



Power Shell



**Certified
Developer**
The Ultimate Tech Degree

DigitalHouse >
Coding School



ÍNDICE

1

Revisão

2

Power Shell

3

Mão na massa





1 | Revisão





Revisão da aula anterior





2 | Power Shell





Principais comandos

Power Shell (Cmdlet)	Power Shell (Alias)	Cmd.exe	Unix Shell	Descrição
Get-ChildItem	gci, dir, ls	dir	ls	Lista todos os arquivos do diretório atual.
Get-Content	gc, type, cat	type	cat	Lista o conteúdo do arquivo.
Get-Command	gcm	help	help	Lista os comandos disponíveis.
Get-Help	help, man	help	man	Comando de ajuda.
Clear-Host	cls, clear	cls	clear	Limpa a tela.
Copy-Item	cpi, copy, cp	copy	cp	Copia um ou mais arquivos.
Move-Item	mi, move, mv	move	mv	Move um arquivo/diretório para um novo local.
Remove-Item	ri, del, erase, rmdir, rd, rm	del, erase, rmdir, rd	rm, rmdir	Deleta um arquivo/diretório.
Rename-Item	rni, ren, mv	ren, rename	mv	Renomeia um arquivo/diretório.
Get-Location	gl, pwd	cd	pwd	Mostra o diretório atual.

Exemplo de aplicação dos comandos:

<https://docs.google.com/document/d/1piCrkLrkTkNDn5nO2cPoAv8slgGiWyyh/edit#heading=h.gjdgxs>





Uso do PowerShell

1 - Automação de tarefas

2 - Gerenciamento de configuração

3 - Monitoramento

4 - Testar infraestrutura

5 - Automação de processos





O que é um módulo

O que é um módulo?

É um pacote que contém objetos do PowerShell, como cmdlets, provedores, funções, fluxos de trabalho, variáveis e aliases. Os objetos ou ações neste pacote podem ser implementados em um script do PowerShell, uma DLL compilada ou uma combinação de ambos. Normalmente esses arquivos são agrupados em um único diretório.

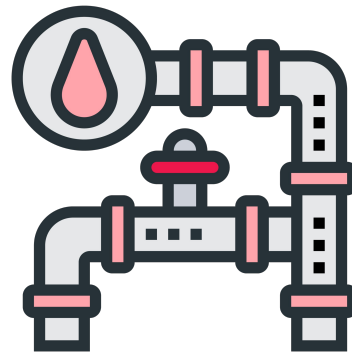




O que é um pipeline?

É uma série de comandos conectados por operadores de 'tubulação' ("|", ASCII 124). Cada operador no pipeline envia os resultados do comando anterior para o próximo comando.

A saída do primeiro comando pode ser enviada para processamento como entrada para o segundo comando e o resultado pode ser enviado para outro comando. O acima é uma cadeia complexa de comandos ou pipeline que é composta por uma série de comandos simples.



Por exemplo: Comando-1 | Comando-2 |
Comando-3

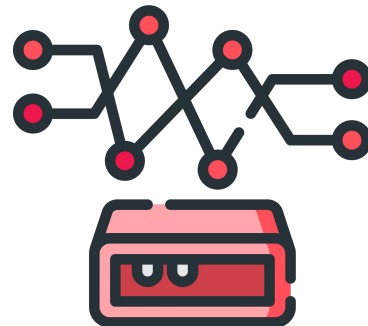




O que é uma variável?

Uma variável é uma unidade de memória na qual os valores são armazenados. No PowerShell, as variáveis são representadas por strings de texto que começam com um cifrão (\$), como \$a, \$process ou \$my_var.

Os nomes de variáveis não diferenciam maiúsculas de minúsculas e podem incluir espaços e caracteres especiais. No entanto, nomes de variáveis que incluem caracteres especiais e espaços são difíceis de usar e devem ser evitados. Para obter mais informações, consulte Nomes de variáveis que incluem caracteres especiais.





O que são estruturas de controle?

As estruturas de controle permitem que você modifique o fluxo de execução das instruções de um programa. Com eles você pode:

De acordo com uma condição, executar um grupo ou outro de instruções (If-Then-Else).

De acordo com o valor de uma variável, executar um ou outro grupo de comandos (Switch-Case).

Executar um grupo de instruções somente quando uma condição for atendida (Do-While).

Executar um grupo de instruções até que uma condição seja atendida (Do-Until).

Executar um grupo de instruções um certo número de vezes (For-Next).





O que é um script do PowerShell?

Um script é um arquivo de texto simples que contém um ou mais comandos do PowerShell. Os scripts do PowerShell têm uma extensão de arquivo

`.ps1`

`code nomearquivo.ps1`

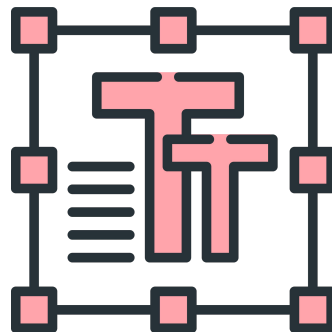
`.sh #/bin/bash`

`.zsh`

`chmod +x frutas.sh`

Executar um script é muito parecido com executar um cmdlet. Eles podem executar scripts no computador ou em uma sessão remota em outro computador.

Escrever um script salva um comando para uso posterior e facilita o compartilhamento com outros usuários. Mais importante, ele permite que você execute comandos simplesmente digitando o caminho do script e o nome do arquivo. Os scripts podem ser tão simples quanto um único comando em um





O que é uma função no PowerShell?

No PowerShell, como em muitas linguagens, uma função é um conjunto de instruções que damos um nome. O principal interesse das funções é que podemos chamá-las várias vezes, sem ter que reescrever as instruções para cada chamada.

Se você escreve scripts ou comandos exclusivos do PowerShell de uma linha e descobre que muitas vezes precisa modificá-los para diferentes cenários, há uma boa chance de que eles sejam bons candidatos para transformá-los em uma função reutilizável.





3

Mão na massa





Exercício



DigitalHouse>
Coding School