

Power Shell

Get-host

```
PS C:\windows\System32> get-host

Name           : ConsoleHost
Version        : 4.0
InstanceId     : e5d5ac4b-58bf-48f8-9eb0-a0112113ca1d
UI             : System.Management.Automation.Internal.Host.InternalHostUserInterface
CurrentCulture : es-ES
CurrentUICulture : es-ES
PrivateData    : Microsoft.PowerShell.ConsoleHost+ConsoleColorProxy
IsRunspacePushed : False
Runspace       : System.Management.Automation.Runspaces.LocalRunspace

PS C:\windows\System32> _
```

Cls = clear limpia

Get-child

```
PS C:\windows\System32> get-childitem c:\

    Directorio: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          22/08/2013         17:52         PerfLogs
d-r--          25/10/2021          9:26         Program Files
d-----          25/10/2021          9:34         Program Files (x86)
d-r--          04/10/2021          9:52         Users
d-----          22/10/2021         13:50         Windows
d-----          22/10/2021         13:51         xampp
-a---          26/10/2021          13:59          161 ejecutable_borrar.bat
-a---          27/10/2021          11:35         8967 pau_ejecutable3.bat

PS C:\windows\System32>
```

Get-alias alias de los comandos

Get-help

```
PS C:\windows\System32> get-help

¿Desea ejecutar Update-Help?
El cmdlet Update-Help descarga los archivos de Ayuda más recientes para los módulos de Windows PowerShell y los instala
en el equipo. Para obtener más información sobre el cmdlet Update-Help, consulte
http://go.microsoft.com/fwlink/?LinkId=210614.
[S] Sí [N] No [U] Suspender [?] Ayuda (el valor predeterminado es "S"):
```

.\lista.txt

```
PS C:\windows\system32> pwd

Path
----
C:\windows\system32

PS C:\windows\system32> pushd
PS C:\windows\system32> popd
PS C:\windows\system32> _
```

Alias

Dir=get-childitem

Gps=get-process

Cls=clear-host

Cmdlet = get-childitem

li=Invoke-item

Sal=set-alias

Get-command alias-> comando

```
PS C:\windows\system32> get-command gps

CommandType      Name                                     ModuleName
-----
Alias             gps -> Get-Process
PS C:\windows\system32> _
```

Get-command comando -> alias

```
PS C:\windows\system32> get-command get-childitem

CommandType      Name                                     ModuleName
-----
Cmdlet           Get-ChildItem                          Microsoft.PowerShell.Management
PS C:\windows\system32> _
```

Set-alias

```
PS C:\windows\system32> set-alias

cmdlet Set-Alias en la posición 1 de la canalización de comandos
Proporcione valores para los parámetros siguientes:
Name: PS C:\windows\system32>
PS C:\windows\system32>
PS C:\windows\system32> set-alias ifconfig ipconfig
PS C:\windows\system32> ifconfig

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . : fe80::74:2a79:bd3:c0ac%12
    Dirección IPv4. . . . . : 192.168.58.1
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 0.0.0.0

Adaptador de túnel isatap.{8A5EBF13-C52A-4AFD-8FEF-5D92A48BCAB5}:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :
PS C:\windows\system32> cls
```

get-proces

```
PS C:\windows\system32> get-process -?
NOMBRE
Get-Process

SINTAXIS
Get-Process [[-Name] <string[]>] [-ComputerName <string[]>] [-Module] [-FileVersionInfo] [<CommonParameters>]
Get-Process [[-Name] <string[]>] -IncludeUserName [<CommonParameters>]
Get-Process -Id <int[]> -IncludeUserName [<CommonParameters>]
Get-Process -Id <int[]> [-ComputerName <string[]>] [-Module] [-FileVersionInfo] [<CommonParameters>]
Get-Process -InputObject <Process[]> -IncludeUserName [<CommonParameters>]
Get-Process -InputObject <Process[]> [-ComputerName <string[]>] [-Module] [-FileVersionInfo] [<CommonParameters>]

ALIAS
gps
ps

NOTAS
Get-Help no encuentra los archivos de Ayuda para este cmdlet en el equipo. Mostrará solo una parte de la Ayuda.
-- Para descargar e instalar los archivos de Ayuda para el módulo que incluye este cmdlet, use Update-Help.
-- Para ver en línea el tema de Ayuda de este cmdlet, escriba "Get-Help Get-Process -Online" o
vaya a http://go.microsoft.com/fwlink/?LinkID=113324.

PS C:\windows\system32> _
```

function

```
PS C:\windows\system32> function prompt {"pau >"}
pau >_
```

Get-app tabulador y va buscando comandos con get-app...

\$valor = Y se asignara ha lo que queramos

\$a = 10 \$a será 10

-eq = igual

-ne = no igual

-gt = más grande que

-ge = mayor o igual

-lt =menor que

-le = menor o igual que

C= case sensitive sensitivo ha mayúsculas

I= case insensitive insensitivo ha mayusculas

\$env:path += ";rut\ejecutable1;ruta\ejecutable2;

clear-variable a

\$

```
PS C:\> cd /windows/System32
PS C:\windows\System32> $precio = 0.35
PS C:\windows\System32> $cantidad = 5
PS C:\windows\System32> $total = $precio * $cantidad
PS C:\windows\System32> $mensaje = "El costo $cantidad naranjas es de $total dolares"
>>
>>
>>
PS C:\windows\System32> $mensaje = "El costo $cantidad naranjas es de $total dolares"
PS C:\windows\System32> $mensaje
El costo 5 naranjas es de 1.75 dolares
PS C:\windows\System32> $
```

Child item

```
PS C:\windows\System32> $DiscoC

    Directorio: C:\

Mode                LastWriteTime         Length Name
----                -
d-----          22/08/2013         17:52         PerfLogs
d-r--          25/10/2021          9:26         Program Files
d-----          25/10/2021          9:34         Program Files (x86)
d-r--          04/10/2021          9:52         Users
d-----          22/10/2021         13:50         Windows
d-----          22/10/2021         13:51         xampp
-a---          26/10/2021         13:59          161 ejecutable_borrar.bat
-a---          27/10/2021         11:35         8967 pau_ejecutable3.bat

PS C:\windows\System32>
```

Todas las variables

```
PS C:\windows\System32> DIR variable:
Name
----
$                CLS
?                True
^                CLS
args             {}
cantidad         5
ConfirmPreference High
ConsoleFileName
DebugPreference SilentlyContinue
DiscoC           (PerfLogs, Program Files, Program Files (x86), Users...)
Error            (El término 'VAR2' no se reconoce como nombre de un cmdlet, función, archivo de scri
ErrorActionPreference Continue
ErrorView        NormalView
ExecutionContext System.Management.Automation.EngineIntrinsics
false           False
FormatEnumerationLimit 4
HOME            C:\Users\Administrador
Host            System.Management.Automation.Internal.Host.InternalHost
input           System.Collections.ArrayList+ArrayListEnumeratorSimple
MaximumAliasCount 4096
MaximumDriveCount 4096
MaximumErrorCount 256
MaximumFunctionCount 4096
MaximumHistoryCount 4096
MaximumVariableCount 4096
mensaje         El costo 5 naranjas es de 1.75 dolares
HyInvocation     System.Management.Automation.InvocationInfo
HostedPromptLevel 0
null
OutputEncoding  System.Text.ASCEncoding
PID             2888
precio          0,35
PROFILE         C:\Users\Administrador\Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
ProgressPreference Continue
PSBoundParameters {}
PSCmdletPath
PSCulture       es-ES
PSDefaultParameterValues {}
PSEmailServer
PSHOME          C:\Windows\System32\WindowsPowerShell\v1.0
PSScriptRoot
PSSessionApplicationName wsman
PSSessionConfigurationName http://schemas.microsoft.com/powershell/Microsoft.PowerShell
PSSessionOption  System.Management.Automation.Remoting.PSSessionOption
PSUICulture     es-ES
PSVersionTable  {PSVersion, WSMANStackVersion, SerializationVersion, CLRVersion...}
PWD             C:\windows\System32
ShellId         Microsoft.PowerShell
StackTrace      en System.Management.Automation.CommandDiscovery.LookupCommandInfo(String command
total          1,75
true           True
VAR1           200
VAR2           200
VerbosePreference SilentlyContinue
WarningPreference Continue
WhatIfPreference False
```

Variables con var

```
PS C:\windows\System32> dir variable:var*
Name
----
VAR1 200
VAR2 200
```

Deletear variable

```
PS C:\windows\System32> dir variable:var*
Name
----
VAR1 200
VAR2 200

PS C:\windows\System32> del variable:\var1
PS C:\windows\System32> $var1
PS C:\windows\System32> _
```

```
PS C:\windows\System32> (15).GetType().Name
Int32
PS C:\windows\System32> (200000000000).GetType().Name
Int64
PS C:\windows\System32> (34.2).GetType().Name
Double
PS C:\windows\System32> _
```

```
PS C:\windows\System32> [int32]$valor = 12
PS C:\windows\System32> ($valor).GetType().name
Int32
PS C:\windows\System32>
```

\$fecha

```
PS C:\windows\System32> [DateTime]$fecha = "January 9, 2021"
PS C:\windows\System32> $fecha

sábado, 9 de enero de 2021 0:00:00

PS C:\windows\System32> $fecha.GetType().NAME
DateTime
PS C:\windows\System32>
```

cls

```
PS C:\windows\System32> $fecha.adddays(9)

lunes, 18 de enero de 2021 0:00:00

PS C:\windows\System32> _
```

Constantes

```
PS C:\windows\System32> $fecha.adddays(9)

lunes, 18 de enero de 2021 0:00:00

PS C:\windows\System32> new-variable constante -value "alexandrio" -Description "prueba de una constante" -Option ReadOnly
PS C:\windows\System32> del variable:\constante
del : No se puede quitar la variable constante porque es constante o es de solo lectura. Si la variable es de solo lectura, intente realizar de nuevo la operación especificando la opción de Force.
En línea: 1 Carácter: 1
+ del variable:\constante
+ ~~~~~
+ CategoryInfo          : WriteError: (constante:String) [Remove-Item], SessionStateUnauthorizedAccessException
+ FullyQualifiedErrorId : VariableNotRemovable,Microsoft.PowerShell.Commands.RemoveItemCommand

PS C:\windows\System32> del variable:\constante -force
PS C:\windows\System32>
```

cls

```
PS C:\windows\system32> $desktop["so", "ip"]
```

windows 10

192.168.2.96

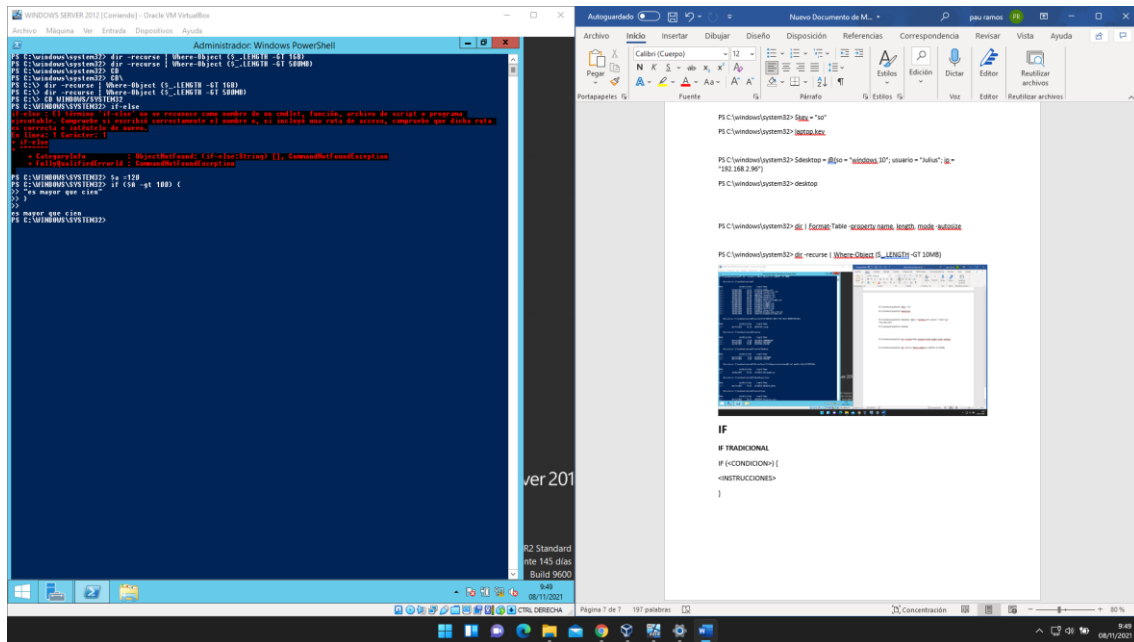
```
PS C:\windows\system32> $so = "LinuxMint"
```

```
PS C:\windows\system32> $laptop = @{so = $so; usuario = "chucho"; ip = "192.168.2.97"}
```

```
PS C:\windows\system32> $laptop.so
```

LinuxMint

}



IF-ELSE

IF else

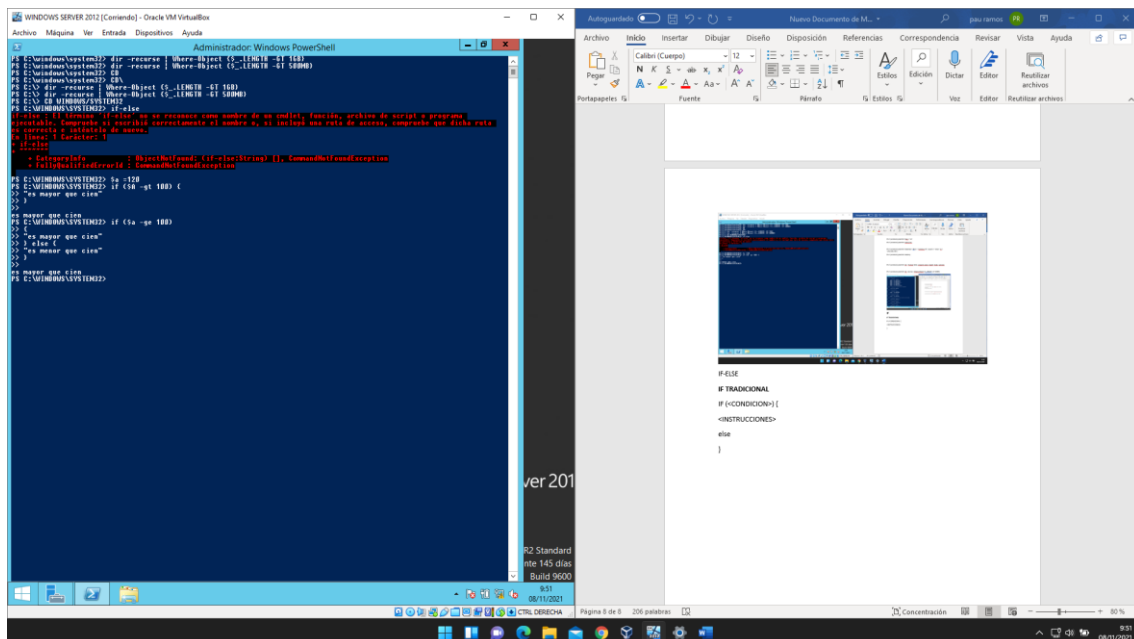
IF (<CONDICION>) {

<INSTRUCCIONES>

} elseif (\$a -lt 200) {

<INSTRUCCIONES>

else}



If else if else

```
IF (<CONDICION>) {  
    <INSTRUCCIONES>  
} elseif {  
    <INSTRUCCIONES>  
} else {  
    <INSTRUCCIONES>  
}
```

Switch

```
(expresiones) {  
    (opción 1)  
PS C:\WINDOWS\SYSTEM32> if ($fruta -eq "mango") {  
    >> "primera coincidencia"  
    >> } elseif ($fruta -eq "manzana") {  
    >> "segunda coincidencia"  
    >> } elseif ($fruta -eq "naranja") {  
    >> "tercera coincidencia"  
    >> } elseif ($fruta -eq "pera") {  
    >> "cuarta coincidencia"  
    >> } else {  
    >> "no se ha encontrado ninguna coincidencia"  
    >> }
```

```
Administrador: Windows PowerShell

PS C:\WINDOWS\SYSTEM32> $fruta = "pera"
PS C:\WINDOWS\SYSTEM32> switch ($fruta)
> "mango" { "primera coincidencia"}
> "manzana" { "segunda coincidencia"}
> "naranja" { "tercera coincidencia"}
> "pera" { "cuarta coincidencia"}
> default { "cuarta coincidencia"}
> }
>
En línea: 1 Carácter: 16
+ switch ($fruta)
+ ~~~~~
Falta el carácter '{' en la instrucción switch .
En línea: 2 Carácter: 9
+ "mango" { "primera coincidencia"}
+ ~~~~~
Token '{' inesperado en la expresión o la instrucción.
En línea: 3 Carácter: 11
+ "manzana" { "segunda coincidencia"}
+ ~~~~~
Token '{' inesperado en la expresión o la instrucción.
En línea: 4 Carácter: 11
+ "naranja" { "tercera coincidencia"}
+ ~~~~~
Token '{' inesperado en la expresión o la instrucción.
En línea: 5 Carácter: 8
+ "pera" { "cuarta coincidencia"}
+ ~~~~~
Token '{' inesperado en la expresión o la instrucción.
En línea: 7 Carácter: 1
+ }
+ ~
Token '}' inesperado en la expresión o la instrucción.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : MissingCurlyBraceInSwitchStatement

PS C:\WINDOWS\SYSTEM32> switch ($fruta) {
> "mango" { "primera coincidencia"}
> "manzana" { "segunda coincidencia"}
> "naranja" { "tercera coincidencia"}
> "pera" { "cuarta coincidencia"}
> default { "cuarta coincidencia"}
> }
cuarta coincidencia
PS C:\WINDOWS\SYSTEM32>
```

Ciclos en powershell

While

While(<expresiones>) {

<instrucciones>

}

```
Administrador: Windows PowerShell

PS C:\WINDOWS\SYSTEM32> $fruta = "pera"
PS C:\WINDOWS\SYSTEM32> switch ($fruta)
> "mango" { "primera coincidencia"}
> "manzana" { "segunda coincidencia"}
> "naranja" { "tercera coincidencia"}
> "pera" { "cuarta coincidencia"}
> default { "cuarta coincidencia"}
> }
>
En línea: 1 Carácter: 16
+ switch ($fruta)
Falta el carácter '{' en la instrucción switch .
En línea: 2 Carácter: 9
+ "mango" { "primera coincidencia"}
+ ~~~~~
Token '{' inesperado en la expresión o la instrucción.
En línea: 3 Carácter: 11
+ "manzana" { "segunda coincidencia"}
+ ~~~~~
Token '{' inesperado en la expresión o la instrucción.
En línea: 4 Carácter: 11
+ "naranja" { "tercera coincidencia"}
+ ~~~~~
Token '{' inesperado en la expresión o la instrucción.
En línea: 5 Carácter: 8
+ "pera" { "cuarta coincidencia"}
+ ~~~~~
Token '{' inesperado en la expresión o la instrucción.
En línea: 7 Carácter: 1
+ }
+ ~
Token '}' inesperado en la expresión o la instrucción.
+ CategoryInfo          : ParserError: (:) [], ParentContainsErrorRecordException
+ FullyQualifiedErrorId : MissingCurlyBraceInSwitchStatement

PS C:\WINDOWS\SYSTEM32> switch ($fruta) {
> "mango" { "primera coincidencia"}
> "manzana" { "segunda coincidencia"}
> "naranja" { "tercera coincidencia"}
> "pera" { "cuarta coincidencia"}
> default { "cuarta coincidencia"}
> }
cuarta coincidencia
PS C:\WINDOWS\SYSTEM32> $i = 10
PS C:\WINDOWS\SYSTEM32> while ($i -ge 0) {}
PS C:\WINDOWS\SYSTEM32> while ($i -ge 0) {
> $i
> $i--
> }
>
10
9
8
7
6
5
4
3
2
1
0
PS C:\WINDOWS\SYSTEM32> _
```

PS C:\WINDOWS\SYSTEM32> \$i = 0

PS C:\WINDOWS\SYSTEM32> shile (\$i -le 10) {

>> \$i

>> &I--

>> }

>>

```
PS C:\WINDOWS\SYSTEM32> $i = 0
```

```
PS C:\WINDOWS\SYSTEM32> shile ($i -le 10) {
```

```
>> $i
```

```
>> &!--
```

```
>> }
```

```
>>
```

Do- while

```
Do{
```

```
$valor++
```

```
Wirite-host "el numero es $valor"
```

```
} while ($valor -ne 5)
```

Do until

```
Do {
```

```
$valor3++
```

```
Wirite-host "el numero es $valor3"
```

```
} until ($valor3 -ge 5)
```

For

```
for ($i = 0; -lt 8; $i++) {
```

```
$i
```

```
}
```

```
For ($i = 0; $($Y = $i * 2; $i))
```

Foreach(<elemento> in <coleccion>)

Foreach (\$i in 1..10) {

\$i

}

PS C:\WINDOWS\SYSTEM32> foreach (\$e in get-process) {

>> '{0}' iniciado el {1}' -f \$e.name, \$e.starttime

>> }

>>

Foreach-object

Get-process Notepad | foreach-object { \$_.closemainwindow()} → cerrar un programa

Get-process | foreach-object {

"el proceso{0} consume {1}"

Get-process | foreach-object

Función<nombre>

PS C:\WINDOWS\SYSTEM32> function hola {

>> "hola mundo"

>> }

>>

PS C:\WINDOWS\SYSTEM32>

PS C:\WINDOWS\SYSTEM32> hola julius

hola julius, bienvenido a powershell

PS C:\WINDOWS\SYSTEM32>

