ge 13) {
    "Teen"
} else {
    "Child"
}
```

### switch

```
switch ($color) {
    "red" { "Stop" }
    "green" { "Go" }
    default { "Unknown" }
}
```

## 循环

```
foreach ($item in $list) { $item }
for ($i=0; $i -lt 5; $i++) { $i }
while ($x -lt 10) { $x++ }
1..5 | ForEach-Object { $_ * 2 }
```

## 函数

### 定义函数

```
function Get-Greeting {
    param([string]$Name = "World")
    "Hello, $Name!"
}
Get-Greeting -Name "Alice"
```

### 高级参数

```
function Copy-SafeFile {
    [CmdletBinding()]
    param(
        [Parameter(Mandatory)][string]$Path,
        [Parameter(Mandatory)][string]$Dest
    )
    Copy-Item $Path $Dest -WhatIf:$WhatIfPreference
}
```

## 对象与管道

### 管道基础

```
Get-Process | Sort-Object CPU -Desc | Select-Object -First 5
Get-Service | Where-Object Status -eq "Running"
Get-ChildItem | Measure-Object -Property Length -Sum
```

### 核心管道 Cmdlet

```
(Where-Object) 过滤对象: `Where ($_.CPU -gt 10)`
(Select-Object) 选取属性: `Select Name, CPU`
```

```
(Sort-Object) 排序: `Sort CPU -Desc`
(ForEach-Object) 遍历每个对象: `ForEach-Object {$_.Name}`
(Measure-Object) 计数、求和、均值、最大值
(Group-Object) 按属性值分组
(Format-Table) 以表格显示: `ft -Auto`
(Export-Csv) 导出 CSV: `Export-Csv out.csv`
```

## 文件

### 文件操作

```
Get-Content log.txt # 读取文件
Set-Content out.txt "hello" # 写入 (覆盖)
Add-Content out.txt "more" # 追加
Get-Content log.txt | Select-String "error" # grep
```

### 路径与文件 Cmdlet

```
(Test-Path $path) 检查文件/目录是否存在
(New-Item -Type File) 创建文件
(New-Item -Type Directory) 创建目录
(Resolve-Path) 获取绝对路径
(Join-Path) 组合路径段
(Split-Path) 获取路径或末段
(Get-ItemProperty) 文件属性与元数据
(Remove-Item -Recurse) 递归删除目录
```

## 字符串

### 字符串基础

```
"Hello, $name" # 插值 (双引号)
'Hello, $name' # 字面量 (单引号)
"Length: $($list.Count)" # 字符串中的表达式
@ # 多行
here-string with $name
```

### 字符串方法

```
ToUpper() / ToLower() 大小写转换
Trim() 去除首尾空白
Split(',') 分割为数组
Replace('a','b') 替换子串
StartsWith() / EndsWith() 检查前缀 / 后缀
Substring(0,5) 提取子串
Contains('text') 检查是否包含
-f operator '0 is {1} -f 'sky','blue'
```

## 模块

### 模块管理

```
Get-Module -ListAvailable # 已安装模块
Find-Module -Name Az # 在库中搜索
Install-Module -Name Pester # 从库安装
Import-Module ActiveDirectory # 加载模块
```

### 模块命令

```
(Get-Module) 列出已加载模块
(Import-Module) 将模块加载到会话
(Remove-Module) 从会话卸载模块
(Update-Module) 更新已安装模块
(Get-Command -Module X) 列出模块中的命令
($env:PSModulePath) 模块搜索路径
```

## 常用模式

### 错误处理

```
try {
    Get-Item "C:\missing" -ErrorAction Stop
} catch {
    "Error: $_"
} finally {
    "Cleanup here"
}
```

### 执行策略与远程

```
(Get-ExecutionPolicy) 查看当前脚本策略
(Set-ExecutionPolicy RemoteSigned) 允许本地脚本
(Enter-PSSession -Computer SRV1) 交互式远程会话
(Invoke-Command -Computer SRV1 -Script { }) 在远程执行脚本
(Start-Job { long-task }) 后台运行任务
(Receive-Job -Id 1) 获取后台任务输出
```